AN ASYMPTOTIC ANALYSIS OF A THRESHOLD LOAD BALANCING POLICY

Kyoo Jeong Lee
GTE Laboratories Incorporated
40 Sylvan Road
Waltham, MA 02254

Don Towsley
Department of Computer and Information Science
University of Massachusetts
Amherst, MA 01003

COINS Technical Report 87-112
July 15, 1987

AN ASYMPTOTIC ANALYSIS OF A THRESHOLD LOAD BALANCING POLICY*

Kyoo Jeong Lee†
GTE Laboratories Incorporated
40 Sylvan Road
Waltham, MA 02254

Don Towsley
Department of Computer and Information Science
University of Massachusetts
Amherst, MA 01003

July 15, 1987

Abstract

This paper describes and analyzes a specific threshold load balancing policy in a distributed computer system that executes a priority scheduling on jobs according to the location of job origination. An approximate analysis is carried out to obtain the response time performance of the system under the load balancing policy. In the analysis we assume that the arrival of jobs at a node transferred from other nodes are governed by a Poisson process. This allows us to decompose the behavior of the system into separate models of each of the nodes. We then map the behavior of each node into the framework of queueing systems subject to breakdown to obtain a closed-form expression for the mean response time of a job. We prove that the Poisson assumption on the job transfers is asymptotically exact and hence the performance predictions of the model is asymptotically exact as the number of nodes in the system increases. Simulation studies reveal that the Poisson job transfer assumption is good even for small systems when we are interested in obtaining the response time performance of the system.

*This work was partially supported by the National Science Foundation under grant number ECS-8406402.
†This work was performed while Kyoo Jeong Lee was at the University of Massachusetts.

I. Introduction

A number of threshold-based scheduling policies for distributed computer systems has been analyzed mathematically in recent years [4] [5] [10] [11] [12] [13]. Each of these policies was approximately analyzed by modeling the system consisting of N nodes as N *independent subsystems* under the assumption of Poisson external job arrivals and exponential service times. This decomposion has required the additional assumption that the arrival of jobs transferred from other nodes at each node is also governed by a Poisson process. The conjecture has been made in several of these studies that this last assumption is true in the limit as the number of nodes goes to infinity and that the analysis becomes asymptotically exact. Results from simulation experiments have been provided to support this conjecture.

In this paper we propose a new threshold load balancing policy for which we provide a similar approximate analysis based on decomposition. However, unlike previous studies, we prove that the assumption that job transfers from other nodes arrive according to a Poisson process is true in the limit as the number of nodes goes to infinity. Simulation results provide evidence that the convergence is rapid and that the model is accurate for systems containing as few as five nodes. As this policy does not differ significantly from earlier policies, we believe that the existence of limit theorems for this policy provides further evidence that this type of approximate analysis is accurate for other policies.

Besides the existence of limit theorems, the load balancing policy is of interest for two other reasons. First, the policy gives priority to jobs that are executed at the node at which they originate. This may be a desirable property for a policy implemented on a distributed system consisting of computers belonging to distinct independent entities. Second, unlike the previous studies, the analysis technique yields a closed-form solution for the mean response time of a job. This makes the determination of optimal load balancing parameters much easier. Such an optimization has been studied in [12].

We state and prove limit theorems for two types of systems. In the first system all nodes are equal partners and can send jobs to any other node (according to some

distribution). We refer to this as the *peer* system. Most previous load balancing studies consider this system where all nodes are identical. We prove a limit theorem that states that, as the number of nodes increases, the performance predictions of the model becomes exact.

In addition to this system, we also consider a second system that receives little treatment in the literature but is in widespread use. This system which we refer to as the *star* system, contains one special node, the *central node* that may contain a significant amount of computing power. We shall refer to the remaining nodes as *leaf nodes.* Although jobs may be transferred between any pair of nodes, the central node is typically a prefered node for job transfers from the leaf nodes. In addition, this node may call on the leaf nodes to perform some of its tasks. For this system we state and prove a limit theorem where the analysis of each node becomes exact as the number of leaf nodes goes to infinity. This provides further evidence that the same modeling technique can be used with other policies operating on star systems.

Section II contains a description of the system and the proposed threshold policy. Section III contains an approximate analysis of a distributed computer system operating under this policy. Section IV includes the limit theorems that indicate that the analysis is accurate for large systems. In this section we will also describe the differences between the new priority policy and earlier policies that make ours easier to analyze. Section V provides some numerical results that show that the approximation is good even for systems containing as few as five nodes. We summarize the results of the paper in section V.

II. System Model and Load Balancing Policy

In this section we describe the system model and the load balancing policy that we study.

II.1 Model Description

The system consists of a number of autonomous host computers interconnected by a communication network (see Figure 1). The communication network can be either a local area network or a store-and-forward network. Specifically, we consider two system topologies. The first one is a *peer* system where all nodes are peers. In this case it is assumed that jobs arriving at a node can be processed either locally or at any other node in the system after being transferred through the communication network. On the other hand, in the second topology, one of the nodes is considered as the *central node* in the system and the remainder as *leaf nodes*. We refer to this system as the *star* system. Here, jobs arriving at leaf nodes can be processed either locally or at the central node. However, there is no workload sharing among leaf nodes. This may be due to software/hardware restrictions among leaf nodes. Jobs arriving at the central node can be processed either locally or at any other leaf node in the system. One variation on this star system would prohibit the central node from sending its jobs to leaf nodes. This is a special case of the star system described above and the results obtained in this paper can be readily applied to that system.

We assume that jobs arrive at each node according to a Poisson process with rate ϕ_i, $i = 0, 1, 2, ..., N$ where $N + 1$ is the number of nodes in the system (node 0 denotes the central node when the star topology is considered). The external workload and/or processing power of each node may differ from each other. An example of a peer system is an interactive transaction processing system consisting of multiple computers (*e.g.*, collection of Service Control Points for database query processing in Common Channel Signaling networks [6]). A distibuted system consisting of a mainframe and a number of workstations

interconnected by a local area network is an example of a star system topology.

If a job is chosen for remote processing, it is transferred from the source (origin) node to a processing (destination) node and the results are returned to the source node. Communication delays consisting of packetization, unpacketization, transmission and queueing delays are incurred during both transfers. We assume that each node contains an off-load processor (communication server) that takes care of job transfer between nodes. Consequently, the node processor is not affected by the job transfer between nodes. For the sake of simplicity, the communication delay is accounted for only at the communication network.

Specifically, we model each node as a single-server queueing system (*i.e.*, all resources and queues in a node are lumped into a single-server model) having a mean service time of $1/\mu_i$, $i = 0, 1, 2, ..., N$. We also model the communication network as a single-server queueing system having a mean service time of $1/\mu_{ch}$. The effect of more complex node models is discussed in [12].

II.2 Load Balancing Policy

The load balancing policy studied in this paper is a *sender-initiated* policy in the sense that the sending node makes decisions for job transfer. Jobs within the system are divided into two classes; namely, *local jobs* and *remote jobs*. Local jobs are those processed at the node of origination and remote jobs are those processed at some other node in the system after being transferred through the communication network. Let $L_i^{(l)}$, $L_i^{(r)}$ and L_i be the random variables denoting the number of local jobs, the number of remote jobs and the number of jobs at node i respectively ($L_i = L_i^{(l)} + L_i^{(r)}$). In the following we describe the load balancing policy executed at node i in a peer system. When the system is of star topology, the only difference is that leaf nodes can transfer jobs only to the central node.

Policy SLO (Sender-initiated LOcal)

- If node i receives a job from the external world with $L_i^{(l)} \geq T_i$, the node sends the

job to node j in the system with probability P_{ij} where $\sum_{k=0,k\neq i}^{N} P_{ik} = 1$ ($P_{ii} = 0$). Otherwise, it processes the job locally.

- Jobs arriving from other nodes are always accepted at the destination node. Hence, jobs can be transferred at most one time.
- Each node schedules jobs on the processor according to a preemptive priority discipline where local jobs are given a higher priority than remote jobs.

Job flows at node i are shown in Figure 2. In the figure, Δ_i denotes the rate at which jobs are transferred from node i to other nodes, γ_i denotes the rate at which jobs arrive from other nodes, and β_i denotes the local job throughput. The parameters T_i's and P_{ij}'s need to be determined to obtain good system performance. The problem of how to determine values for these parameters is beyond the scope of this paper and is discussed elsewhere [12]. This paper only focusses on the analysis of the system for given parameter values. Note that when $T_i \rightarrow \infty$ for all i, it corresponds to the system without load balancing.

Jobs can be processed by any scheduling algorithm (*e.g.*, First-Come-First-Served, Last-Come-First-Served, Processor Sharing) if they have the same priority. Policy SLO favors local jobs over remote jobs by assigning a higher priority (*i.e.*, it is a *selfish* policy). Consequently, remote jobs from other nodes experience, on the average, longer delays than local jobs. We can have different priority assignment rules in order to obtain different performance characteristics and to meet different design goals. These topics are discussed in [11]. Policy SLO uses the number of local jobs as a workload indicator. Policies using other workload indicators are discussed in [4] [5] [12].

Policy SLO is a highly decentralized control policy in the sense that the job transfer decision is solely based on the local state information. Hence it is possible that an arriving job from other node may find a busy destination node. In this case, this job transfer probably does not improve the response time performance of the job. However, this is the price one pays for using a decentralized algorithm that makes use only of local state

information. In order to avoid this undesirable situation, the source node may probe possible destination nodes to see whether they are busy or not. It then sends the job to a node which is not busy. This class of policies requiring nonlocal state information has been studied in [4] [5] [12] [13]. However, it has been shown that the peformance gain obtained by using probes diminishes either when the communication delay is large (in this case probing information is outdated) [13] or when each node provides sufficient concurrent processing and allows multiprogramming [12].

III. Analysis of the SLO Policy

In this section we analyze the behavior of the system under Policy SLO. This is done by assuming that the remote jobs arrive at each node according to Poisson processes. Thus the model of the system can be decomposed into independent models of each node and the communication network. We then obtain the mean response time of a job in the system by mapping the behavior of each node into the framework of queueing systems subject to breakdown. The Poisson assumption on remote job arrivals will be shown to be exact under certain limiting conditions in the next section.

When the communication delay is negligible, one means of describing the behavior of the system is to use a Markov chain with a state defined as $(L_0^{(l)}, L_0^{(r)}, L_1^{(l)}, L_1^{(r)},..., L_N^{(l)}, L_N^{(r)})$. More variables are required in the state when the communication delay is not negligible. In either case, the Markov chains are not amenable to simple, efficient solution. To circumvent this problem, we make the assumption that the remote jobs arrive at each node according to Poisson processes. Using this assumption, we can now view each node as a queueing system having two kinds of Poisson arrivals; namely, external job arrivals with rate ϕ_i and remote job arrivals from other nodes with rate γ_i. Figure 3 illustrates the Markov chain describing the behavior of node i under Policy SLO when $T_i = 4$. The state is defined as $(L_i^{(l)}, L_i^{(r)})$. Note that we cannot use this decomposition technique to obtain joint statistics for two or more nodes since the remote job arrivals at each node may not be independent of each other.

External job arrivals are subject to remote processing according to the threshold-based decisions. Since local jobs are given a higher priority than remote jobs, they do not experience any delay by remote jobs. Hence jobs overflow at node i for remote processing in the same way as they do in M/M/1/T_i queueing systems. Consequently we have

$$\Delta_i = \phi_i P[L_i^{(l)} \geq T_i] = \frac{\phi_i(1-u_i)u_i^{T_i}}{(1-u_i^{T_i+1})}, \tag{1}$$

where $u_i = \phi_i/\mu_i$. This overflow process has been proven to be a renewal process by Cinlar

and Disney [2]. Remote job arrivals at each node are related to the job overflows according to

$$\gamma_i = \sum_{k=0, k\neq i}^{N} \Delta_k P_{ki}. \tag{2}$$

Let D be a random variable denoting the response time of a job in the system. From Little's results [9] we obtain the mean response time of a job as follows.

$$E[D] = \frac{\sum_{i=1}^{N} E[L_i] + E[L_{ch}]}{\sum_{i=1}^{N} \phi_i}, \tag{3}$$

where $E[.]$ is the expectation operator and L_{ch} is the random variable denoting the number of jobs in transition in the communication network. $E[L_{ch}]$ can be readily obtained from the M/M/1 formula [9] as,

$$E[L_{ch}] = 2 \ \frac{2\sum_{i=1}^{N} \gamma_i}{\mu_{ch} - 2\sum_{i=1}^{N} \gamma_i}, \tag{4}$$

where the factor 2 is due to the round trip delays of a job processed remotely. On the other hand, $E[L_i]$ can be obtained by solving the Markov chain shown in Figure 3. We can use either a matrix-geometric formulation [14] [15] or a partial generating function method (complex variable analysis) to solve this Markov chain. However, neither method provides a simple closed-form solution for the mean response time of a job.

In order to obtain $E[L_i]$ in closed-form, we obtain $E[L_i^{(l)}]$ and $E[L_i^{(r)}]$ in turn where,

$$E[L_i] = E[L_i^{(l)}] + E[L_i^{(r)}]. \tag{5}$$

The behavior of local jobs at node i is identical to that in M/M/1/T_i queueing systems since local jobs do not experience any delay by remote jobs. From the M/M/1/T_i formula [9], we obtain,

$$E[L_i^{(l)}] = \frac{u_i\{1 - (T_i + 1)u_i^{T_i} + T_i u_i^{T_i+1}\}}{(1 - u_i)(1 - u_i^{T_i+1})}. \tag{6}$$

To study the behavior of remote jobs, we use results from Avi-Itzhak and Naor [1] which deals with a single-server queueing system subject to breakdown. They considered a queueing station in which the server can be in one of two states, active or inoperative. Upon breaking down, the server becomes inoperative for a random period of time (repair time) after which it returns to its normal state of activity. They derived a closed-form expression for the mean queue length of this queueing system under the following assumptions: 1) jobs arrive to the system according to a Poisson process, 2) the service time of a job is an arbitrarily distributed random variable having a finite second moment, 3) the time between the repair of a breakdown and the subsequent breakdown is an exponential random variable, and 4) the repair time is an arbitrarily distributed random variable having a finite second moment.

In order to map our problem into the framework of queueing system subject to breakdown, we make the following observations.

- A remote job sees an inoperative server whenever there is at least one local job, since local jobs are given a higher priority than remote jobs.
- Remote jobs arrive at each node according to a Poisson process with rate γ_i, $i = 0, 1, 2, ..., N$.
- The duration of uninterrupted availability experienced by remote jobs is determined by the interarrival time of external jobs which is an exponentially distributed random variable with mean $1/\phi_i$, $i = 0, 1, 2, ..., N$.
- The repair time is the same as the busy period of node by local jobs (*i.e.*, busy period of M/M/1/T_i queueing system).

We define B_{T_i} as a random variable denoting the busy period of M/M/1/T_i queue. The following expression for the mean remote job queue length can be taken from [1] after allowing for exponential service times,

$$E[L_i^{(r)}] = \frac{\gamma_i}{\mu_i P_i^{(0)}} + \frac{\gamma_i E[B_{T_i}](C_{B_{T_i}}^2 + 1)P_i^{(0)}P_i^{(1)} + 2(\gamma_i/\mu_i)^2/P_i^{(0)}}{2(P_i^{(0)} - \gamma_i/\mu_i)}, \tag{7}$$

where

$$P_i^{(0)} = P[L_i^{(l)} = 0] = \frac{1 - u_i}{1 - u_i^{T_i+1}},$$

$$P_i^{(1)} = P[L_i^{(l)} \geq 1] = 1 - P_i^{(0)},$$

$$C_{B_{T_i}}^2 = \frac{E[B_{T_i}^2] - E^2[B_{T_i}]}{E^2[B_{T_i}]}.$$

Note that $C_{B_{T_i}}^2$ is the coefficient of variation for B_{T_i}. The following expressions for $E[B_{T_i}]$ and $E[B_{T_i}^2]$,

$$E[B_{T_i}] = \frac{1 - (\phi_i/\mu)^{T_i}}{\mu_i - \phi_i},$$

$$E[B_{T_i}^2] = \frac{2\{\mu_i - \phi_i(\phi_i/\mu_i)^{2T_i}\}}{(\mu_i - \phi_i)^3} - \frac{2(1 + 2T_i)(\phi_i/\mu_i)_i^T}{(\mu_i - \phi_i)^2},$$

have been derived in the Appendix.

IV. Limit Theorems

In this section we show that the Poisson assumption on remote job arrivals made in the previous section becomes exact under certain limiting conditions. In order to show this we use limit results regarding the independent thinning and superposition of random processes in the literature.

Independent thinning of a point process with probability p, $0 < p < 1$, is defined as follows; a point is retained with probability p and deleted with probability $1 - p$, independently for each point. Note that job transfers from node k to node i is a *thinned* process of job overflows at node k with probability P_{ki}. In peer systems, remote job arrivals at each node are a superposition of thinned processes of job overflows from other nodes whose rate is determined by equation (2). On the other hand, in star systems, remote job arrivals at the central node is a superposition of job overflows from the leaf nodes whereas remote job arrivals at each leaf node are a thinned process of job overflows from the central node.

We have the following lemma regarding the superposition of renewal processes [7]. A more general statement can be found in [8].

Lemma 1: Suppose a random process represents the superposition of N independent renewal processes. Let λ_i denote the rate of the ith stream. Then the superposition process becomes a Poisson process with rate $\lambda = \sum_{i=1}^{N} \lambda_i$ in the limit as $N \to \infty$ and λ_i's, $i = 1, 2, ..., N$, all tend to zero while $\sum_{i=1}^{N} \lambda_i$ remains constant.

This result is similar, in spirit, to the central limit theorem. Discussions on the statistical properties of superposition of a finite number of renewal processes can be found in [3].

In addition to this, we have the following lemma regarding the independent thinning of a point process [16].

Lemma 2: Let λ be the rate of a point process which undergoes an independent thinning with p. As $\lambda \to \infty$, $p \to 0$ and $p\lambda$ remains constant, the thinned process becomes a Poisson

process with rate $p\lambda$.

From these two lemmas, we derive the following theorems.

Theorem 1: In a peer system, remote job arrivals at node i become a Poisson process with rate γ_i in the limit as $N \to \infty$ and $\Delta_k P_{ki} \to 0$, $k \neq i$, while $\sum_{k=0,k\neq i}^{N} \Delta_k P_{ki}$ remains constant.

Proof: Directly from Lemma 1.

Theorem 2: In a star system remote job arrivals at the central node become a Poisson process with rate γ_0 in the limit as $N \to \infty$ and $\Delta_i \to 0$, $i = 1, 2, ..., N$, while $\sum_{i=1}^{N} \Delta_i$ remains constant.

Proof: Directly from Lemma 1.

Theorem 3: In a star system remote job arrivals at leaf node i become a Poisson process with rate γ_i in the limit as $N \to \infty$, $\Delta_0 \to \infty$, and $P_{0i} \to 0$ while $\Delta_0 P_{0i}$ remains constant.

Proof: Directly from Lemma 2.

Hence the Poisson remote job arrival assumption is good in large systems. Simulation results given in the next section reveal that this is a reasonable assumption even in small systems (*e.g.*, five nodes in the system) when we are interested in obtaining the mean job response time of the system.

The reason why it is easy to prove such limit theorems for our policy but not for others is that under Policy SLO the job overflow processes at each node are independent of each other. This allows us to apply Lemmas 1 and 2 directly. However, for other threshold policies studied in the literature [4] [5] [10] [11] [12] [13], the job overflow processes at each node are not independent of each other. We believe stronger results regarding the superposition of *dependent* renewal processes are required before asymptotic results will be developed for these other policies. Furthermore, we have observed that under Policy SLO

the mean response time of a job obtained by analysis converges to the simulation results faster than that under other policies [12] as the number of nodes increases. We believe that this is also due to the independence of the overflow processes.

Before we end this section, we present an application of the limit theorems.

Example: Consider a system that contains M distinct classes of nodes, $m = 1, \cdots, M$. We assume that there are N_m nodes in class m, and that all nodes in this class are identical *i.e.*, nodes in class m have the same job arrival rate ϕ_m, service rate μ_m, and transfer probabilities $Q_{m,n}$ where $Q_{m,n}$ is the probability that a job is transferred to a class n node given that it is transferred from a class m node. We assume that if a job is transferred to a class n node, then it is equally likely to be transferred to any class n node. In other words, if i is a class m node and j is a class n node, then $P_{i,j} = Q_{m,n}/(N_n - \delta_{m,n})$.[1]

If we allow $N_m \rightarrow \infty$ such that N_m/N_n remains unchanged, $n, m = 1, \cdots, M$ and the parameters ϕ_m, μ_m, and $Q_{m,n}$ are unchanged, then the hypothesis of Theorem 1 is satisfied. Thus the decomposition technique yields accurate results for heterogeneous systems where there are a large number of each type of node.

We conclude this section with the following conjecture. In the previous section we pointed out that the decomposition technique used in the analysis cannot be used to obtain joint statistics for two or more nodes since the remote job arrivals at each node may not be independent of each other. However, as the system becomes large, the correlation among remote job arrivals becomes weak.

Conjecture: In the limit as $N \rightarrow \infty$ and $\Delta_k P_{ki} \rightarrow 0$, $k \neq i$, while $\sum_{k=0, k\neq i}^{N} \Delta_k P_{ki}$ remains constant, the joint queue length distribution of the system can be expressed as the product of the marginal queue length distributions of all the nodes, *i.e.*, for $m_i = 0, 1, \cdots, T_i$, $n_i = 0, 1, \cdots$, and $i = 1, \cdots, N$,

$$\lim_{N\rightarrow\infty} P[L_0^{(l)} = m_0, L_0^{(r)} = n_0, \cdots, L_N^{(l)} = m_N, L_N^{(r)} = n_N]$$

[1] Here $\delta_{m,n}$ is the Kronecker delta which takes on value 1 if $n = m$ and 0 otherwise.

$$= \lim_{N \to \infty} \prod_{i=0}^{N} P[L_i^{(l)} = m_i, L_i^{(r)} = n_i].$$

V. Numerical Results

In this section we consider numerical examples to evaluate the performance gain obtained by Policy SLO over no load balancing. We also validate our Poisson remote job arrival assumption by comparing the mean response time of a job obtained analytically with that obtained by simulation.

We first consider a peer system consisting of five nodes where each node has the same external job arrival rate (ϕ) and job processing rate (μ). We define $u = \phi/\mu$. We obtain the values of the thresholds and transfer probabilities that yields the minimum mean response time of a job through an exhaustive search. In this homogeneous system where all nodes are identical, however, each node has the same threshold (denoted T) and $P_{ij} = 0.25$ for $i \neq j$ at the optimal solution. Hence we can search only over T in order to obtain the optimal solution.

Figure 4 shows the mean response time of a job under Policy SLO (denoted Case 1) as a function of T when $1/\mu = 1.0$, $1/\mu_{ch} = 0.01$ and $u = 0.8$. It also shows the mean response time of a job under no load balancing (denoted NLB). As T increases, the mean response time under approaches that of no load balancing. The optimal mean response time is obtained when $T = 2$. Case 2 corresponds to the mean response time when the communication network is slow ($1/\mu_{ch} = 1.0$). In this case the optimal threshold is large ($T = 6$) and the corresponding mean response time increases.

In Table 1 we compare the optimal mean response time under Policy SLO (denoted SLO) with that under no load balancing. Corresponding optimal thresholds are given in the parentheses. Simulation results for the mean response time of a job (denoted SIM) are also provided (point estimates along with 90% confidence intervals). The percentage error of the analytical predictions with respect to simulation results are given (denoted %). As we can see, the analytical predictions are slightly larger than the simulation results for a wide range of utilization. This is due to the fact that for this small system the actual remote job arrivals are burstier than Poisson process. As the number of nodes increases, this difference decreases. In all cases the percentage error is negligibly small. Note that the

performance under Policy SLO is increasingly better than that under no load balancing as the utilization of each node increases.

We next consider a star system where five leaf nodes have the same external workload (ϕ_s) and the same processing power (μ_s). We consider the case where $1/\mu_0 = 0.2$, $1/\mu_s = 1.0$ and $1/\mu_{ch} = 0.01$. We define $u_0 = \phi_0/\mu_0$ and $u_s = \phi_s/\mu_s$ respectively. In such a homogeneous system leaf nodes have the same threshold (denoted T_s) and $P_{0i} = 0.2$ for $i = 1, 2, .., 5$ at the optimal solution. Hence the optimal mean response time can be obtained by searching over T_0 and T_s. Table 2 compares the optimal mean response time under Policy SLO with that under no load balancing. It also presents simulation results along with percentage errors of analytical predictions. Simulation results show that the Poisson assumption is still good although the percentage errors are slightly larger than those in peer system case. Note that Policy SLO achieves a significant improvement in performance over no load balancing especially when the central node is lightly loaded.

In the above two examples we consider homogeneous systems only; all nodes are identical in peer systems and all leaf nodes are identical in star systems. This makes the determination of optimal load balancing parameters relatively simple. However, when the system is not homogeneous, we have to solve a nonlinear integer optimization problem in order to obtain the optimal mean response time of a job. This problem is much more difficult and is studied in [12].

VI. Conclusions

In this paper we studied a specific threshold load balancing policy in distributed computer systems that favors local jobs over remote jobs. A queueing model for the system was developed and an approximate analysis was carried out to obtain a closed-form expression for the mean response time of a job. In the analysis we assumed that the remote job arrivals at each node are Poisson processes. This allowed us to decompose the behavior of the system into the behavior of each of the nodes and the communication network. The mean response time of a job is then obtained by mapping the behavior of each node into the framework of queueing systems subject to breakdown. We showed that the Poisson assumption is exact and hence the performance predictions of the model is exact as the number of nodes in the system increases. Simulation results indicate that this assumption is good even for small systems containing several nodes when we are interested in obtaining the response time performance of the system. Numerical examples show that the load balancing policy achieves a significant improvement in performance over no load balancing.

Appendix

In this appendix we compute the mean and the coefficient of variation of the busy period of M/M/1/i queueing system that has job arrival rate ϕ and service rate μ. Let us define B_i as a random variable denoting the busy period of M/M/1/i queueing system and let $F_{B_i}(s)$ be the Laplace transform of the probability density function of B_i *i.e.*, $F_{B_i}(s) = E[e^{-sB_i}]$.

A busy period begins with an arrival of a job to an idle system. Once a busy period starts, the system can transit to one of two states as shown in Figure A.1. One possibility is completion of this busy period by the service completion of the job which initiates the busy period. This event occurs with probability $\mu/(\phi+\mu)$. In this case the busy period is an exponentially distributed random variable with mean $1/(\phi+\mu)$ since this period is the minimum of two exponentially distibuted random variables with rates ϕ and μ respectively. The other possibility is an arrival of another job before this job completes its service. This newly arrived job actually initiates another busy period of an M/M/1/$(i-1)$ system. After this new busy period (*i.e.*, B_{i-1}) ends, only one job remains in the original system. Since the system is memoryless (Markovian arrivals and Markovian server), the remaining job is considered to start a new busy period of M/M/1/i system. Therefore in this case the busy period is a summation of three random variables *i.e.*, an exponentially distributed random variable with mean $1/(\phi+\mu)$, busy period of M/M/1/$(i-1)$ system, and busy period of M/M/1/i system. This event occurs with probability $\phi/(\phi+\mu)$.

Hence we can write down the following recursive relation for the busy period of M/M/1/i system.

$$F_{B_i}(s) = \left(\frac{\mu}{\phi+\mu}\right)\left(\frac{\phi+\mu}{s+\phi+\mu}\right) + \left(\frac{\phi}{\phi+\mu}\right)\left(\frac{\phi+\mu}{s+\phi+\mu}\right) F_{B_{i-1}}(s)F_{B_i}(s). \qquad (A.1)$$

Therefore,

$$F_{B_i}(s) = \frac{\mu}{\phi+\mu+s-F_{B_{i-1}}(s)}. \qquad (A.2)$$

Using the moment generating property of Laplace transform of a probability density function, we obtain the following recursive relation for the first and second moments.

$$E[B_i] = \frac{1 + \phi E[B_{i-1}]}{\mu}, \tag{A.3}$$

$$E[B_i^2] = \frac{2\{1 + \phi E[B_{i-1}]\}^2}{\mu^2} + \frac{\phi E[B_{i-1}^2]}{\mu}, \tag{A.4}$$

with initial conditions $E[B_0] = 0$ and $E[B_0^2] = 0$. These first-order difference equations are easily solved to yield the following expressions.

$$E[B_i] = \frac{1 - (\phi/\mu)^i}{\mu - \phi}, \tag{A.5}$$

$$E[B_i^2] = \frac{2\{\mu - \phi(\phi/\mu)^{2i}\}}{(\mu - \phi)^3} - \frac{2(1 + 2i)(\phi/\mu)^i}{(\mu - \phi)^2}, \tag{A.6}$$

$$C_{B_i}^2 = \frac{E[B_i^2] - E^2[B_i]}{E^2[B_i]}. \tag{A.7}$$

References

[1] B. Avi-Itzhak and P. Naor, "Some queueing problems with the service station subject to breakdown," *Operations Research*, pp. 303-320, May-June 1963

[2] E. Cinlar and R.L. Disney, "Stream of overflows from a finite queue", *Operations Research*, pp.131-134, Feb. 1967

[3] D.R. Cox and W.L. Smith, "On the superposition of renewal processes", *Biometrika*, vol. 41, pp. 91-99, 1954

[4] D.L. Eager, E.D. Lazowska, and J. Zahorjan, "Adaptive load sharing in homogeneous distributed systems," *IEEE Trans. Software Eng.* vol. SE-12, pp. 662-675, May 1986

[5] D.L. Eager, E.D. Lazowska, and J. Zahorjan, "A comparison of receiver-initiated and sender-initiated adaptive load sharing," *Performance Eval.*, vol. 6, pp. 53-68, March 1986

[6] R. Hass and R. Robrock, "The intelligent network of the future," Proceedings of Globecom, pp. 1311-1315, 1986

[7] S. Karlin and H.M. Taylor, A First Course in Stochastic Processes, Second Edition, Academic Press, 1975

[8] A.Y. Khinchine, Mathematical Methods in the Theory of Queueing, New York: Hafner Publishing Co., 1960

[9] L. Kleinrock, Queueing Systems, Vol. I: Theory, Wiley, 1975

[10] J.F. Kurose, S. Singh and R. Chipalcatti, "A study of quasi-dynamic load sharing in soft real-time distributed computer systems," Proceedings of Real-Time Systems Symposium, New Oreans, Lousiana, 1986

[11] K.J. Lee and D.F. Towsley, "A comparison of priority-based decentralized load balancing policies," Proceedings of Performance'86 and ACM Sigmetrics 1986 Joint Conference, Raleigh, North Carolina, 1986

[12] K.J. Lee, Load Balancing in Distributed Computer Systems, Ph.D Dissertation, Dept. of Electrical and Computer Engineering, University of Massachusetts, 1987

[13] R. Mirchandaney and D. Towsley, "The effects of delays on the performance of load balancing policies", Proceedings of 2nd Intnl. Workshop on Appl. Math. and Perform./Reliab. Models of Comp./Comm. Syst., pp. 213-228, May 1987.

[14] M.F. Neuts, "Markov chains with applications in queueing theory, which have a matrix-geometric invariant probability vector," *Adv. Appl. Prob.*, vol. 10, pp. 185-212, 1978

[15] M.F. Neuts, Matrix-Geometric Solutions in Stochastic Models - an Algorithmic Approach, Johns Hopkins University Press, 1981

[16] M. Westcott,"Simple proof of a result on thinned point process," *The Annals of Prob.*, vol. 4, No. 1, pp. 89-90, 1976

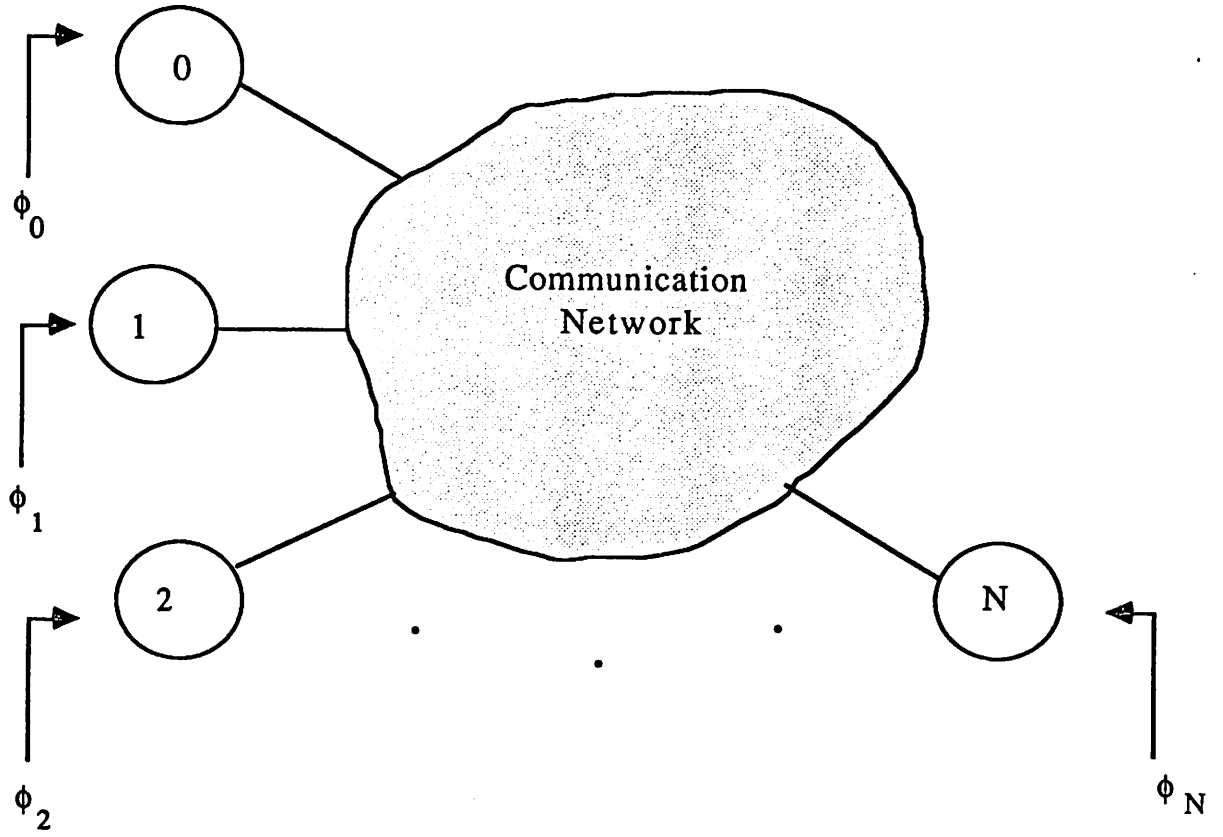

Figure 1. System model

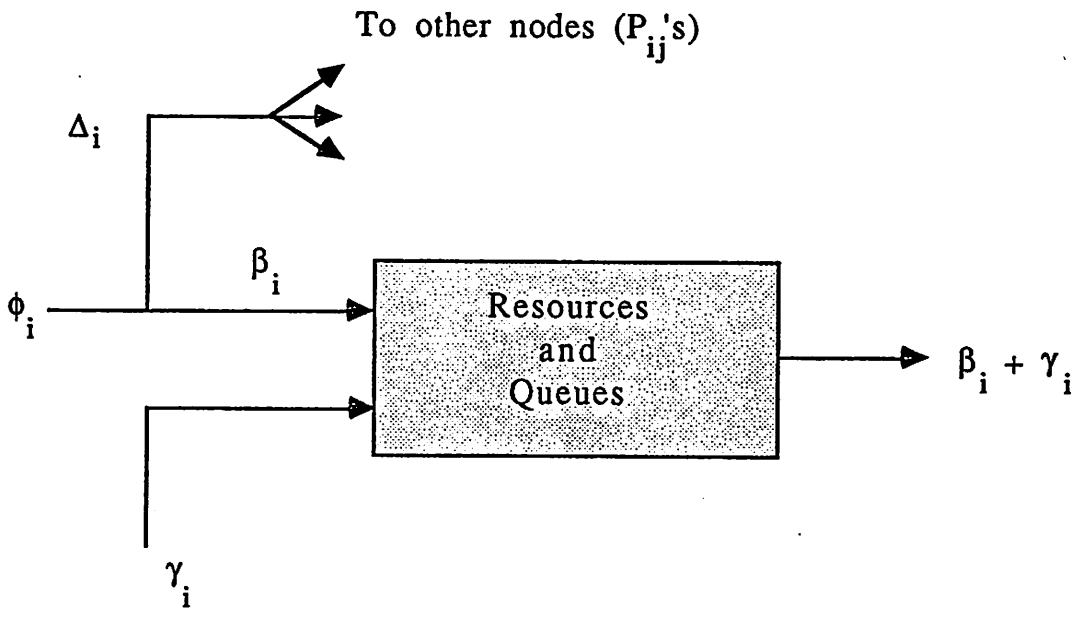

Figure 2. Job flows at node i

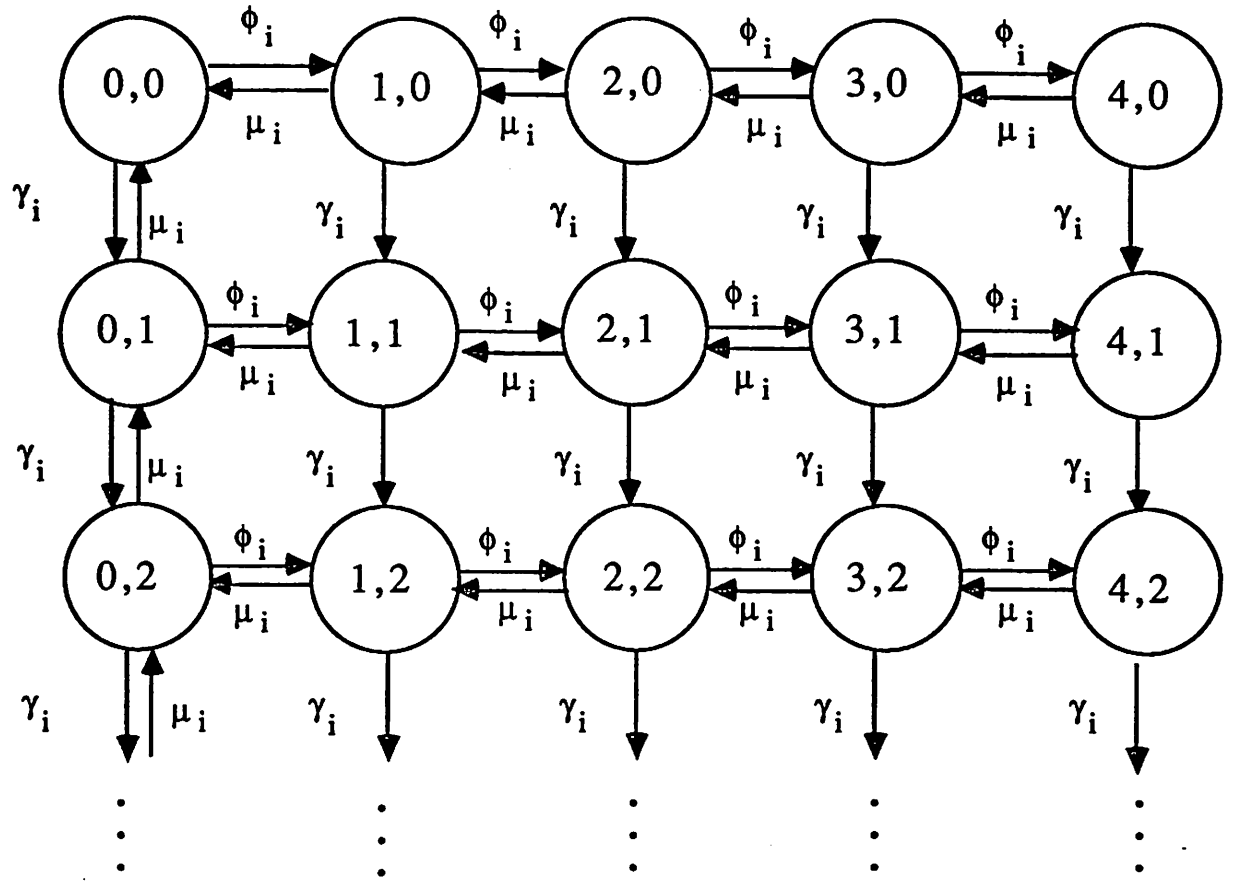

Figure 3. Markov chain describing the behavior of node i under Policy SLO when $T_i = 4$.

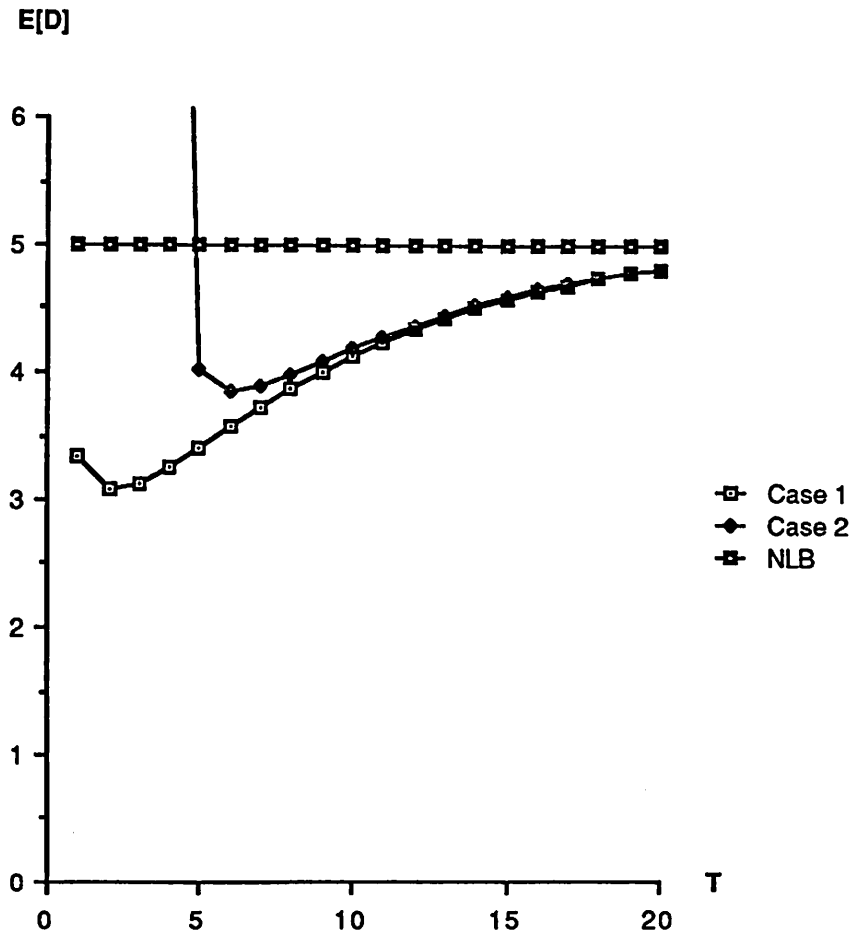

Figure 4 Behavior of the mean response time of a job as a function of threshold

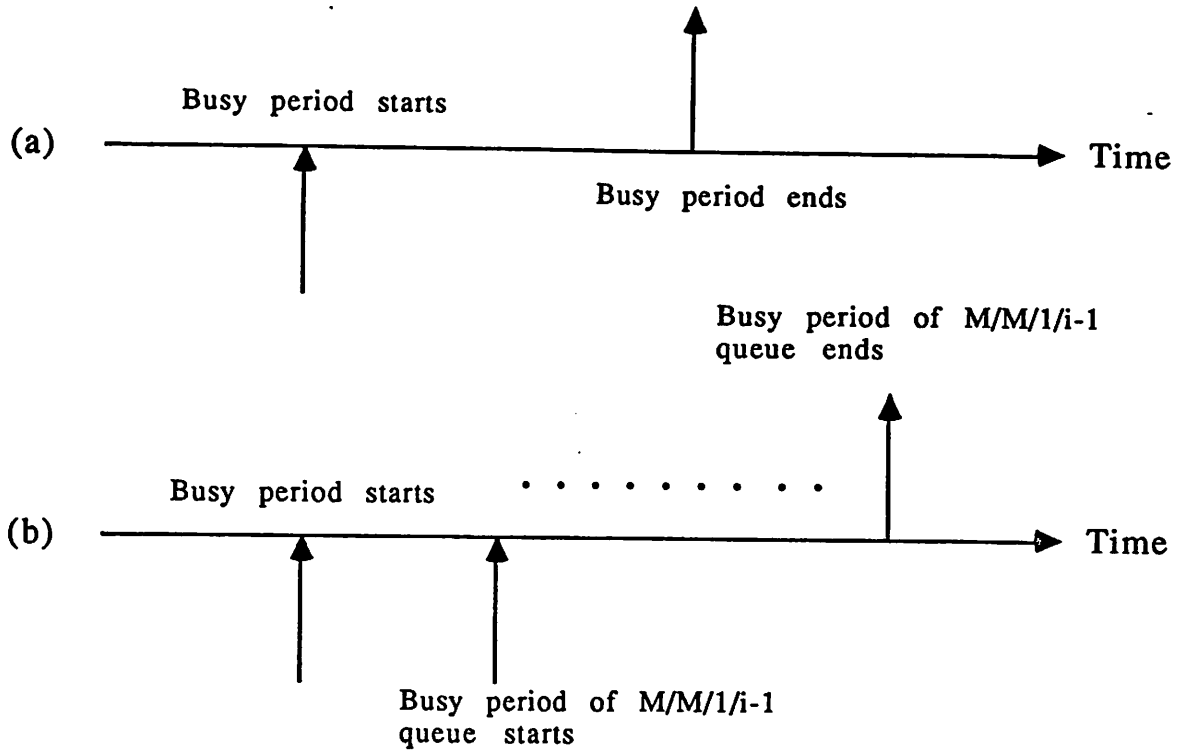

Figure A.1 Busy period analysis
(a) Busy period consisting of a single job
(b) Busy period consisting of at least two jobs

u	*SLO* (T)	*SIM* (%)	*NLB*
0.3	1.16 (1)	1.17 ± 0.01 (0.8 %)	1.43
0.5	1.49 (1)	1.52 ± 0.01 (1.9 %)	2.00
0.7	2.24 (2)	2.28 ± 0.05 (1.7 %)	3.33
0.9	5.48 (3)	5.59 ± 0.21 (1.9 %)	10.0

Table 1 Mean response time of a job in a peer system

$N = 4$, $1/\mu = 1.0$ and $1/\mu_{ch} = 0.01$

u_0, u_s	*SLO* (T_0, T_s)	*SIM* (%)	*NLB*
0.4, 0.7	0.76 (3,1)	0.77 ± 0.01 (1.3 %)	2.24
0.4, 0.9	0.95 (2,1)	1.00 ± 0.03 (5.0 %)	7.03
0.4, 1.1	1.23 (3,2)	1.29 ± 0.02 (4.7 %)	∞
0.8, 0.7	1.32 (4,2)	1.39 ± 0.03 (5.0 %)	2.09
0.8, 0.9	2.00 (4,3)	2.14 ± 0.05 (6.5 %)	5.76
0.8, 1.1	4.68 (5,5)	5.05 ± 0.27 (7.3 %)	∞

Table 2 Mean response time of a job in a star system

$N = 5$, $1/\mu_0 = 0.2$, $1/\mu_s = 1.0$ and $1/\mu_{ch} = 0.01$