February 1991 LIDS-P-2021

OPTIMAL DISTRIBUTED POLICIES FOR CHOOSING AMONG MULTIPLE SERVERS [†]

by

George D. Stamoulis [‡] and John N. Tsitsiklis [‡]

Abstract

We analyze a system consisting of multiple identical deterministic servers. Customers arrive in several streams; each customer has to decide which server to join by looking only at previous decisions of customers of the same stream. For three variations of this problem, we prove that Round Robin is the policy minimizing the total expected delay over all customers of an individual stream. We also consider the problem of optimizing the total expected delay over all streams; we investigate the performance of Round Robin and we argue that it is not optimal for this problem. Most of our results also apply under more general service time distributions.

[†] Research supported by the NSF under Grant ECS-8552419 and by the ARO under Grant DAAL03-86-K-0171.

[‡] Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, Mass. 02139, USA; e-mail: stamouli@lids.mit.edu, jnt@lids.mit.edu.

1. INTRODUCTION

The topic of server allocation in multiserver systems has received significant attention in the literature. There have been numerous papers dealing with policies for assigning the arriving customers to one of the available servers; the objective is usually the optimization of a performance measure such as the average delay per customer, the throughput etc. For most of the systems analyzed in the literature, it is assumed that they consist of exponential servers and that decisions are made under perfect information of the system's state; see [Sti85] and references therein. In this paper, we consider problems involving identical deterministic servers and decision-making under imperfect information. The general context of our analysis is depicted in Fig. 1: There are multiple arrival streams and multiple deterministic servers; each scheduler allocates the customers of the corresponding stream by only looking at its own previous decisions. There are basically two types of policies that can reasonably be applied, namely randomized allocation and Round Robin. For most of the variations of our problem, we establish that Round Robin is optimal for minimizing the total expected delay over all customers of a stream; we also present several related open questions.

Motivation for analyzing the problems considered in this paper primarily arises from the context of routing in store-and-forward data networks and in interconnection networks of multiprocessor computers. In most of such networks, there are several alternative paths for each origin-destination pair; usually, routing is done in a distributed fashion, with limited centralized coordination; see [BeG87], [BOSTT89] and references therein. Thus, each node routes the packets it generates (towards one of the alternative paths) by possibly knowing the routing policies of the other processors, but without having any further information on the actual paths and the timing of packets generated elsewhere. This fact motivates our considering systems that involve several contending streams, where each of them has only limited knowledge of the global state. Furthermore, note that, in the literature of routing in parallel computers, it is usually assumed that processors communicate by exchanging packets of fixed length (see [BOSTT89] and references therein); also packets of fixed length are used in the emerging standard of high-speed communications, namely the Asynchronous Transfer Mode [Min89], as well as in some standards for store-and-forward data networks [BeG87]. Therefore, the various interconnecting arcs in such networks should be modeled as deterministic servers. Finally, our problems are also related to the context of resource allocation in multiprocessor systems, where there are usually several contending sources generating jobs to be processed; even though these jobs may have different service times, several of our results are still relevant, because they also hold under more general service time distributions.

In [EVW80], Ephremides et al. analyze the following problem: A stream of arriving customers have to choose from L identical exponential servers, where $L \geq 2$; each customer has to join a server upon arrival, in such a way that the expected total delay over all customers arriving up to a certain time is minimized. It is established therein that Round Robin (RR) is the optimal allocation policy for the aforementioned problem. In this paper, we prove that the same result also applies for the case of deterministic servers; see §2.1. In §2.2, we consider a more general problem, namely we assume that each of the deterministic servers also receives a stream dedicated to be served by this same server; all of these streams are taken to have the same statistics; see Fig. 3. The customers allowed selection of a server are now called special; each of them chooses which server to join (upon arrival) by only knowing the previous decisions by special customers. (Arrivals corresponding to dedicated streams are not observable prior to decision-making.) Thus, each special customer selects its respective server while having imperfect information on the workload of the available servers. Again, RR is established to be the policy minimizing the expected total delay over all special customers. A similar problem was analyzed by Bonomi and Kumar [BoK90], with all dedicated and special streams being of the Poisson type. However, only static randomized policies are considered therein, that is all customers decide which server to join by applying the same probabilistic rule; this gives rise to a nonlinear optimization problem. (Obviously, RR is a non-static policy.)

In §3, we consider the system depicted in Fig. 1, where all arrival streams are identically distributed. Each customer chooses which of the available deterministic servers to join, by knowing only the previous decisions of customers belonging to the same stream. We analyze two optimization problems. In the first one, each stream of customers wishes to minimize its individual total expected delay; in this context, we prove that a set of randomized versions of RR constitute an equilibrium set of policies (in the sense of Game Theory); see §3.1. On the contrary, in §3.2, the objective is social optimization, that is to minimize the steady-state average delay per customer over all streams (which are taken Poisson). Regarding the latter problem, we present some first results and discuss some open questions that we intend to investigate in the future. A similar problem (with exponential servers and Poisson streams) was analyzed by Ni and Hwang [NiH85]. However, only static randomized policies are considered therein, thus reducing the problem to a nonlinear program.

To the best of our knowledge, the results of this paper are new. Even though there exists an extensive literature on stochastic scheduling, the use of non-static policies in problems with imperfect information has received limited attention. Related is the work by Beutler and Teneketzis [BeT88], who gave a framework for analyzing problems with only two alternative scheduling decisions. However, the problems analyzed therein involve one scheduler, in contrast with the problems discussed in §3 of the present paper.

2. PROBLEMS WITH A SINGLE SCHEDULER

2.1 One Arrival Stream and Several Servers

In this subsection, we prove that Round Robin (RR) is an optimal policy for assigning any stream of K arriving customers to one of L identical deterministic servers with unit service time; see Fig. 2. All servers are assumed to be initially empty; the optimization criterion is the expected total delay over the K arriving customers. A little thought reveals that RR simulates the $G/D/L$ queue, where each customer joins the first available server; this is due to the assumption of deterministic service time. Thus, it is intuitively clear that the RR policy should be optimal. However, the above argument does not constitute a rigorous proof, because it only shows that each customer suffers the smallest possible delay given the decisions of customers that arrived previously. Below, we present the rigorous proof for the case of $L = 2$ servers, and then we extend the result to all $L > 2$.

Proposition 1: Round Robin is an optimal policy for the case of 2 servers, for any sequence of arrival instants (either fixed or random). ■

Proof: Let t_k be the arrival time of the kth customer, for $k = 1, \ldots, K$, and let $w_k^{(j)}$ be the unfinished work at the jth server at time t_k-, for $j = 1, 2$. It is initially assumed that $t_1, \ldots, t_K$ are fixed and known and that the $w_k^{(j)}$'s are observable; these assumptions will prove to be redundant. It can be established that the optimal decision for the kth customer is to join the second server if and only if $w_k^{(1)} \geq w_k^{(2)}$. (In fact, for $w_k^{(1)} = w_k^{(2)}$, both decisions are equivalent.) The proof is done by a straightforward (yet tedious) Dynamic Programming argument with finite horizon; in order not to break continuity, we present this argument in the Appendix. Thus, the optimal policy is the myopic one, namely for each customer to join the least loaded server. Furthermore, we have $w_1^{(1)} = w_1^{(2)} = 0$, by the assumption that both servers are initially empty. Assuming that the first customer joins the second server, then we have $w_2^{(1)} + 1 \geq w_2^{(2)} \geq w_2^{(1)} = 0$, which implies that the second customer should join the first server. (Note that if $w_2^{(2)} = 0$, then the second customer could alternatively join the second server; however, this does not apply for all sequences of arrival instants.) Furthermore, we have

$$w_3^{(1)} = [w_2^{(1)} + 1 - (t_3 - t_2)]^+ \qquad \text{and} \qquad w_3^{(2)} = [w_2^{(2)} - (t_3 - t_2)]^+ ,$$

where $[\alpha]^+ \overset{\text{def}}{=} \max\{0, \alpha\}$; this implies that $w_3^{(2)} \leq w_3^{(1)} \leq w_3^{(2)} + 1$ and thus the third customer should join the second server etc. It follows that RR is an optimal policy, for any fixed sequence of arrival instants; since the structure of RR does not depend on the arrival instants, its optimality is preserved under random arrivals. Finally, the assumption that workloads at time $t_2, \ldots, t_K$ are observable also proved to be redundant. **Q.E.D.**

As already mentioned, there are cases where the optimal policy is not unique; e.g., if all interarrival times exceed unity (which equals the service time), then all policies are equivalent.

However, Proposition 1 guarantees that RR is always optimal. Next, we consider the case of more than two servers.

Proposition 2: Round Robin is an optimal policy for the case of $L > 2$ servers, for any sequence of arrival instants (either fixed or random). ■

Proof: The result is trivial when the total number K of customers does not exceed L. Henceforth, we assume that $K > L$. Since there are finitely many policies, at least one of them is optimal. We fix an optimal policy; let $c_{i1}, c_{i2}, \ldots$ be the ordered indices of customers choosing the ith server under this policy. Let us focus on the subset of customers choosing either the ith server or the jth server under the optimal policy (with $i \neq j$). By the optimality of RR for the case of two servers (see Proposition 1), there should hold

$$c_{i1} < c_{j1} < c_{i2} < c_{j2} < \cdots \qquad \text{or} \qquad c_{j1} < c_{i1} < c_{j2} < c_{i2} < \cdots\,; \tag{1}$$

for, otherwise, we can redistribute the customers between servers i and j and (in general) reduce the expected total delay. [For certain sequences of arrival instants, such a redistribution would not change the value of the expected total delay, thus yielding a new optimal policy that satisfies (1).] Therefore, we have $c_{i1} < c_{j2}$. Combining these inequalities over all pairs i, j it follows easily that $\{c_{11}, \ldots, c_{L1}\} = \{1, \ldots, L\}$; that is, each of the first L customers is assigned to a different server. Let us assume, without loss of generality, that $c_{i1} = i$ for $i = 1, \ldots, L$. Applying (1) with $i = 1$ and $j > 1$, it is seen that the leftmost set of inequalities apply; thus, we have $c_{12} < c_{j2}$ for all $j > 1$, which implies that $c_{12} = L + 1$. That is, after all L servers have been exhausted, the very first one is selected again. Furthermore, applying (1) with $i = 2$ and $j > 2$, it follows that $c_{22} < c_{j2}$ for all $j > 2$, which implies that $c_{22} = L + 2$. Continuing this argument, we can establish the optimality of the RR policy. **Q.E.D.**

Using Propositions 1 and 2, it is easily seen that, in the case of an ergodic arrival process, RR is optimal for minimizing the steady-state average delay per customer.

It is worth noting that Proposition 1 (and consequently Proposition 2) also holds under another interesting optimization criterion, namely for minimization of the expected departure time of the last customer to complete service. The proof follows the same lines: first, a finite horizon Dynamic Programming argument proves that joining the least loaded server is again optimal; then, using this, the optimality of RR is established as in Proposition 1.

2.2 Server Allocation in the Presence of Dedicated Streams

In the problem of §2.1, there was only a stream of K arriving customers to be assigned to the servers available; this stream will henceforth be referred to as special. In this subsection, we analyze a more general problem; in particular, we now assume that in addition to the special stream, there are L identically distributed streams of customers, with each of them being dedicated to a different server; see Fig. 3. No restrictions apply for either the marginal or the joint statistics of the dedicated streams; however, we assume that the special stream is independent of the dedicated ones. Each server operates on a FIFO basis. The scheduler receives only the special stream of customers and decides how to assign them to the servers, based only on its previous decisions; the scheduler cannot observe the workload of the servers. We shall prove that Round Robin is still optimal for minimizing the expected total delay over all special customers. This result makes perfect intuitive sense. Indeed, since the scheduler cannot observe the dedicated streams, it has the same "estimate" for the additional load imposed at each different server; thus, the optimal policy should be the same as in the absence of the dedicated streams. Though somewhat tedious, the proof to follow is based on this idea. [Of course, if the scheduler were allowed to observe the workloads of the servers, then the optimal policy would (in general) be different, since it would take into account this additional information.]

Proposition 3: Round Robin is an optimal policy for any sequence of arrival instants of the special customers (either fixed or random). ■

Proof: We shall only consider the case $L = 2$; a similar proof also applies for $L > 2$. Let $t_1, \ldots, t_K$ be the arrival instants of the special customers, which are initially taken to be fixed. Let $U_k^{(j)}$ denote the random variable corresponding to the unfinished work at the jth server at time t_k-; this includes the work induced by both the special and the dedicated streams. Our optimization problem is as follows:

$$\text{minimize} \sum_{k=1}^{K} E\left[x_k(U_k^{(1)} + 1) + (1 - x_k)(U_k^{(2)} + 1)\right], \quad \text{over } (x_1, \ldots, x_K) \in \{0,1\}^K. \tag{2}$$

Notice that $x_k = 1$ (resp. $x_k = 0$) corresponds to the kth customer joining the first server (resp. the second server). Let us assume that the service discipline is changed from FIFO to the following: all dedicated customers are alloted preemptive resume priority over the special ones. The distributions of the $U_k^{(j)}$'s remain the same, because the new service discipline is work-conserving. Henceforth, we assume that this new discipline applies. We have

$$U_k^{(j)} = V_k^{(j)} + W_k^{(j)}, \tag{3}$$

where the random variable $V_k^{(j)}$ corresponds to the contribution of the dedicated customers to the unfinished work at the jth server at time t_k-; similarly, $W_k^{(j)}$ corresponds to the unfinished work due to special customers. Using (2) and (3) and omitting a constant term, it is seen that our optimization problem is equivalent to the following:

$$\text{minimize} \sum_{k=1}^{K} E\left[\left(x_k V_k^{(1)} + (1 - x_k)V_k^{(2)}\right) + \left(x_k W_k^{(1)} + (1 - x_k)W_k^{(2)}\right)\right],$$
$$\text{over } (x_1, \ldots, x_K) \in \{0,1\}^K; \tag{4}$$

Clearly, special customers are transparent to the dedicated ones; this implies that $V_k^{(j)}$ does not depend on $(x_1, \ldots, x_K)$. Moreover, by symmetry between the dedicated streams, $V_k^{(1)}$ and $V_k^{(2)}$ are identically distributed. Thus, we have

$$E\left[x_k V_k^{(1)} + (1 - x_k) V_k^{(2)}\right] = E[V_k^{(1)}],$$

and the optimization problem of (4) reduces to the following:

$$\text{minimize} \sum_{k=1}^{K} E\left[x_k W_k^{(1)} + (1 - x_k) W_k^{(2)}\right], \quad \text{over } (x_1, \ldots, x_K) \in \{0,1\}^K. \tag{5}$$

Recall now that special customers are served only in the absence of any dedicated customers at the same server. Therefore, during the interval $[t_k, t_{k+1})$, the jth server (where $j \in \{1, 2\}$) can reduce the unfinished work due to special customers by as much as $I_k^{(j)}$, where the random variable $I_k^{(j)}$ is the total idle period over the interval $[t_k, t_{k+1})$ of a $\cdot/D/1$ queue serving only the dedicated stream of the jth server. Recalling also the interpretation of x_k (and that service times equal unity), we have

$$W_{k+1}^{(1)} = \left[W_k^{(1)} + x_k - I_k^{(1)}\right]^+ \tag{6.a}$$

and

$$W_{k+1}^{(2)} = \left[W_k^{(2)} + (1 - x_k) - I_k^{(2)}\right]^+, \tag{6.b}$$

with $W_1^{(1)} = W_1^{(2)} = 0$. Since the two dedicated streams are symmetric and independent of the special stream, the random vectors $(I_1^{(1)}, \ldots, I_k^{(1)})$ and $(I_1^{(2)}, \ldots, I_k^{(2)})$ are identically distributed for any $k \in \{1, \ldots, K - 1\}$. Hence, for each fixed $(x_1, \ldots, x_K)$, the distribution of the vector $(W_1^{(2)}, \ldots, W_K^{(2)})$ remains the same if we replace $I_k^{(2)}$ with $I_k^{(1)}$ in $(6.b)$ (*). Let us assume that each $W_k^{(2)}$ is updated according to this rule [instead of the rule in $(6.b)$], namely that

$$W_{k+1}^{(2)} = \left[W_k^{(2)} + (1 - x_k) - I_k^{(1)}\right]^+, \tag{6.c}$$

Then, for each fixed $(x_1, \ldots, x_K)$, the value of the "cost" function in (4) still remains the same, because the expectations involved depend only on the marginal distributions of the vectors $(W_1^{(1)}, \ldots, W_K^{(1)})$ and $(W_1^{(2)}, \ldots, W_K^{(2)})$. [Note that by replacing $(6.b)$ with $(6.c)$, the joint distribution of these two vectors is (in general) modified.] Notice now that $(6.a)$ and $(6.c)$ are the updating rules for the unfinished work in a two-server system receiving only the special

(*) This becomes apparent after "unfolding" the iteration in $(6.b)$ as follows:

$$W_{k+1}^{(2)} = \max\{0, (1 - x_k) - I_k^{(2)}, (1 - x_{k-1}) + (1 - x_k) - I_{k-1}^{(2)} - I_k^{(2)}, \ldots,$$
$$(1 - x_1) + \cdots + (1 - x_k) - I_1^{(2)} - \cdots - I_k^{(2)}\}.$$

customers at the random arrival instants $I_1^{(1)}, I_1^{(1)} + I_2^{(1)}, \ldots, I_1^{(1)} + I_2^{(1)} + \cdots + I_{K-1}^{(1)}$. Since the optimization problem in (4) also is the same as the one considered in §2.1, it follows from Proposition 1 that RR is an optimal policy for any fixed sequence of arrival instants of the special customers. Again, RR is also optimal for random such instants. **Q.E.D.**

It is worth noting that Proposition 3 also holds for any service time distribution for which Proposition 1 applies; e.g., for the exponential distribution, according to the result of [EVW80].

As already mentioned in §1, the problem analyzed above is motivated from the context of distributed routing. Indeed, let us consider the example of Fig. 4. Nodes 0,1 and 3 send packets to node 2; nodes 1 and 3 behave symmetrically. Of course, node 2 cannot observe the packets generated by 1 and 3. According to Proposition 3, its optimal routing policy is to alternate in sending packets through paths $0 \to 1 \to 2$ and $0 \to 3 \to 2$. Note that arcs $1 \to 2$ and $3 \to 2$ correspond to the servers of our problem.

3. PROBLEMS WITH SEVERAL SCHEDULERS

The system to be analyzed in this section is depicted in Fig. 1. There are L deterministic servers, with $L \geq 2$. Customers arrive in N independent and identically distributed streams, where $N \geq 2$. Upon arrival of a customer, the corresponding scheduler decides which server she will join, based only on its own previous decisions. It is assumed that each scheduler knows the policy of the rest, without ever receiving any additional information. Two problems will be analyzed in this context, namely one with individual optimization (per stream) and another with social optimization (over all streams). Note that the term "individual" here refers to a single stream, rather than to a single customer (as in [BeS83]); however, using this term in the present context is appropriate, because each stream is allocated to the available servers on the basis of individual information.

Again, the system under consideration is motivated from the context of distributed routing. An example of the same spirit as that of Fig. 4 can be easily constructed; see Fig. 5. For the network depicted therein, it is assumed that nodes 1 and 2 send packets to node 5; obviously, we have $N = 2$ and $L = 2$, with arcs $3 \to 5$ and $4 \to 5$ corresponding to the servers.

3.1 Individual Optimization of Contending Streams

In this subsection, we assume that each of the streams of customers wishes to minimize its individual total expected delay. Since the situation is "competitive", we are interested in finding equilibrium sets of policies. If the schedulers follow such a set of policies, then none of them would have incentive to deviate from its own policy; this would ensure fairness among the various nodes. As will be proved below, any N-tuple of Symmetrically Randomized Round Robin policies (Sym.Rand.RR) is an equilibrium set. This class of policies is defined as follows: Assuming that a scheduler applies RR, we define as its decision pattern the vector of the first

L allocation decisions (*); of course, this is a permutation of $(1, \ldots, L)$ and it is sufficient to define the entire sequence of decisions of the scheduler, because it is repeated periodically. A policy will be said to be Sym.Rand.RR if the scheduler selects its decision pattern randomly, in such a way that each entry assumes any fixed value $m \in \{1, \ldots, L\}$ with the same probability (namely, with probability $\frac{1}{L}$). For example, one such policy is obtained when the decision pattern of a scheduler is selected randomly, with all $L!$ possible orders being equiprobable; another such policy is obtained when the selection is over all L cyclic shifts of $(1, \ldots, L)$, with all permissible outcomes having a priori probability $\frac{1}{L}$. It should be noted that when a scheduler is known to adopt a Sym.Rand.RR policy, the other schedulers cannot observe its decision pattern, even though they may know what the possible patterns are. This assumption is consistent with on-line distributed routing (see also §1), where the decision pattern of a node may be determined progressively, as more packets are generated.

Proposition 4: Any N-tuple of Symmetrically Randomized Round Robin policies is an equilibrium set of policies, for any marginal distribution of the arrival streams. ■

Proof: Let us assume that each of the schedulers corresponding to the first $N - 1$ streams adopts a Sym.Rand.RR policy (not necessarily the same); we shall prove that, given this information, the Nth scheduler should also adopt such a policy, in order to minimize the total expected delay of the customers of the Nth stream. Indeed, the Nth scheduler visualizes the situation as follows: Each of the servers will receive a "dedicated" stream, which in fact consists of customers originating from the first $N - 1$ "original" streams. Since all servers are treated symmetrically by a Sym.Rand.RR policy, the Nth scheduler can tell a priori that all these "dedicated" streams have the same statistics. Therefore, according to Proposition 3, RR (with any decision pattern) is an individually optimal policy for the Nth scheduler. This implies that any Sym.Rand.RR policy also is individually optimal for the Nth scheduler, which proves the result. **Q.E.D.**

Since the above result is a consequence of Proposition 3, it also holds for any service time distribution for which Proposition 1 applies.

It should be noted that an N-tuple of RR policies is not necessarily an equilibrium point. Consider, for example, the case of $N = L = 2$, with both streams consisting of 2 customers arriving at times 0 and 1. Clealry, if both N schedulers apply RR with decision pattern $(1, 2)$, then each of them would have been better off if it had choosen the decision pattern $(2, 1)$. In fact, if one of the schedulers chooses $(1, 2)$ as its decision pattern while the other chooses $(2, 1)$, then each customer would suffer the minimum possible delay (namely, one time unit). This pair of RR policies results in the minimum total expected delay over all customers of both streams; that is, it constitutes a social optimum. On the contrary, a pair of Sym.Rand.RR

(*) In order to avoid trivialities, we assume that, with positive probability, each stream consists of at least L customers.

policies is not a social optimum, because with positive probability the two schedulers choose the same decision pattern.

It would be desirable to attain an equilibrium set of policies that do not employ any randomization at all. Consider, for example, the following variation of the problem under analysis: All arrival streams are Poisson with rate $\rho < 1$ (see also §3.2) and, for each stream, the objective is individual minimization of the steady-state average delay per customer. It is can be proved that any N-tuple of RR policies is an equilibrium set. The idea of the proof is as follows: Let P_j denote the decision pattern of the jth scheduler, for $j = 1, \ldots, N$; note that P_j is some fixed permutation of $(1, \ldots, L)$. Let $s_k^{(j)}$ denote the index of the server to receive the first customer of the jth stream to be served in the kth busy period. Given the initial decision patterns of the schedulers, it is straightforward that the vector $(s_k^{(1)}, \ldots, s_k^{(N)})$ evolves as an irreducible finite-state homogenious Markov chain. (Note that all entries of the corresponding transition matrix are positive.) Thus, for fixed initial decision patterns $P_1, \ldots, P_N$, the steady-state performance is the same as that of a system using a set of Sym.Rand.RR policies with the permissible initial decision patterns for the jth scheduler being the L cyclic shifts of P_j (for all $j \in \{1, \ldots, N\}$). Using ergodicity and Proposition 4, it follows that RR with decision patterns $P_1, \ldots, P_N$ also constitutes an equilibrium set of policies.

3.2 Social Optimization of Contending Streams

Next, we discuss a problem of optimizing a "global" performance measure in the system introduced in the beginning of this section. We now assume (for simplicity) that each arrival stream is Poisson with rate ρ. We are interested in minimizing the steady-state average delay per customer, where the average is taken over all streams. We shall consider the simple case $N = L$, and we assume that $\rho < 1$ so that stability is attainable; e.g., by having each customer choosing radomly which server to join (with all servers being equiprobable), the system reduces to N independent $M/D/1$ queues, each with utilization ρ.

We denote by $f(N;\rho)$ the optimal average delay per customer. We are not able to find an exact expression for $f(N;\rho)$; however, we derive some bounds that provide us with some qualitative view of its behavior. In particular, it is easily seen that

$$f(N_1N_2;\rho) \le \min\{f(N_1;\rho), f(N_2;\rho)\}\,. \tag{7}$$

Indeed, for $N = N_1N_2$, we can group the servers and the streams in N_1-tuples and dedicate a different group of servers to each group of streams; by applying the optimal policy corresponding to $N = N_1$ within each of the groups of streams, we attain an average delay of $f(N_1;\rho)$ per customer. This implies that $f(N_1N_2;\rho) \le f(N_1;\rho)$; the inequality $f(N_1N_2;\rho) \le f(N_2;\rho)$ can be proved in exactly the same way. Investigating the structure of the optimal set of policies and the behavior of $f(N;\rho)$ as a function of N seems to be a rather hard problem. Since our problem is related to distributed routing, it is of interest to analyze the asymptotic case

$N \to \infty$. (In the literature of interconnection networks, asymptotics with respect to the network size play a key role.) It is easily seen from (7) that the performance of the optimal set of policies does not deteriorate; this, however, does not necessarily imply that the optimal delay decreases as $N \to \infty$. It is conjectured that $\lim_{N\to\infty} f(N;\rho)$ exists and that it is bounded away from 1. In other words, some delay due to contention is inevitable even for very large N. This is in contrast with the $M/D/N$ queue, for which the average delay per customer tends to 1, for $N \to \infty$ and fixed utilization ρ; such a queue would be obtained if all N streams were merged to one.

For all problems analyzed so far, Round Robin proved to be an optimal policy. Unfortunately, this is not the case for the present problem. Indeed, let us assume that each of the schedulers applies an RR policy. Then, for any n and l, the sub-stream of customers from the nth stream that join the lth server form a renewal process, with interarrival time distributed as Erlang with N degrees of freedom and expected value $\frac{N}{\rho}$. Each server is fed by the sum of N such processes; for $N \to \infty$, this compound process converges weakly to a Poisson process with rate ρ (see [Çin72]). Therefore, as $N \to \infty$, each of the N queues in the system "tends to behave" as an $M/D/1$ queue with utilization ρ. Thus, letting $r(N;\rho)$ be the average delay per customer attained by RR, we have $\lim_{N\to\infty} r(N;\rho) = 1 + \frac{\rho}{2(1-\rho)}$ (see [Kle75]); though intuitively clear, the derivation of this limit is technically complicated and will be clarified further in the final version of the paper. On the other hand, it will be proved below that $r(2;\rho)$ is strictly smaller than this limiting value; see Proposition 5. If N is an even number, an average delay per customer equal to $r(2;\rho)$ would be attained by dedicating a different pair of servers to each pair of streams (and forcing each scheduler to apply RR between the corresponding two servers). Hence, RR over all N servers is definitely non-optimal for large N.

Proposition 5: There holds

$$r(2;\rho) < 1 + \frac{\rho}{2(1-\rho)}\,. \qquad \blacksquare$$

Proof: Assuming that $N = 2$, we denote as $M/D/1$ the policy that assigns all customers of the jth stream to the jth server, for $j = 1,2$; clearly, this results in a steady-state average delay of $1 + \frac{\rho}{2(1-\rho)}$. Thus, it suffices to prove that, if both schedulers apply RR, then the performance is better than the one attained under $M/D/1$. Starting at time 0 with an empty system, let $s^{(j)}$ be the index of the server to be joined by the first customer to arrive through the jth stream (when applying RR), for $j = 1,2$. We consider the first two customers to arrive; clearly, the probability that they both "originate" from the same stream equals $\frac{1}{2}$. If $s^{(1)} = s^{(2)}$, then RR with two customers from the same stream (resp. from different streams) is equivalent to $M/D/1$ with two customers from different streams (resp. from the same stream); since the two scenarios are equiprobable, both RR and $M/D/1$ perform the same for $s^{(1)} = s^{(2)}$. If $s^{(1)} \neq s^{(2)}$, then the two customers served will join different servers (under RR), regardless of which streams they "originate" from; this outperforms $M/D/1$, because if both customers originate from the same stream then one of them will delay the other with probability $1-e^{-2\rho}$. By appropriately coupling the two systems, namely that with $M/D/1$ and that with RR, it follows that the total delay for the first two customers is definitely not smaller under $M/D/1$; moreover, under $M/D/1$, the next two customers will encounter a system at least as loaded as under RR. Continuing this argument by considering the arriving customers pairwise, it follows (by induction) that, under RR, the total expected delay over the first $2m$ customers to arrive does not exceed that attained under $M/D/1$, for $m = 1, \ldots$ Letting $m \to \infty$ and using ergodicity, it is seen that RR is at least as good as $M/D/1$, regarding the steady-state average delay per customer. Moreover, RR is strictly better, because the stationary probability that $s^{(1)} \neq s^{(2)}$ at the beginning of a busy period is positive. (To see this, just notice the following: If a busy period starts with $s^{(1)} = s^{(2)}$, then, with probability $e^{-2\rho}$, only one customer will be served until the system empties; in such a case, the next busy period will start with $s^{(1)} \neq s^{(2)}$.)

Q.E.D.

It is worth noting that the idea used in the proof of Proposition 5 can be applied for any service time distribution; thus, it can be proved that, for a general such distribution (with expected value equal to 1 and coefficient of variation c), there holds $r(2;\rho) < 1 + \frac{(1+c^2)\rho}{2(1-\rho)}$. (This upper bound equals the steady-state average delay per customer of the corresponding $M/G/1$ queue; see [Kle75].)

$\rho = 0.5$		
N	RR	$M/D/1$
1	1.485	1.485
2	1.234	1.487
3	1.227	1.477
4	1.250	1.469
5	1.272	1.475
10	1.355	1.455
20	1.398	1.468
50	1.440	1.458

Table I

Next, we comment on the behavior of $r(N;\rho)$ as a function of N, for fixed ρ. Numerical experiments suggest that $r(N;\rho)$ exhibits a global minimum at a value N', which depends on ρ. Under light or medium traffic, N' appears to be very small (either 2 or 3) and the minimum is rather sharp. In Table I, we present some experimental results for $\rho = 0.5$. The entries of the column labeled RR correspond to $r(N;0.5)$, while those of the column labeled $M/D/1$ correspond to the average delay under the $M/D/1$ policy defined in the proof of Proposition 5; both policies are compared under the same sequences of arrivals. (Note that, for $\rho = 0.5$, the upper bound given by Proposition 5 equals 1.5.) Other experimental results suggest that, as $\rho \to 1$, the value of N' increases and the minimum becomes more flat. The behavior of $r(N;\rho)$ agrees with intuition. As N increases, there is a trade-off between the increased choice of servers for each individual customer and the increased "entropy" in the compound arrival process of each server. For very small N, the former factor prevails and there is a benefit in increasing N, while, for larger N, it is the latter factor that prevails and the performance deteriorates. Proving rigorously the validity of these observations seems to be rather hard.

So far we have only dealt with the case $L = N$; as far as asymptotics with respect to N are concerned, the case $L = \beta N$ (where β is constant) can be treated similarly. Of interest are also the cases $N = o(L)$ and $L = o(N)$, which however seem to be simpler. For example, for constant N, $\rho = \frac{\alpha L}{N}$ (where $\alpha < 1$ is a constant), and $L \to \infty$, the optimal delay should converge to 1, at least as fast as the delay of an $M/D/\ell$ queue with arrival rate α and $\ell \to \infty$; indeed, in this case, an efficient policy is to allocate different $\lfloor \frac{L}{N} \rfloor$ servers to each of the arrival streams. On the contrary, for constant L, $\rho = \frac{\alpha L}{N}$, and $N \to \infty$, the optimal delay should converge to that of an $M/D/1$ queue with arrival rate α; in this case, each of the servers (under any "reasonable" policy) is fed by a process that converges to Poisson, as $N \to \infty$. A more detailed discussion of such cases will be presented in the final version of the paper.

4. CONCLUDING REMARKS

In this paper, we have analyzed server allocation problems involving deterministic servers and decision-making under imperfect information. We began with problems involving a single scheduler and then we turned our attention to those with multiple schedulers. For the latter type of problems, we considered both cases of individual (per stream) and social (over all streams) optimization. Apart from deriving several results on the corresponding optimal policies, we stated some conjectures which we intend to investigate in the future. All problems considered are motivated from the context of distributed routing in data networks and in multiprocessor computers. Given its diversity and extent of applications, that field appears to be rich in scheduling problems that have not attracted yet the attention of researchers. Analyzing such problems seems to be an interesting as well as challenging direction for further research.

REFERENCES

[BeG87] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall.

[BeS83] C.E. Bell and S. Stidham, "Individual versus Social Optimization in the Allocation of Customers to Alternative Servers", *Management Science*, vol. 29, pp. 831-839.

[BeT88] F.J. Beutler and D. Teneketzis, "Routing in Queueing Networks Under Imperfect Information: Stochastic Dominance and Thresholds", *Stochastics and Stochastics Reports*, vol. 26, pp. 81-100.

[BOSTT89] D.P. Bertsekas, C. Ozveren, G.D. Stamoulis, P. Tseng, and J.N. Tsitsiklis, "Optimal Communication Algorithms for Hypercubes", Report LIDS-P-1847, Laboratory for Information and Decision Systems, M.I.T.

[BoK90] F. Bonomi and A. Kumar, "Adaptive Optimal Load Balancing in a Nonhomogeneous Multiserver System with a Central Job Scheduler", *IEEE Trans. Comput.*, vol. C-39, pp. 1232-1250.

[Çin72] E. Çinlar, "Superposition of Point Processes", In P. Lewis (Ed.), *Stochastic Point Processes: Statistical Analysis, Theory and Applications*, John Wiley, pp. 549-606.

[EVW80] A. Ephremides, P. Varaiya, and J. Walrand, "A Simple Dynamic Routing Problem", *IEEE Trans. Auto. Control*, vol. AC-25, pp. 690-693.

[Kle75] L. Kleinrock, *Queueing Systems, Vol. I: Theory*, John Wiley.

[Min89] S.E. Minzer, "Broadband ISDN and Asynchronous Transfer Mode", *IEEE Communications Magazine*, vol. 27, pp. 17-57.

[NiH85] L.M. Ni and K. Hwang, "Optimal Load Balancing in a Multiprocessor System", *IEEE Trans. Softw. Eng.*, vol. SE-11, pp. 491-496.

[Sti85] S. Stidham, "Optimal Control of Admission to a Queueing System", *IEEE Trans. Auto. Control*, vol. AC-30, pp. 705-713.

APPENDIX

In this appendix, we establish a result used in the proof of Proposition 1.

We consider the system of Fig. 2, for the case of 2 servers. Let t_k be the arrival time of the kth customer (which is taken to be fixed), for $k = 1, \ldots, K$, and let $w_k^{(j)}$ be the unfinished work at the jth server at time t_k-, for $j = 1, 2$. The objective is to minimize the total delay over all customers. Assuming that the $w_k^{(j)}$'s are observable by the scheduler, it will be proved that the optimal decision for the kth customer is to join the second server if and only if $w_k^{(1)} \geq w_k^{(2)}$; in fact, for $w_k^{(1)} = w_k^{(2)}$, both decisions are equivalent.

We shall apply Dynamic Programming with finite horizon. Let $V_k(w^{(1)}, w^{(2)})$ denote the optimal cost-to-go function (at stage k) and let $\mu_k^*(w^{(1)}, w^{(2)})$ be the corresponding optimal decision for $w_k^{(1)} = w^{(1)}$ and $w_k^{(2)} = w^{(2)}$. Bellman's equations are as follows:

$$\begin{aligned} V_k(w^{(1)}, w^{(2)}) = \min\Big\{ & (w^{(1)} + 1) + V_{k+1}([w^{(1)} + 1 - \tau_k]^+, [w^{(2)} - \tau_k]^+), \\ & (w^{(2)} + 1) + V_{k+1}([w^{(1)} - \tau_k]^+, [w^{(2)} + 1 - \tau_k]^+) \Big\} , \\ & \text{for } k = 1, \ldots, K-1 , \qquad (A.1) \end{aligned}$$

where $\tau_k \overset{\text{def}}{=} t_{k+1} - t_k$; also,

$$V_K(w^{(1)}, w^{(2)}) = \min\{w^{(1)} + 1, w^{(2)} + 1\} . \qquad (A.2)$$

Obviously, $V_k(w^{(1)}, w^{(2)})$ is symmetric and increasing in both its arguments. Because of symmetry, we only need to consider the case $w^{(1)} \geq w^{(2)}$ and show that $\mu_k^*(w^{(1)}, w^{(2)}) = 2$. Indeed, we shall prove by backwards induction that

$$\mu_k^*(w^{(1)}, w^{(2)}) = 2 \qquad \text{and} \qquad V_k(w^{(1)}+1, w^{(2)}) - V_k(w^{(1)}, w^{(2)}+1) \geq w^{(2)} - w^{(1)} \,, \qquad \text{for } w^{(1)} \geq w^{(2)} \,(A.3)$$

First, notice that, for $k = K$ and $w^{(1)} \geq w^{(2)}$, we have from $(A.2)$ that $\mu_k^*(w^{(1)}, w^{(2)}) = 2$ and

$$\begin{aligned} V_K(w^{(1)}+1, w^{(2)}) - V_K(w^{(1)}, w^{(2)}+1) &= \min\{w^{(1)}+1, w^{(2)}\} - \min\{w^{(1)}, w^{(2)}+1\} \\ &= w^{(2)} - \min\{w^{(1)}, w^{(2)}+1\} \\ &\geq w^{(2)} - w^{(1)} \,, \end{aligned}$$

which establishes $(A.3)$ for the final stage. Next, we fix some $k \in \{1, \ldots, K-1\}$. Assuming that $(A.3)$ holds for $k+1$ we shall prove that it holds for k as well. We have to consider four different cases:

Case of $w^{(2)} \leq \tau_k - 1$: The righthand quantity in Bellman's equation $(A.1)$ simplifies to

$$1 + \min\left\{w^{(1)} + V_{k+1}\left([w^{(1)} + 1 - \tau_k]^+, 0\right), w^{(2)} + V_{k+1}\left([w^{(1)} - \tau_k]^+, 0\right)\right\} ;$$

using monotonicity and the fact $w^{(1)} \geq w^{(2)}$, it follows that $\mu_k^*(w^{(1)}, w^{(2)}) = 2$.

Case of $\tau_k - 1 < w^{(2)} \leq w^{(1)} \leq \tau_k$: The righthand quantity in Bellman's equation $(A.1)$ simplifies to

$$1 + \min\left\{w^{(1)} + V_{k+1}(w^{(1)} + 1 - \tau_k, 0), w^{(2)} + V_{k+1}(0, w^{(2)} + 1 - \tau_k)\right\} ;$$

using symmetry, monotonicity and the fact $w^{(1)} \geq w^{(2)}$, it follows that $\mu_k^*(w^{(1)}, w^{(2)}) = 2$.

Case of $\tau_k - 1 < w^{(2)} < \tau_k < w^{(1)}$: The righthand quantity in Bellman's equation $(A.1)$ simplifies to

$$1 + \min\left\{w^{(1)} + V_{k+1}(w^{(1)} + 1 - \tau_k, 0), w^{(2)} + V_{k+1}(w^{(1)} - \tau_k, w^{(2)} + 1 - \tau_k)\right\} . \qquad (A.4)$$

Applying the induction hypothesis $(A.3)$, we obtain

$$V_{k+1}(w^{(1)} + 1 - \tau_k, 0) \geq V_{k+1}(w^{(1)} - \tau_k, 1) + \tau_k - w^{(1)} ; \qquad (A.5)$$

Notice now that, for the present case, we have $w^{(2)} \leq \tau_k$ and $w^{(2)} + 1 - \tau_k \leq 1$; thus, it follows that

$$V_{k+1}(w^{(1)} - \tau_k, 1) + \tau_k - w^{(1)} \geq V_{k+1}(w^{(1)} - \tau_k, w^{(2)} + 1 - \tau_k) + w^{(2)} - w^{(1)} .$$

Combining this with $(A.5)$, we obtain

$$V_{k+1}(w^{(1)} + 1 - \tau_k, 0) \geq V_{k+1}(w^{(1)} - \tau_k, w^{(2)} + 1 - \tau_k) + w^{(2)} - w^{(1)}\,,$$

which together with $(A.4)$ implies that $\mu_k^*(w^{(1)}, w^{(2)}) = 2$.

Case of $\tau_k \leq w^{(2)} \leq w^{(1)}$: The righthand quantity in Bellman's equation $(A.1)$ equals

$$1 + \min\left\{w^{(1)} + V_{k+1}(w^{(1)} + 1 - \tau_k, w^{(2)} - \tau_k), w^{(2)} + V_{k+1}(w^{(1)} - \tau_k, w^{(2)} + 1 - \tau_k)\right\}.$$

This together with the induction hypothesis $(A.3)$ implies that $\mu_k^*(w^{(1)}, w^{(2)}) = 2$.

So far, we have established the leftmost part of $(A.3)$. As for the rightmost part, we have (for $w^{(1)} \geq w^{(2)}$)

$$\begin{aligned}
V_k(w^{(1)} + 1, w^{(2)}) &= (w^{(2)} + 1) + V_{k+1}\big([w^{(1)} + 1 - \tau_k]^+, [w^{(2)} + 1 - \tau_k]^+\big) \\
&= (w^{(2)} - w^{(1)}) + (w^{(1)} + 1) + V_{k+1}\big([w^{(1)} + 1 - \tau_k]^+, [w^{(2)} + 1 - \tau_k]^+\big) \\
&\geq (w^{(2)} - w^{(1)}) + \min\Big\{(w^{(1)} + 1) + V_{k+1}\big([w^{(1)} + 1 - \tau_k]^+, [w^{(2)} + 1 - \tau_k]^+\big), \\
&\qquad\qquad (w^{(2)} + 2) + V_{k+1}\big([w^{(1)} - \tau_k]^+, [w^{(2)} + 2 - \tau_k]^+\big)\Big\} \\
&= (w^{(2)} - w^{(1)}) + V_k(w^{(1)}, w^{(2)} + 1)\,,
\end{aligned}$$

where we have also used Bellman's equation $(A.2)$. This completes the inductive proof of $(A.3)$ and the derivation of the optimal policy. **Q.E.D.**

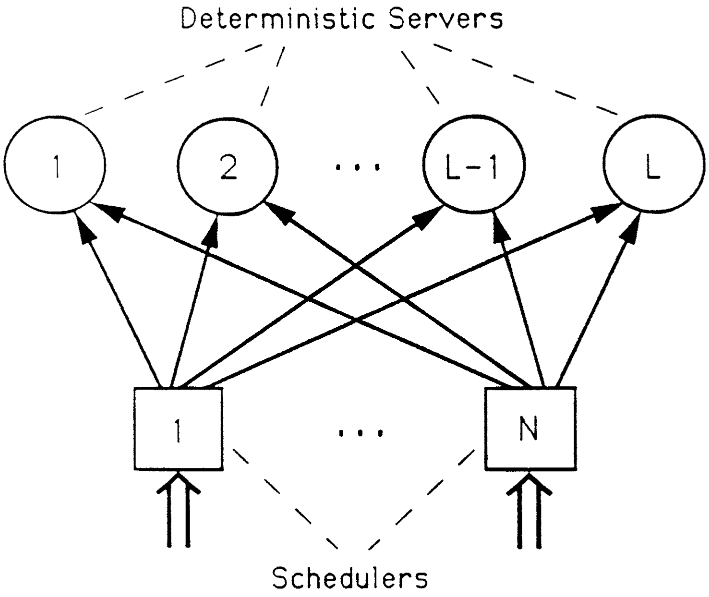

Figure 1: The general system.

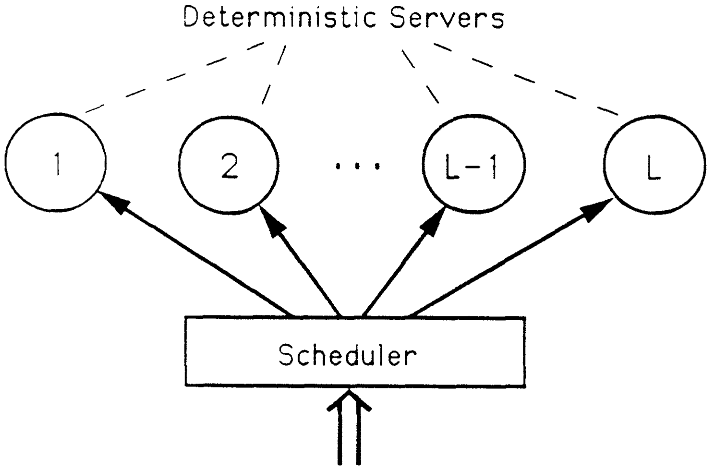

Figure 2: One arrival stream and several servers.

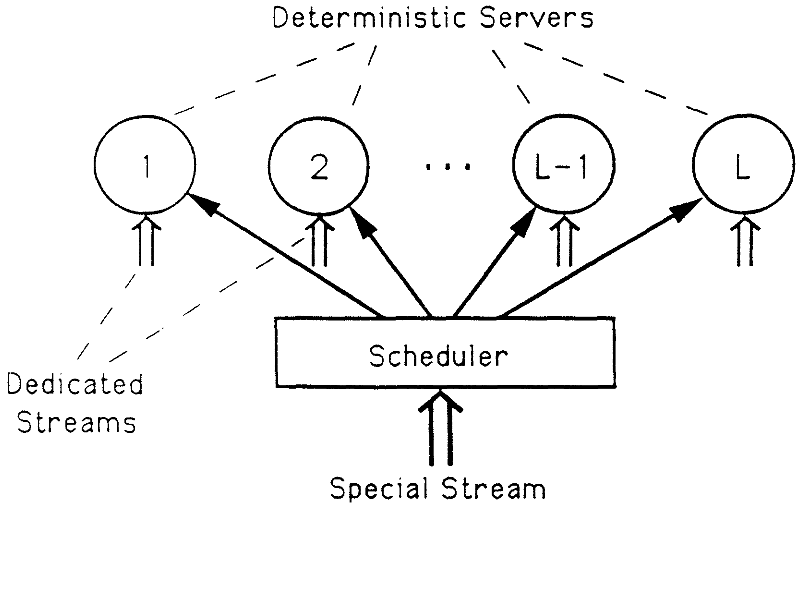

Figure 3: Server allocation in the presence of dedicated streams.

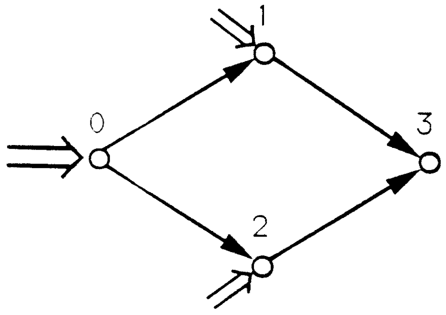

Figure 4

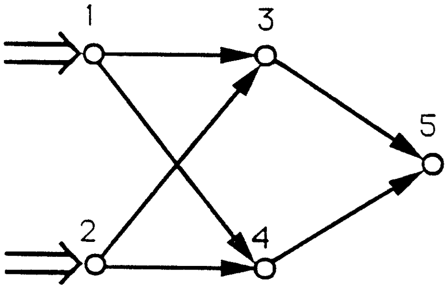

Figure 5