Load Sharing In Limited Access Distributed Systems*

Venkatesh Harinarayan†
Leonard Kleinrock
Dept. Of Computer Science, University of California, Los Angeles

Abstract

In this paper we examine dynamic load sharing in limited access distributed systems. In this class of distributed systems all servers are not accessible to all sources, and there exist many different accessibility topologies. We focus our attention on the ring topology and provide an analytic model to derive the approximate mean waiting time (our metric of performance). We then consider other limited access topologies and find that rather different interconnection patterns give similar performance measurements. We conjecture that the number of servers accessible to a source is the parameter with the greatest performance impact, in a limited access topology with load sharing. We also introduce another variable called diversity that is indicative of the degree of load sharing and speculate that performance is reasonably insensitive to diversity so long as it is non-zero. Using these conjectures we show how a reasonable estimate of the mean waiting time can be analytically derived in many limited access topologies.

1 Introduction

High performance is achieved in distributed systems by distributing load among the many processors available, and so a prime objective is to ensure that scheduling of jobs to processors is done efficiently and using decentralized control. Many load sharing schemes have been proposed in the literature. Surveys and classifications may be found in [6] and [2]. In this paper we shall be concerned with *Dynamic* load sharing algorithms, which are responsive to the current system state and make decisions based on it.

In many distributed computer systems it is not desirable to allow every source to have access to every server. Two factors which may contribute to the need for limited access are security and the interconnection cost. For example in an general interconnection scheme, where sources and servers are connected over a point to point, wide area network, every server may not be accessible to every source within the same time frame. It may thus be prudent to avoid sending a job for execution to a server in a remote site many hops away. Limited access in such a case would reflect proximity – sources have access only to those servers that are in some sense close to them. With each limited access scheme we can associate a topology which is formed by connecting all the servers that can be accessed by a given source to the source. The limited access topologies that we consider shall be 'regular' in that all sources have the same number of connections; so also do the servers. The number of connections emanating from each source is independent of the size (number of sources) of the network. We impose these restrictions since we are interested in the inherent performance degradation caused by limited access and do not want topology dependent details obscuring the exposition and complicating the analysis.

2 The Model

We first give a formal model of the limited access systems that we consider in this paper. We have two distinct entities in the systems — the sources and the servers. Servers are grouped together to form clusters; servers in a given cluster are accessible from the same sources and cannot be distinguished individually. The number of servers in a cluster is the same over all clusters. The sources all generate job arrivals from a Poisson process. In keeping with the 'regularity' constraint we assume that all sources have identical arrival rates and that all jobs have service times exponentially distributed with the same mean. In summary, we have a set of N sources from each

*supported in part by DARPA contract MDA 903-C-87-0663

†presently with Tandem Computers, Inc.

of which jobs arrive at a rate λ. Each source has access to k clusters each of which consists of m servers. With no loss of generality, the mean service time of a job is taken to be 1. Fig 6, 7 and 8 show examples of limited access topologies (we will examine these in Section 4). Sources are connected to clusters they can access.

In general we require that queues form at the sources only. We also make an assumption that Inter Process Communication (IPC) time is small relative to the mean job service time which means that a source or server can get an instantaneous snapshot of the system state whenever it desires.

- The *arrival* protocol must decide to which accessible cluster a source should send a job, when there is more than one accessible cluster that is not busy. A busy cluster is a cluster, all of whose servers are busy serving jobs; if at least one server is free, the cluster is said to be free. There can be many different protocols which make use of differing amounts of state information.

- The *service* protocol decides which source to serve next when a given job has completed execution.

We define the source degree *(sd)* to be the number of servers (resources) to which a source has access. Likewise, we define the resource degree *(rd)* to be the number of sources that can access a resource. For egodicity, we require that

$$\lambda < sd/rd \tag{1}$$

3 The Ring

In the ring topology sources are arranged in a ring and between every pair of sources lies a cluster of servers. Every source can access only those clusters that are adjacent to it. Thus every source can access exactly two clusters each of which it shares with one of its neighbors.

Let there be N sources $s_0, s_1, \ldots, s_{N-1}$ and N clusters $c_0, c_1, \ldots, c_{N-1}$. Then s_i is connected (has access) to clusters c_i and $c_{(i+1)modN}$ only.

Let m be the number of servers in a cluster. The source degree (sd) is then $2m$ and the resource degree (rd) is 2. Thus the condition for ergodicity from equation 1 is

$$\lambda < m \tag{2}$$

The arrival protocol that we use in our analytic model is the first-fit arrival protocol, which means that a new arrival to a source is sent for execution to a cluster only if there is a free cluster available, otherwise it is queued at the source. In case both clusters are free, one of them is selected at random.

We use the 'capturing' service protocol. In this protocol, a server that has finished executing a job returns to serve the source from which the job came. Only in case this source has no jobs in its queue awaiting service is the server free to serve the other source that can access it. Thus once a server is 'captured' by a source it is forced to serve that source until that source has no more backlogged work. This protocol is similar to a cyclic service protocol in a complete access topology as analyzed in [5].

3.1 Intractability

The Markov chain that describes the system is $3N$ dimensional and it can easily be seen to be nonreversible [3]. This state space explosion coupled with irreversibility make an exact solution for the steady state probabilities for this Markov Process prohibitive. We circumvent this intractability by zooming in on one source or cluster and attempt to approximate the effects of the other sources and clusters on this entity.

We decouple the queue at a source from the number of servers captured by the source. Since we are interested in the mean waiting time which can be derived from the solution for the source queue length distribution, we replace the number of servers captured by the source by the average number of servers captured by the source.

We also neglect the influence of clusters which are not adjacent to a given cluster in solving for the steady state probabilities of the given cluster. We formalize our assumptions and derive a value for the approximate mean waiting time below.

3.2 The Approximate Solution

We use the following notation in the solving for the approximate mean waiting time of a job.

f_i is the probability that a cluster has i servers idle (free),$0 \leq i \leq m$.

b is the probability that both clusters accessible to a given source are busy; in other words the source is 'blocked'.

q_i denotes the probability of there being i, $i \geq 0$, jobs in the queue of a source, given that the source is blocked.

Finally, W, the object of our search is the average waiting time of a job.

Our assumption that ripple effects of order 2 and above be ignored is equivalent to stating that clusters that are not adjacent to each other behave independently of one another.

We focus on two Markov processes of interest – the source queue process and the cluster process.

With probability b (the source queue is in the blocked phase) the source queue process can be described by the markov chain in fig. 1; with probability $1-b$ the source is not blocked and so has a queue length of 0. The source queue process has as its state descriptor the number of jobs in the source queue. The downward transition rate from state i to $i-1$, $i>0$, is given by $s+(2m-s)t_0$, where s is the mean number of servers captured by a source given that it is blocked, t_0 is the probability that a source has no jobs in its queues, given that one of the clusters that it can access is blocked. From symmetry considerations it can be seen that $s=m$. We have made the assumption that the transition rates are independent of the number of jobs in the queue.

Consider now the cluster process. There are three cases with respect to a cluster, determined by the state of the neighboring clusters, to be distinguished. The Markov chain corresponding to each case is different. The state descriptor that we use for the cluster process is the number of free (idle) servers,i, $0 \leq i \leq m$ in the cluster. The three cases correspond to the following situations with the probability of occurence of the case given in parenthesis:

- case 1: both neighboring clusters are busy (f_0^2) (fig. 2).
- case 2: exactly one neighboring cluster is busy ($2f_0(1-f_0)$) (fig. 3).
- case 3: no neighboring cluster is busy ($(1-f_0)^2$) (fig. 4).

Note that we assume independence of non-adjacent clusters; the probability of a busy cluster by our notation is f_0.

Let f_0^α, f_0^β and f_0^γ be the probabilities that the cluster is busy in cases 1,2 and 3 respectively. Then the total probability that a cluster is busy is given by

$$f_0 = f_0^2 f_0^\alpha + 2f_0(1-f_0)f_0^\beta + (1-f_0)^2 f_0^\gamma \qquad (3)$$

We now give an expression for t_0, the probability that a source has 0 jobs in its queue, given that a cluster that it can access is blocked.

Let us denote the two clusters that the source has access to by u and v. Let cluster u be busy. Consider now cluster v; if cluster v is free then t_0 must be 1. If cluster v is busy then t_0 equals q_0. Let $p_{v|u}$ be the probability that v is busy, given that u is busy. Then,

$$t_0 = (1-p_{v|u}) + p_{v|u}q_0 \qquad (4)$$

Cluster v has two clusters adjacent to it, one of which is u; let the other be w. Now the event, cluster w is busy, is independent of u being busy (by our independence assumption) and has a probability equal to f_0. Conditioning and unconditioning on the state of w we get

$$p_{v|u} = f_0 f_0^\gamma + (1-f_0)f_0^\beta \qquad (5)$$

We must find b, the probability that a source is blocked. Let the clusters accessible to the source under consideration be u and v.

$$b = Probability(v \ busy \ and \ u \ busy)$$

$$b = p_{v|u} \cdot f_0 \qquad (6)$$

The solution to the q_is is of the same form as the steady state number in system distribution of an M/M/1 queue [4]. ρ, the utilization in an M/M/1 system, equals $\lambda/m(1+t_0)$ in this case. Thus,

$$q_0 = 1-\rho \qquad (7)$$

$$W = b\rho/(\lambda(1-\rho)) \qquad (8)$$

The solution for the cluster processes, f_0^α, f_0^β and f_0^γ can easily be derived (in terms of λ, m and q_0).

We have a set of non linear simultaneous equations. We proceed to solve them by guessing initial values for f_0 and q_0 and iterating over the equations. Once we have the values for t_0 and b, W can be determined using equation 8. We have not investigated here the convexity or the rate of convergence.

3.3 Results

In fig. 5(a) and (b), the mean waiting time predicted by the analytic model is compared with the simulation results and the no load sharing case. In our simulations the 95 percent confidence interval relative width is less than 7 percent. There are 2 servers in a cluster in fig. 5. The no load sharing case thus corresponds to an M/M/2 queueing discipline.

Our analytic model derived above gives a reasonably good estimate of the mean wait time at low to high utilizations (fig. 5). At very high utilizations, however, the model is not as accurate. We suppose this is primarily because assumptions made in our approximations become untenable at very high utilizations; for example the ripple effect across clusters which we have neglected in clusters not adjacent to each other, is more pronounced and the effects of a remote cluster on a given cluster cannot now be neglected.

Our simulations and models show that performance is reasonably insensitive to the arrival and service protocols followed for given information complexity.

4 Topological Excursions

The ring topology is only one of a panoply of limited access topologies possible. In this section we investigate other topologies that we describe below.

4.1 The Mesh

Consider a toroidal mesh, with a source at each node of this mesh. A cluster of m servers exists between every pair of sources adjacent to each other on this mesh (fig. 6). More formally let the source set S be

$$S = \{s_{i,j} | 0 \leq i, j \leq n - 1\}$$

where $N = n^2$ is the total number of sources in the mesh. Then $s_{i,j}$ shares a different cluster (of m servers) with each of

$$s_{i,(j-1)modn}, s_{i,(j+1)modn}, s_{(i-1)modn,j}, s_{(i+1)modn,j}$$

Here too, like in the ring topology, each cluster is accessible by exactly two sources. However unlike in the ring, in a mesh each source can access four clusters.

4.2 The Modified Ring

The modified ring topology ($MR - d$) (fig. 7), has an additional parameter d, and is defined as follows: There are two types of sources — the 'direct' and the 'alternate' sources. The direct source set B comprises the direct sources $b_0 \ldots b_{n-1}$. The alternate source set A comprises the alternates sources $a_0 \ldots a_{n-1}$. N, the total number of sources in the system equals $2n$ while the total number of clusters is n. Let the clusters be $c_0, \ldots, c_{n-1}$. Each direct source $b_i, 0 \leq i \leq n-1$ can access clusters c_i and $c_{(i+1)modn}$. Each alternate source $a_i, 0 \leq i \leq n-1$ can access clusters c_i and $c_{(i+d)modn}$. In the modified ring system each cluster can be accessed by four sources while each source has access to two clusters only.

4.3 The Quad

Finally, we describe the quad topology (fig. 8). The quad topology has N sources $s_0 \ldots s_{n-1}$ and N clusters $c_0 \ldots c_{N-1}$. Source s_i has access to the clusters

$$c_i, c_{(i+1)modN}, c_{(i+2)modN}, c_{(i+3)modN}$$

Each source has access to four clusters and each cluster is accessible to four sources. In general a quad topology has three parameters and is denoted by quad-j,k,l. In a quad-j,k,l topology, source s_i can access clusters

$$c_i, c_{(i+j)modN}, c_{(i+k)modN}, c_{(i+l)modN}$$

The quad topology described previously is the quad-1,2,3; we drop the 1,2,3 and we use quad as a synonym for the quad-1,2,3 topology. For other quad topologies the parameters are explicitly given.

4.4 Interpreting The Results

The independence assumptions given in the ring model do not seem to hold here since the neighbours of a cluster may serve the same source. Thus we determine the mean waiting times through simulation and plot the mean waiting time against utilization for different topologies (fig. 9). Utilization, ρ, $= \lambda/(sd/rd)$. The sources have the same arrival rate λ and the service rate of every server is 1. Though the topologies are vastly different the performance does not seem to reflect this. We seek to identify parameters of the topology, that have an important bearing on performance.

Our first parameter is one we have already encountered — the source degree(sd). The source degree in itself seems sufficient to explain our results, if one postulates that the greater the source degree the better the performance — a rather intuitive hypothesis.

However the source degree is not a complete litmus of performance. This can be seen in fig. 10 where it can be seen that the mean waiting time of the M/M/8 system ($sd = 8$) is significantly higher than the MR-2 system which has $sd = 8$. The ring system shown has $sd = 4$ and the M/M/2 system has $sd = 2$.

We now extract another topological parameter which we call 'diversity'. A source x is a neighbour of a source y, if both x and y have access to some server s. Let n_s be the number of neighbors of source s that have access to at least one server that is not accessible to the given source s. We define the diversity of the topology to be the ratio of n_s to rd, noticing that n_s is the same for all sources in the network and rd is the resource degree.

The diversities for the different topologies are: M/M/m : 0, ring : 1, mesh : 2, modified ring : 1.5, quad-1,2,3 : 1.5.

Intuitively, the more the diversity the better the performance. The marginal benefit in performance due to diversity seems to drop off rapidly at low to moderate utilizations. At high utilizations increasing diversity results in improved performance. The performance difference between a non-zero diversity and zero diversity topology however, is markedly in favor of the former.

Reexamining our results in the light thrown by these two parameters we speculateas follows:

Conjecture : In any regular topology, mean wait time for a job, at a given utilization is influenced by two factors. The primary factor is the source degree — increasing the source degree improves performance. The other factor is the diversity — going from zero diversity to a non-zero diversity improves performance but increasing diversity once we have non-zero diversity gives marginal gains visible only at high utilizations.

A corollary is that two topologies with the same source degree and with non zero diversities must have comparable performances especially when the utilization is not very high.

The ring topology can be used as a template to create arbitrary *sd* and *rd* values and a non-zero diversity. Let us assume we want to determine the mean waiting time in a regular, non-zero diversity topology with $sd = n$ and at a utilization ρ. By our conjecture the performance is comparable to that of a ring topology with $n/2$ servers per cluster at the same utilization. The analytical model we have presented to solve for the approximate mean waiting time in the ring topology in section 3 can then be used to get an estimate of the mean waiting time.

Based on the technique outlined above we analytically derive an estimate for the mean waiting time in a quad-1,3,5 topology by evaluating the equivalent ring (fig. 11). The quad-1,3,5 topology we are considering has 4 servers per cluster, giving it an *sd* of 16, an *rd* of 4 and a diversity of 2.5. The ring system we use in our analytic solution has 8 servers per cluster which means that it has an *sd* of 16, an *rd* of 4 and unit diversity (different from 2.5 but not of great impact, by our conjecture).

Thus we see that a fair estimate of the mean waiting time can be derived for many regular limited access topology when the utilization factor is not extremely high, using an analytic model. The diversity of a ring topology being low, the estimate is usually an upper bound on the mean waiting time in the system.

5 Conclusion

We have investigated the determination of the mean waiting time in an arbitrary regular limited access topology and have attempted to provide reasonable and useful analytic and design guidelines. Our conjecture given in the previous section, states that the degree of resource sharing is the parameter with most performance impact. The diversity of a topology is in some sense indicative of the degree of load sharing and we conjecture that once there is some load sharing in the system (non-zero diversity) increasing diversity really does not help in improving performance as much as increasing the number of resources a source has access to. From a design standpoint, given a certain number of sources with fixed arrival rates and a fixed number of servers it would be best to try and increase the number of servers a source has access to; being very clever with the access topology does not really help too much, so long as the non-zero diversity (non M/M/m structure) threshold has been crossed. We have also given a technique to analytically estimate the mean waiting time in many limited access topologies by constructing the equivalent ring.

It would be interesting to consider load sharing schemes in a LAN environment with probe limits, like in [1], from the standpoint of limited access systems. The important difference being that the servers accessible to a source are chosen at random, dynamically, rather than being predetermined. The impact of the information complexities of the arrival and service protocols is also not fully understood.

Acknowledgements

We would like to thank the referees for their suggestions and Sathvik Krishnamurthy for his help.

References

[1] D.L. Eager, E.D. Lazowska and J. Zahorjan, "Dynamic Load Sharing In Homogenous Distributed Systems", 84-10-01, Dept. Of Computer Science, Univ. Of Washington.

[2] J.J. Green, "Load Balancing Algorithms In A Distributed Processing Environment", Ph.D. Dissertation, Dept. Of Computer Science, UCLA, 1988.

[3] F.P. Kelly, *Reversibility And Stochastic Networks*, J. Wiley, 1979.

[4] L. Kleinrock, *Queueing Systems. Vol. 1, Theory.*, J.Wiley, 1975.

[5] R.J.T. Morris and Y.T. Wang, "Some Results For Multi-Queue Systems With Multiple Cyclic Servers", *Proc. 2nd Symp. Perform. Comput. Commun. Syst.*, Zurich, Switz., Mar. 21-23, 1984.

[6] Y.T. Wang and R.J.T. Morris, "Load Sharing In Distributed Systems", *IEEE Trans. Comp.*, C-34(3), pp204-216.

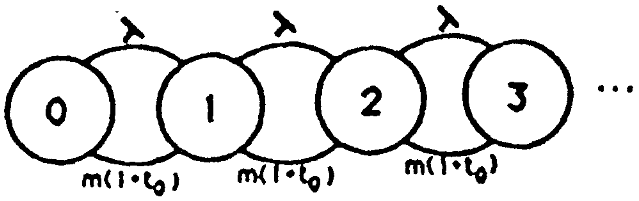


fig. 1: The Source Queue Markov Chain

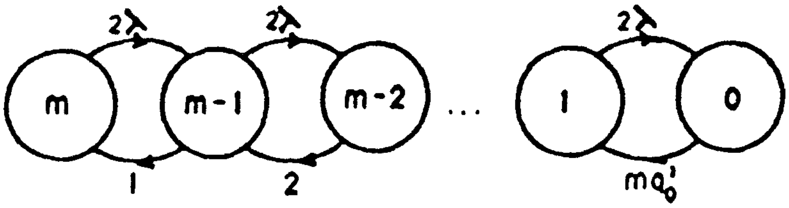


fig. 2 : The Cluster Markov Chain -- Case 1.
(Both Neighbors Busy)

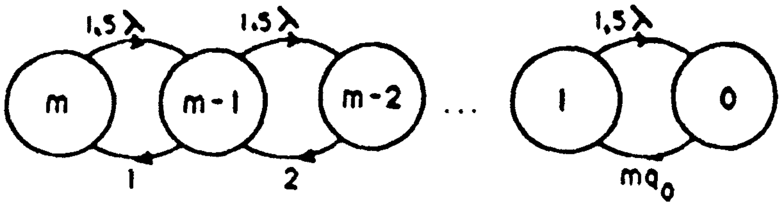


fig. 3 : The Cluster Markov Chain -- Case 2.
(Exactly One Neighbor Busy)

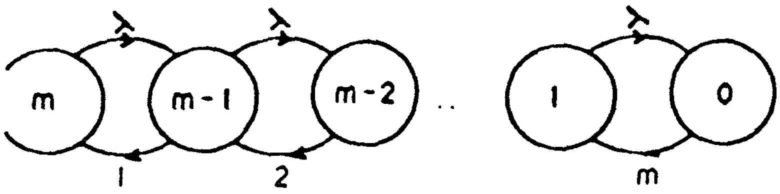


fig. 4 : The Cluster Markov Chain -- Case 3.
(No Neighboring Clusters Are Busy)

fig. 5(a) : Modeling The Ring Topology

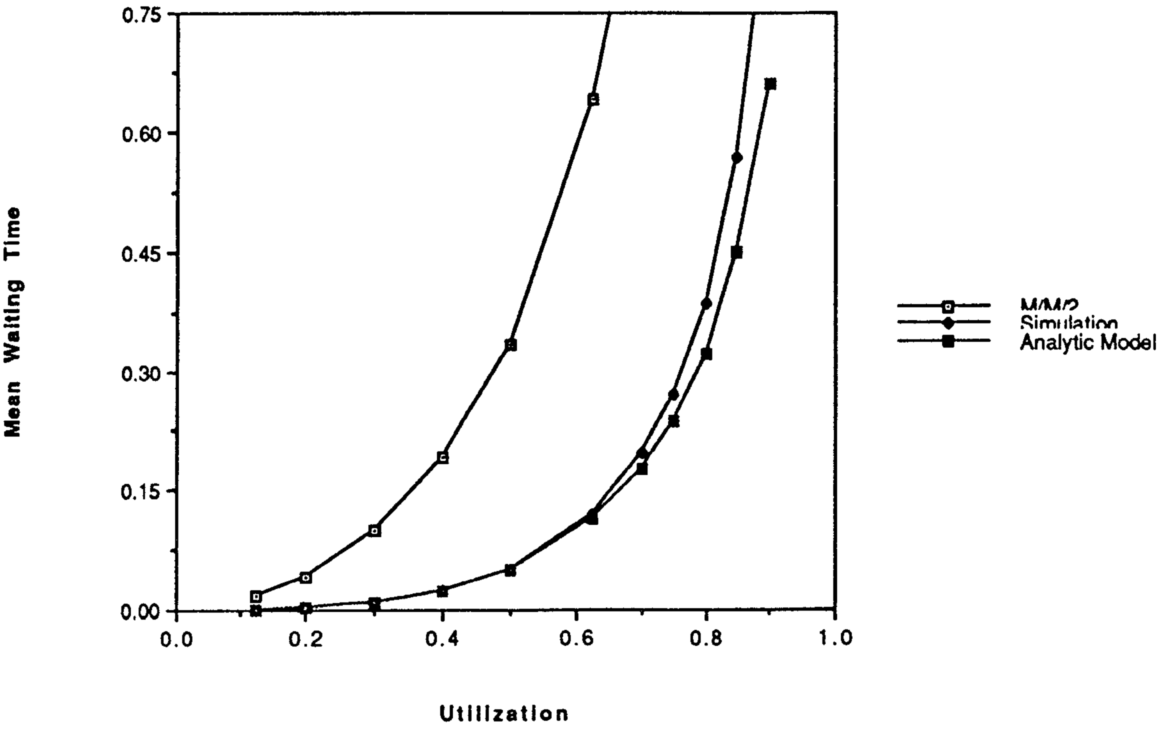
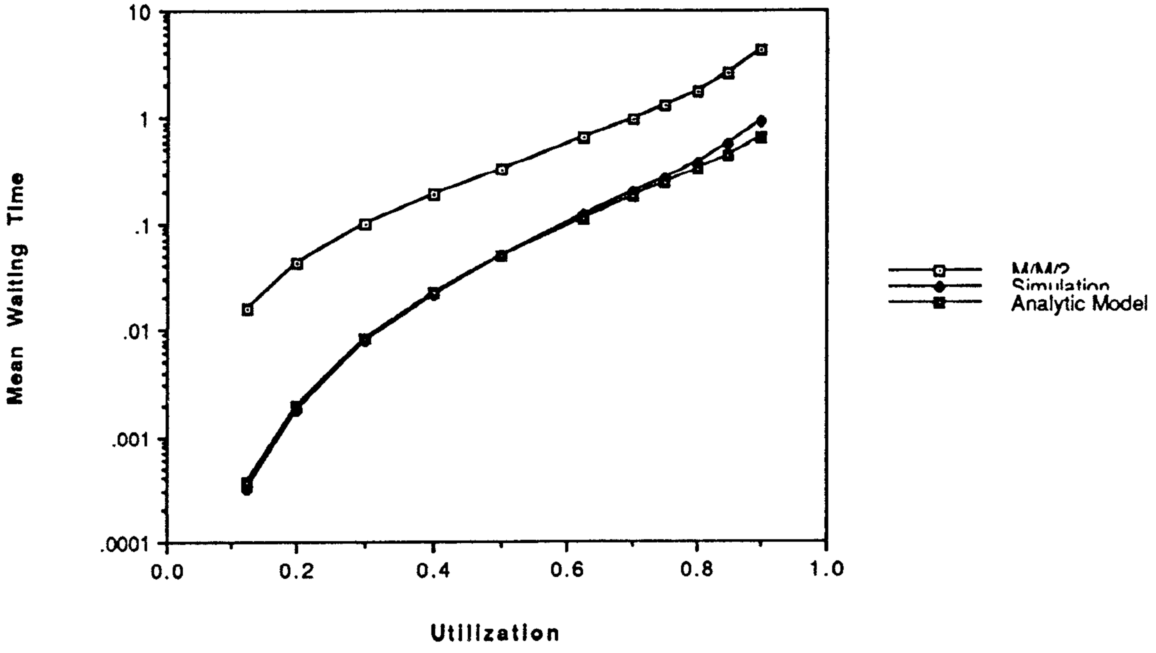


fig. 5(b): Modeling The Ring Topology (log scale)



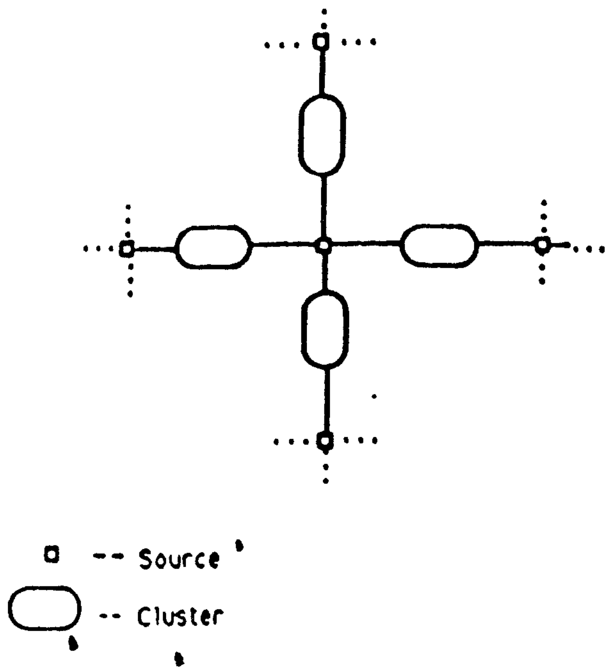


fig. 6 : The Mesh Topology.

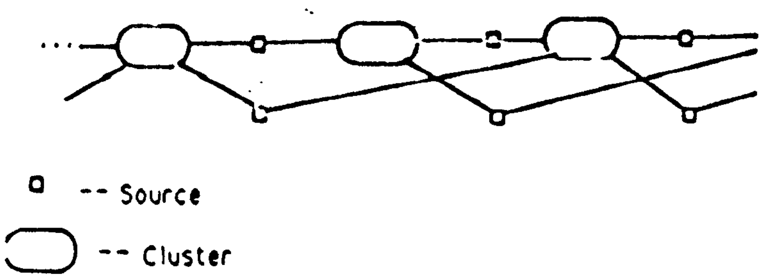


fig. 7 : The MR-2 Topology.

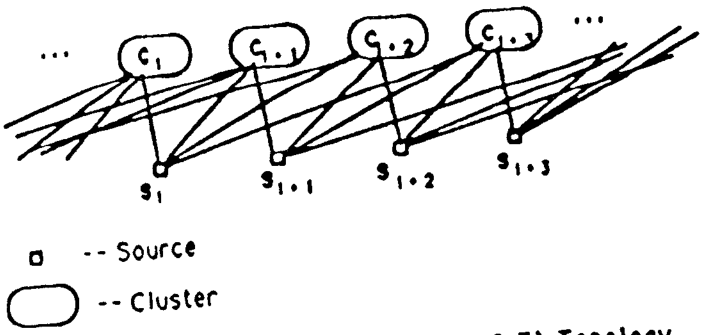


fig. 8 : The Quad (quad-1,2,3) Topology.

fig. 9(a) : Comparing Performance In Different Topologies

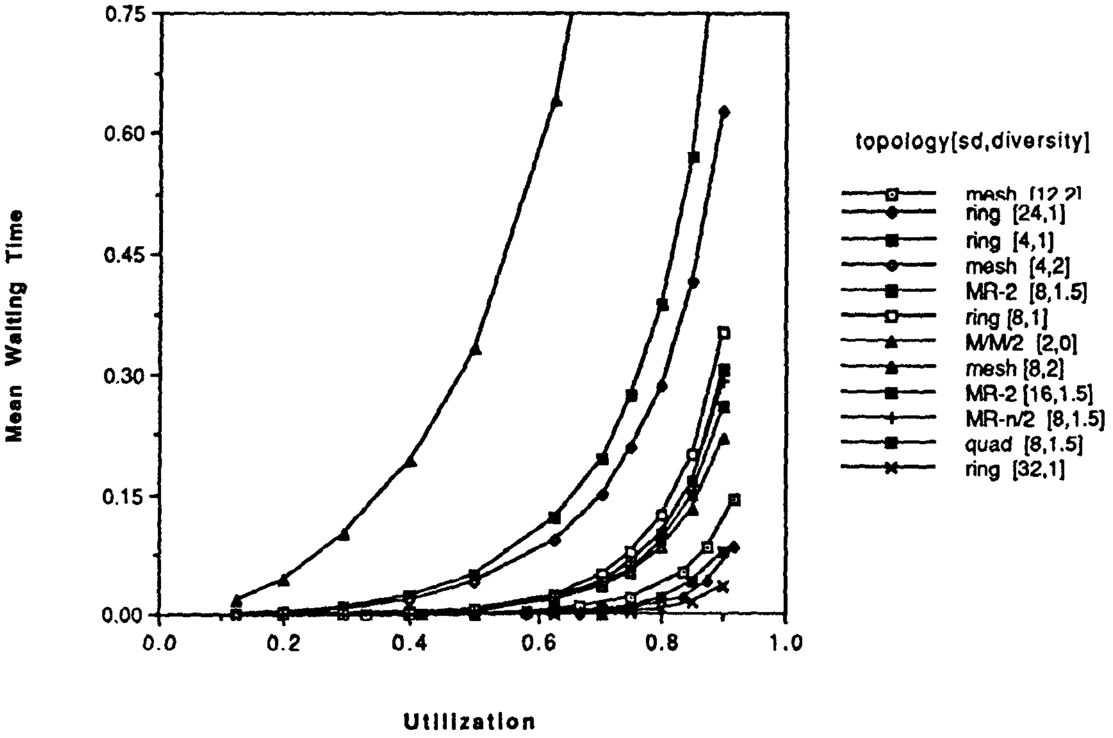


fig. 9(b) : Comparing Performance In Different Topologies (log scale)

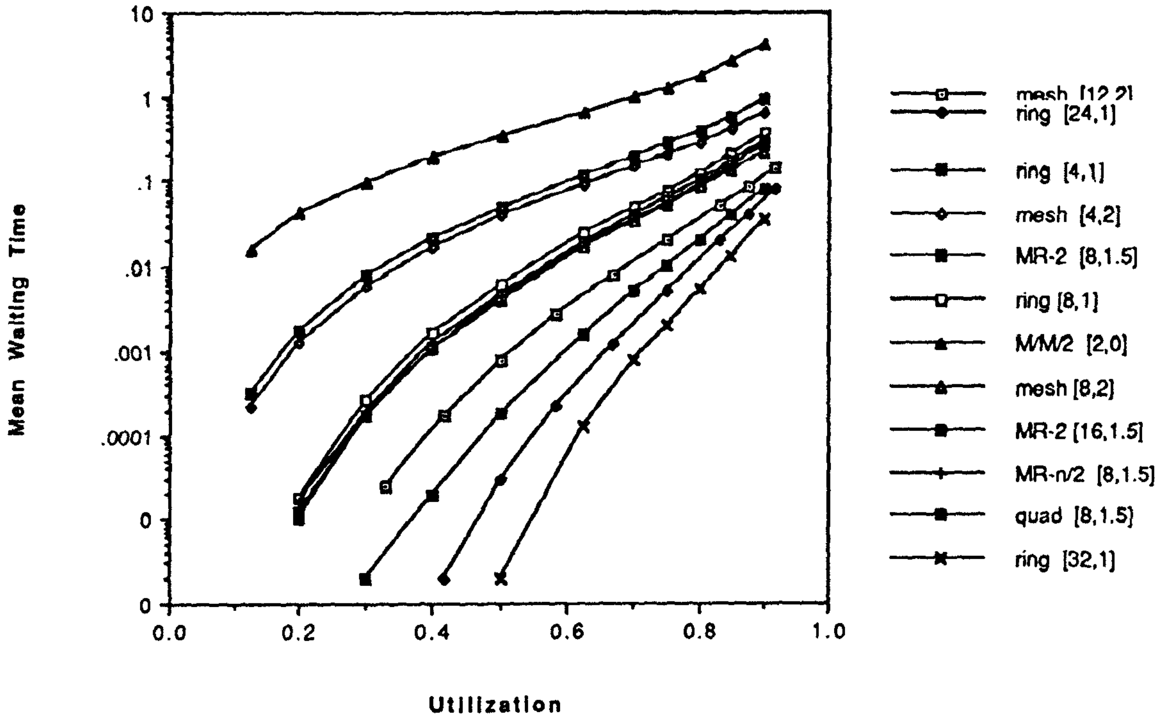


fig. 10 : Source Degree Alone Is Not Enough

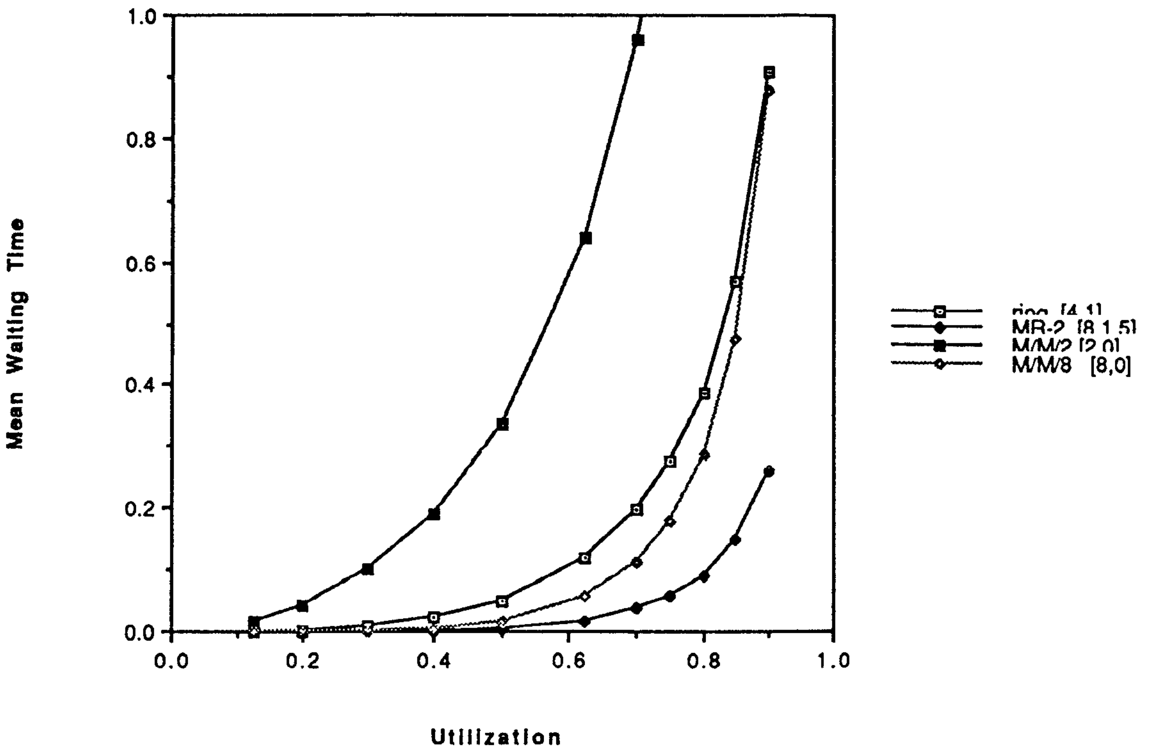
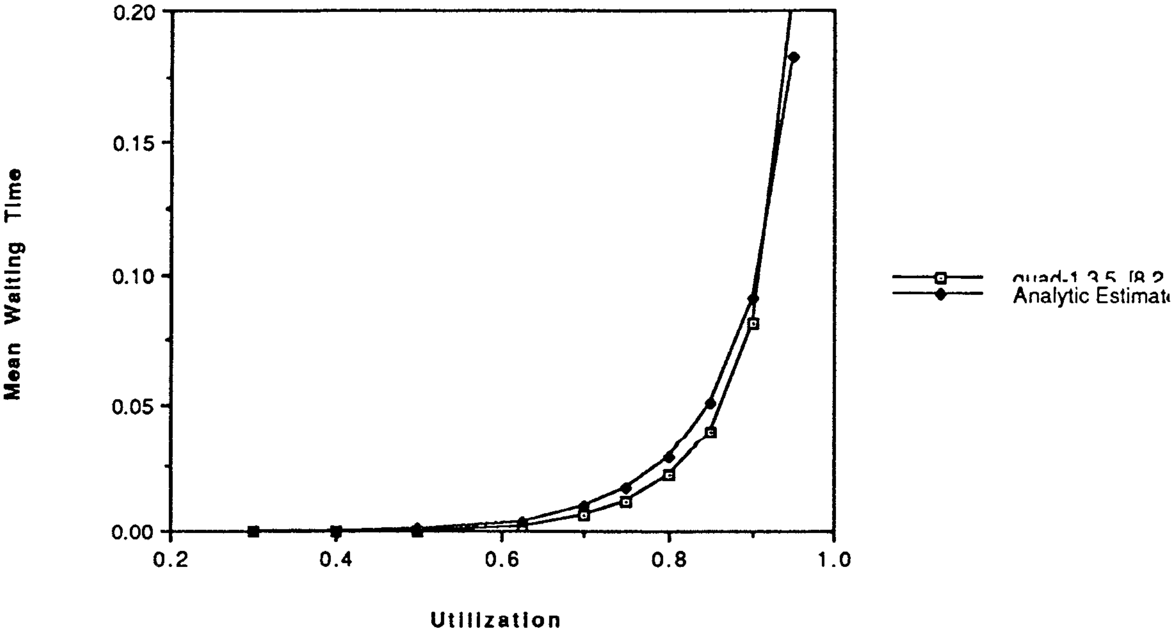


fig. 11: Analytically Estimating Performance


Mean Waiting Time
Utilization
quad-1.3.5 [8,2
Analytic Estimat