Queueing Systems, 6 (1990) 155–172

DOMINANCE RELATIONS IN POLLING SYSTEMS

Hanoch LEVY

Computer Science Department, The Raymond Beverly Sackler Faculty of Exact Science, Tel-Aviv University, Tel-Aviv 69978, Israel

Moshe SIDI

Electrical Engineering Department, Technion – Israel Institute of Technology, Haifa 32000, Israel

Onno J. BOXMA

Centre for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Received 10 February 1989; revised 20 October 1989

Abstract

In this paper we compare several service disciplines commonly used in polling systems. We present a sample path comparison which allows us to evaluate the efficiency of the different policies based on the *total* amount of work found in the system *at any time*. The analysis is carried out for a large variety of polling schemes under fairly general conditions and can be used to construct a hierarchy of the different service schemes.

Keywords: Stochastic comparison, dominance relations, polling systems.

1. Introduction and model description

It is commonly perceived that certain service disciplines in polling systems are more efficient than others. For instance, the exhaustive discipline is considered to be more efficient than the gated discipline. Nevertheless, this perception has been based mostly on experience and common sense and less on analysis. Analytic results regarding the comparison of the different service disciplines have been limited to the *expected value* of some performance measures. These comparisons have been of two types: (1) A comparison based on the *expected amount of work* present in the system (in steady state) can be carried out via the work decomposition results presented in Boxma and Groenendijk [3] and in Boxma [2]; (2) comparison based on the *expected delay* in *fully symmetric* systems can be applied for certain disciplines (see, e.g., Takagi [16] for cyclic polling and Kleinrock and Levy [7] for random polling).

The goal of this paper is to extend those results and to conduct such comparisons with respect to the *total* amount of unfinished work found in the system *at any time*; in contrast, the previous comparisons have been based on the *expected value* (evaluated over *all* customers) of the performance measures examined. Our approach is to conduct a sample path comparison from which the desired results immediately follow. The analysis is used to compare different service disciplines, in a variety of polling systems and under fairly general conditions, and to build some hierarchy of dominance among these disciplines. Specifically, we prove that some policies *dominate* others by being more *efficient*, in the sense that the *total* amount of unfinished work found in the system *at any time* (and thus by *any arriving customer*) when one policy is employed is smaller than when another policy is employed. An important result of our comparison is that the *exhaustive policy*, in which the server does not leave a queue until consuming all work in the queue, dominates *any* other service policy. A detailed description of the policies considered in this paper appears in section 2.

The underlying model we use in this paper is rather general. We consider a system with N queues served by a single server. Let A_l ($l \geqslant 1$) denote the arrival epoch of the lth customer, let B_l ($l \geqslant 1$) denote its service time and Q_l ($l \geqslant 1$, $l \leqslant Q_l \leqslant N$) denote the queue that this customer joins. We do not impose any restrictions upon the process (A_l, B_l, Q_l, $l \geqslant 1$). The single server moves among the queues according to some order and serves the customers of the queues according to the specified service discipline. We do not impose any restriction upon the order in which the server polls the different queues. Thus, the comparison presented applies to all orders mentioned in the literature – these include the cyclic polling (see e.g., Takagi [16]), the polling according to polling tables (Eisenberg [4] and Baker and Rubin [1]), and the random polling order (Kleinrock and Levy [7]). Finally, we do not restrict the order in which customers are served within a queue, as long as the selection of a customer for service is done independently of its particular service time. Yet, we do assume that a customer that enters the system does not leave it until receiving its required service.

To facilitate the presentation, it is convenient to sequentially index the server visits of the queues. Thus, these visits are indexed by $i = 1, 2, 3, \ldots$. The period during which the server serves the customers (in the ith visit) is called the ith *service period*. We let $I(i) \in \{1, 2, \ldots, N\}$ be the type of the ith visit (namely, this is the index of the visited queue). A visit to a queue may take zero time if the queue is empty when polled.

To complete the description, we note that switching from one queue to another may require some time from the server. This is called the *switch-over period*. The switch-over period succeeding the ith visit is also indexed i (even when it has zero length), and its duration is denoted by S_i. The difference in workloads under two different policies is obviously influenced by the lengths of the switch-over periods. In particular, in the case of zero length switch-over periods, all service disciplines lead to the same amount of total work (work conservation).

The only restrictions imposed on the service policies considered in this paper are as follows:

(1) Work conservation; in the context of this paper work conservation means that, apart from switching, the server does not create or destroy work, and when serving, his rate of service is constant.
(2) The server does not wait idling in a queue. Once the service of a queue is completed, the server switches to the next queue to be served. If this restriction is violated, one can easily construct counterexamples to the dominance relations to be presented below.

Finally, a word about recent related literature. In an independent study, Tedijanto [17] derived a similar comparison for single server queueing systems with server vacations.

The rest of the paper is organized as follows. In section 2 we describe the service policies that we consider in this paper. In sections 3 and 4 we formulate some mathematical properties of various service disciplines and prove the main comparison theorems. In section 5 we construct the hierarchy among the various service disciplines. Discussion of the results is provided in section 6.

2. Service policies

Most service policies can be classified into two classes: *gated* policies and *exhaustive* policies. A gated-type policy is characterized by the following property: Whenever the server polls a queue, the only customers that are considered for service during this visit of the queue are those customers that were present at the polled queue at the polling instant. Exhaustive-type policies are characterized by the fact that customers arriving at the queue while it is being served can also be served during the current visit.

There are several well-known variations of gated-type and exhaustive-type policies (see, e.g., Takagi [16]). The most well-known are the *pure* policies. In a pure gated policy *all* customers that are present at a queue when it is polled are served before the server switches from the queue. In the pure exhaustive policy the server switches from the queue only when there are *no more* customers to serve in that queue.

An important class of service policies consists of *limited* policies that can be viewed as variations of the gated and the exhaustive policies (see, e.g., Fuhrmann and Wang [5]). Limited policies are similar to pure policies except that the server does not serve more than a pre-specified number of customers (the limit) in a single visit. In other words, the server applies the pure policy, but switches from the queue if the number of customers already served during the current visit reaches the limit. Stochastic bounds for vacation models with limited service have been derived by Servi and Yao [12].

The pure and the limited policies are deterministic policies. There are recent variations of service policies which are *stochastic*. Some of the stochastic policies are essentially limited policies where the limit is probabilistically chosen at the beginning of every server visit to the queue. The probability distribution of the limit can be arbitrary, and may possibly depend on the number of customers at the queue when it is polled. For instance, in *Bernoulli* policies (Keilson and Servi [8], Servi [11]) the distribution of the limit is (shifted) geometric with some parameter p (in this specific case the probability distribution *does not* depend on the number of customers found in the queue when it is polled). Specifically, with the Bernoulli-exhaustive policy, the server decides to continue to serve the queue with probability p or to exit the queue with probability $1-p$ after each completion of service until there are no more customers in the queue. The Bernoulli-gated policy is similar except that the server does not serve during a single visit more than the number of customers he found in the queue when the queue was polled.

Another stochastic policy is the *binomial-gated* policy (Levy [9]) which is a degenerate limited policy in which the limit is identical to the actual number of customers served during the service period. The number of customers served is distributed according to the binomial distribution with parameters $0 < p \leqslant 1$ and X, where X is the number of customers present at the queue at the polling instant. An exhaustive version of the binomial-gated policy is the *binomial-exhaustive* (or fractional-exhaustive) policy (Levy [10]). In this policy the server "counts" the customers present at the queue at the polling instant and those arriving at the queue during the service period, and according to this total number (denote it Y) determines the number of customers to be served. Specifically, the actual number served is determined by performing Y Bernoulli experiments and summing their outcomes. This determination can be updated dynamically as Y increases during the service period (due to new arrivals) by performing an additional Bernoulli experiment whenever a new customer arrives. It is interesting to note that this policy cannot be viewed as a limited policy.

There are other service policies that are not of a limited type. For instance, consider an exhaustive policy in which the server serves the queue until the number of customers present in the queue is reduced by a pre-specified number (relatively to the number of customers present in the queue when it was polled). An example of such a policy is the *semi-exhaustive* policy (Takagi [15]) in which the pre-specified number is one.

3. Gated-type policies

In this section we consider gated-type policies. A gated-type policy $\mathbf{f}$ is formally defined as an infinite sequence of functions $\mathbf{f} = \{f_i(x)\}_{i=1,2,\ldots}$ (with $f_i(x) \geqslant 0$ for $x \geqslant 0$), such that $f_i(x)$ is the number of customers served in the ith

service period of the system, as a function of the number of customers (x) present in the visited queue when it is polled. For instance, $f_i(x) = x$ for a pure gated policy and $f_i(x) = \min\{x, l_i\}$ for a gated-limited policy where l_i is the limit during the ith visit.

We say that a policy $\mathbf{f}$ is a *monotonic policy* if for all i, $f_i(x) \geqslant f_i(y)$ for all $x \geqslant y$, namely, $f_i(x)$ is a monotonically non-decreasing function (for all i). A function $f_i(\cdot)$ is called a *contraction* if for every $x \geqslant y$, $f_i(x) - f_i(y) \leqslant x - y$. A policy $\mathbf{f}$ is called a *contractive policy* if for all $i \geqslant 1$, $f_i(\cdot)$ is a contraction. It is easy to see that both the pure-gated and the limited-gated policies are monotonic and contractive policies.

Let $\mathbf{f}$ be a gated-type policy and let us consider a sample path of the evolution of the system under this policy. Let $\tau_{\mathbf{f}}^i$ and $t_{\mathbf{f}}^i$ be the epochs at which the ith service period starts and ends, respectively. Let $L_{\mathbf{f}}^n(t)$ be the number of type-n customers (customers that arrived at station n) already served by time t. Let $C_{\mathbf{f}}^n(t)$ be the number of type-n customers present at time t in the system. Assuming that station n is served during the ith service period, we have that $C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i)$ is the number of customers which are *candidates* for service during the ith service period. This implies that the number of customers served during the ith service period is $f_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i))$.

LEMMA 1

Let $\mathbf{f} = \{f_i(\cdot)\}_{i=1,2,\ldots}$ and $\mathbf{g} = \{g_i(\cdot)\}_{i=1,2\ldots}$ be two gated-type policies that consider customers of a certain queue in FCFS order (see section 6 for other orders). Assume that the system is empty at $t = 0$ and that the two policies operate with the same realizations of the processes (A_l, B_l, Q_l, $l \geqslant 1$) and (S_i, $i \geqslant 1$) and the same realization of the polling order. If (i) $f_i(x) \geqslant g_i(x)$ for every $i \geqslant 1$ and $x \geqslant 0$, and (ii) $\mathbf{g}$ is a monotonic and contractive policy, then, for every $i \geqslant 1$, we have,

(1) $L_{\mathbf{f}}^k(t_{\mathbf{f}}^i) \geqslant L_{\mathbf{g}}^k(t_{\mathbf{g}}^i) \qquad 1 \leqslant k \leqslant N,$

(2) $t_{\mathbf{f}}^i \geqslant t_{\mathbf{g}}^i.$

Proof

The proof proceeds by induction. For $i = 1$ the proof is trivial. Assuming correctness for the $(i-1)$st service period, we show the correctness for the ith service period. We assume that queue n is served during the ith service period. For type-k customers ($k \neq n$), the induction step that leads to (1) is trivial since these customers are not served during the ith service period.

Let $A^n(t)$ be the number of type-n customers arriving to the system by time t. Then for type-n customers we observe that the number of candidate customers at the beginning of the ith service period under the two policies is given by

$$C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i) = A^n(\tau_{\mathbf{f}}^i) - L_{\mathbf{f}}^n(t_{\mathbf{f}}^{i-1}), \tag{3.1a}$$

$$C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^i) = A^n(\tau_{\mathbf{g}}^i) - L_{\mathbf{g}}^n(t_{\mathbf{g}}^{i-1}). \tag{3.1b}$$

The inductive assumption $t_{\mathbf{f}}^{i-1} \geqslant t_{\mathbf{g}}^{i-1}$ implies $\tau_{\mathbf{f}}^{i} \geqslant \tau_{\mathbf{g}}^{i}$, since the duration of S_{i-1} is identical under the two policies. Thus $A^n(\tau_{\mathbf{f}}^{i}) \geqslant A^n(\tau_{\mathbf{g}}^{i})$, and (3.1a, b) imply that

$$C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i}) - C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^{i}) \geqslant L_{\mathbf{g}}^n(t_{\mathbf{g}}^{i-1}) - L_{\mathbf{f}}^n(t_{\mathbf{f}}^{i-1}). \tag{3.2}$$

We observe that the number of customers served during the ith service period under policy **f** and policy **g** is $f_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i}))$ and $g_i(C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^{i}))$, respectively. Therefore,

$$L_{\mathbf{f}}^n(t_{\mathbf{f}}^{i}) = L_{\mathbf{f}}^n(t_{\mathbf{f}}^{i-1}) + f_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i})), \tag{3.3a}$$

$$L_{\mathbf{g}}^n(t_{\mathbf{g}}^{i}) = L_{\mathbf{g}}^n(t_{\mathbf{g}}^{i-1}) + g_i(C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^{i})). \tag{3.3b}$$

From assumption (i) in the lemma, we have

$$f_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i})) \geqslant g_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i})). \tag{3.4}$$

We now distinguish between two cases:

Case 1. $C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i}) > C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^{i})$: In this case we have,

$$\begin{aligned} L_{\mathbf{f}}^n(t_{\mathbf{f}}^{i}) - L_{\mathbf{g}}^n(t_{\mathbf{g}}^{i}) &= L_{\mathbf{f}}^n(t_{\mathbf{f}}^{i-1}) - L_{\mathbf{g}}^n(t_{\mathbf{g}}^{i-1}) + f_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i})) - g_i(C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^{i})) \\ &\geqslant f_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i})) - g_i(C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^{i})) \geqslant f_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i})) - g_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i})) \geqslant 0, \end{aligned} \tag{3.5}$$

where the first inequality follows from the induction hypothesis, the second inequality follows from the fact that **g** is a monotonic policy, and the last inequality follows from (3.4).

Case 2. $C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i}) \leqslant C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^{i})$: In this case assumption (ii) (**g** is a contractive policy) yields,

$$g_i(C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^{i})) - g_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i})) \leqslant C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^{i}) - C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i}).$$

Thus,

$$g_i(C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^{i})) - f_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i})) \leqslant C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^{i}) - C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i}). \tag{3.6}$$

In words, the difference between the number of customers served under the two policies is bounded by the difference between the number of candidates. Therefore,

$$\begin{aligned} L_{\mathbf{f}}^n(t_{\mathbf{f}}^{i}) - L_{\mathbf{g}}^n(t_{\mathbf{g}}^{i}) &= L_{\mathbf{f}}^n(t_{\mathbf{f}}^{i-1}) - L_{\mathbf{g}}^n(t_{\mathbf{g}}^{i-1}) + f_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i})) - g_i(C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^{i})) \\ &\geqslant L_{\mathbf{f}}^n(t_{\mathbf{f}}^{i-1}) - L_{\mathbf{g}}^n(t_{\mathbf{g}}^{i-1}) + C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^{i}) - C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^{i}) \geqslant 0, \end{aligned} \tag{3.7}$$

where the first inequality follows from (3.6) and the second from (3.2). Thus we completed the proof of (1), and claim (2) is directly implied by (1). □

We now turn to the main result of this section.

THEOREM 1

Let **f** and **g** be two gated-type policies. Let $U_{\mathbf{f}}(t)$ and $U_{\mathbf{g}}(t)$ be the total amount of unfinished work at time t in the system when the corresponding policy is employed. Under the assumptions of Lemma 1 we have that $U_{\mathbf{f}}(t) \leqslant U_{\mathbf{g}}(t)$ for every $t \geqslant 0$.

Proof

Assume that t falls in the ith visit in the system when policy **f** is employed. Then since $t_{\mathbf{f}}^{i-1} \geqslant t_{\mathbf{g}}^{i-1}$ and the switch-over times are identical in both systems, we have that $\tau_{\mathbf{f}}^{i} \geqslant \tau_{\mathbf{g}}^{i}$. Let $I_{\mathbf{f}}(t)$ and $I_{\mathbf{g}}(t)$ be the total amount of time the server was idle (i.e., switching) during $(0, t)$ in the system when the corresponding policy is employed. Obviously, $I_{\mathbf{f}}(\tau_{\mathbf{f}}^{i}) = I_{\mathbf{g}}(\tau_{\mathbf{g}}^{i})$. Since the server in the system that employs policy **f** is busy in the interval $(\tau_{\mathbf{f}}^{i}, t)$, we must have $I_{\mathbf{f}}(t) \leqslant I_{\mathbf{g}}(t)$ and thus $U_{\mathbf{f}}(t) \leqslant U_{\mathbf{g}}(t)$.

Assume now that t falls in the ith switch-over period in the system when policy **f** is employed. Obviously $I_{\mathbf{f}}(t_{\mathbf{f}}^{i}) = I_{\mathbf{g}}(t_{\mathbf{g}}^{i})$. Also, from Lemma 1, $t_{\mathbf{g}}^{i} \leqslant t_{\mathbf{f}}^{i}$. Now, since the server in the system that employs policy f must be idle during the interval $(t_{\mathbf{g}}^{i}, t_{\mathbf{g}}^{i} + (t - t_{\mathbf{f}}^{i}))$, we have $I_{\mathbf{g}}(t) \geqslant I_{\mathbf{f}}(t)$, and thus $U_{\mathbf{g}}(t) \geqslant U_{\mathbf{f}}(t)$. □

4. Exhaustive-type policies

Exhaustive-type policies differ from gated-type policies in considering for service customers which arrived to the polled queue during the service period. The *service candidates* in an exhaustive scheme consist of the customers present at the queue at the polling instant as well as those who arrived during the service period. Note that the number of service candidates in these systems is dynamically changing during the service period. In this section we present a framework for the comparison of different exhaustive-type policies similar to the framework developed for gated-type policies in the previous section. In addition, we conclude that the pure exhaustive policy dominates *any* other policy that fulfills the restrictions introduced in section 1.

Following the notation used in the analysis of gated systems, let **f** be a service policy and $L_{\mathbf{f}}^{n}(t)$ be the number of type-n customers served by time t. Let B_{j}^{n} be the service time of the jth type-n customer. Let $X_{\mathbf{f}}^{n}(t)$ be the number of type-n customers present at queue n at time t, and let $A^{n}(t_1, t_2)$ be the number of customers arriving to this queue during the period (t_1, t_2). Let us concentrate on the ith service period of the system, which starts at $\tau_{\mathbf{f}}^{i}$ and ends at $t_{\mathbf{f}}^{i}$, and assume that queue n is served in this period. For any $\tau_{\mathbf{f}}^{i} \leqslant t \leqslant t_{\mathbf{f}}^{i}$, the number of candidates, $C_{\mathbf{f}}^{n}$, is a function of both $\tau_{\mathbf{f}}^{i}$ and t and obeys $C_{\mathbf{f}}^{n}(\tau_{\mathbf{f}}^{i}, t) = X_{\mathbf{f}}^{n}(\tau_{\mathbf{f}}^{i}) + A^{n}(\tau_{\mathbf{f}}^{i}, t)$. In other words, the number of candidates considered between $\tau_{\mathbf{f}}^{i}$ and t is equal to the number of those who are present in the queue at the polling instant plus the number of those arriving during $(\tau_{\mathbf{f}}^{i}, t)$.

Similarly to gated-type policies, an exhaustive-type policy $\mathbf{f}$ is defined by an infinite sequence of service functions $\mathbf{f} = \{f_i(\cdot)\}_{i=1,2,\ldots}$. The exhaustive-type policy differs from the gated-type policy by *continuously* applying this function during the service period on the *dynamically changing parameter* $C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i, t)$.

We consider exhaustive policies which are characterized by the following two conditions:

(a) $$\sum_{j=L_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i)+1}^{L_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i)+f_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i,\, t_{\mathbf{f}}^i))} B_j^n = t_{\mathbf{f}}^i - \tau_{\mathbf{f}}^i.$$

(b) For any $\tau_{\mathbf{f}}^i < t < t_{\mathbf{f}}^i$: $$\sum_{j=L_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i)+1}^{L_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i)+f_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i,\, t))} B_j^n > t - \tau_{\mathbf{f}}^i,$$

where $C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i, t) = X_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i) + A^n(\tau_{\mathbf{f}}^i, t)$. Furthermore, we require that $\mathbf{f}$ be monotonic.

Condition (a) states that at the end of the service period the amount of work associated with the customers selected for service is exactly identical to the amount of service granted by the server during this period; thus the queue has no more customers to be served and the service period ends. Condition (b) states that at any moment t during the service period, condition (a) is not met; namely, the amount of work associated with the candidates that are selected for service is larger than the time the server has served so far.

Remark 4.1: The monotonicity of $\mathbf{f}$ is required for the consistency of the exhaustive policy. Non-monotonic policies can lead to situations in which a decision made at some time t of serving k customers during the service period will be violated by a later decision of serving $k' < k$ customers.

Remark 4.2: Note that conditions (a) and (b) characterize, as a degenerate case, gated-type policies as well. For these policies, however, $C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i, t) = X_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i)$ is not dynamically changing and is not a function of t. For this reason, the analysis of gated-type policies was somewhat simpler.

It is easy to verify that all common exhaustive policies meet these conditions. Specifically, service policies falling into this category are the (pure) exhaustive, the limited-exhaustive, the Bernoulli and the binomial-exhaustive policies (not included in this category is the semi-exhaustive policy which falls in a more general category, discussed in section 5). For example, the service function associated with the pure-exhaustive policy is $f_i(x) = x$, and the one associated with the limited-exhaustive (with limit l_i) is $f_i(x) = \min\{l_i, x\}$.

Having formalized the behavior of exhaustive-type policies, we may now apply an analysis similar to that from section 3 to exhaustive-type policies. We keep the presentation and notation the same and prove the following lemma:

LEMMA 2

Let $\mathbf{f} = \{f_i(\cdot)\}_{i=1,2,\ldots}$ and $\mathbf{g} = \{g_i(\cdot)\}_{i=1,2,\ldots}$ be two exhaustive-type policies that consider customers of a certain queue in FCFS order (see section 6 for other

orders). Assume that the system is empty at $t=0$ and that the two policies operate with the same realizations of the processes $(A_l,\ B_l,\ Q_l,\ l \geqslant 1)$ and $(S_i,\ i \geqslant 1)$ and the same realization of polling order. If (i) $f_i(x) \geqslant g_i(x)$ for every $i \geqslant 1$ and $x \geqslant 0$, and (ii) **g** is a contractive policy, then, for every $i \geqslant 1$ we have,

(1) $L_{\mathbf{f}}^k(t_{\mathbf{f}}^i) \geqslant L_{\mathbf{g}}^k(t_{\mathbf{g}}^i) \qquad 1 \leqslant k \leqslant N.$

(2) $t_{\mathbf{f}}^i \geqslant t_{\mathbf{g}}^i.$

Proof

The proof is by induction, where we assume correctness of both claims for $i-1$ and prove them for i. The only non-trivial step is to show the correctness of the claims for the queue served during the ith service period, which we assume is queue n.

By way of contradiction assume that $t_{\mathbf{f}}^i < t_{\mathbf{g}}^i$. From the inductive assumption, $\tau_{\mathbf{g}}^i \leqslant \tau_{\mathbf{f}}^i$ and thus system **g** must be actively serving queue n at $t_{\mathbf{f}}^i$. Now, we have

$$C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i,\ t_{\mathbf{f}}^i) = \left[A^n(0,\ \tau_{\mathbf{f}}^i) - L_{\mathbf{f}}^n(t_{\mathbf{f}}^{i-1})\right] + A^n(\tau_{\mathbf{f}}^i,\ t_{\mathbf{f}}^i), \tag{4.1}$$

$$C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^i,\ t_{\mathbf{f}}^i) = \left[A^n(0,\ \tau_{\mathbf{g}}^i) - L_{\mathbf{g}}^n(t_{\mathbf{g}}^{i-1})\right] + A^n(\tau_{\mathbf{g}}^i,\ t_{\mathbf{f}}^i), \tag{4.2}$$

in which the term in brackets represents the number of customers present at queue n at the polling instant and the second term represents the number of arrivals between the beginning of the ith service period and $t_{\mathbf{f}}^i$. Noting that

$$A^n(0,\ \tau_{\mathbf{f}}^i) + A^n(\tau_{\mathbf{f}}^i,\ t_{\mathbf{f}}^i) = A^n(0,\ \tau_{\mathbf{g}}^i) + A^n(\tau_{\mathbf{g}}^i,\ t_{\mathbf{f}}^i),$$

we obtain

$$C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^i,\ t_{\mathbf{f}}^i) - C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i,\ t_{\mathbf{f}}^i) = L_{\mathbf{f}}^n(t_{\mathbf{f}}^{i-1}) - L_{\mathbf{g}}^n(t_{\mathbf{g}}^{i-1}) \geqslant 0, \tag{4.3}$$

in which the inequality results from the inductive assumption. Now, from condition (i) in the lemma,

$$f_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i,\ t_{\mathbf{f}}^i)) \geqslant g_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i,\ t_{\mathbf{f}}^i)), \tag{4.4}$$

and from condition (ii) in the lemma (contraction),

$$g_i(C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^i,\ t_{\mathbf{f}}^i)) - g_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i,\ t_{\mathbf{f}}^i)) \leqslant C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^i,\ t_{\mathbf{f}}^i) - C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i,\ t_{\mathbf{f}}^i). \tag{4.5}$$

Thus, from (4.4) and (4.5)

$$g_i(C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^i,\ t_{\mathbf{f}}^i)) - f_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i,\ t_{\mathbf{f}}^i)) \leqslant C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^i,\ t_{\mathbf{f}}^i) - C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i,\ t_{\mathbf{f}}^i),$$

and together with (4.3) we get:

$$L_{\mathbf{g}}^n(t_{\mathbf{g}}^{i-1}) + g_i(C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^i,\ t_{\mathbf{f}}^i)) \leqslant L_{\mathbf{f}}^n(t_{\mathbf{f}}^{i-1}) + f_i(C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i,\ t_{\mathbf{f}}^i)) \tag{4.6}$$

Now, let $T_{\mathbf{f}}^k(t)$ be the amount of time spent by the server under policy **f** on serving queue k during $(0,\ t)$ and $I_{\mathbf{f}}(t)$ be the total amount of time the server has

been idle during $(0, t)$. Similarly, define $T_{\mathbf{g}}^k(t)$ and $I_{\mathbf{g}}(t)$. From the inductive assumption and since under policy **g** the server is actively serving queue n at $t_{\mathbf{f}}^i$, it is clear that for any $k \neq n$:

$$T_{\mathbf{g}}^k(t_{\mathbf{f}}^i) \leqslant T_{\mathbf{f}}^k(t_{\mathbf{f}}^i) \tag{4.7}$$

and, in addition

$$I_{\mathbf{g}}(t_{\mathbf{f}}^i) = I_{\mathbf{f}}(t_{\mathbf{f}}^i). \tag{4.8}$$

From (4.7), (4.8) and work conservation we have $T_{\mathbf{g}}^n(t_{\mathbf{f}}^i) \geqslant T_{\mathbf{f}}^n(t_{\mathbf{f}}^i)$. However, (4.6) and the fact that the server is actively serving under **g** at $t_{\mathbf{f}}^i$ imply (through conditions (a) and (b)) that $T_{\mathbf{g}}^n(t_{\mathbf{f}}^i) < T_{\mathbf{f}}^n(t_{\mathbf{f}}^i)$. Thus, by way of contradiction, we proved $t_{\mathbf{f}}^i \geqslant t_{\mathbf{g}}^i$.

The proof of (1) now immediately follows from (4.6), and from the fact that $t_{\mathbf{f}}^i \geqslant t_{\mathbf{g}}^i$ which implies that $C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^i, t_{\mathbf{f}}^i) \geqslant C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^i, t_{\mathbf{g}}^i)$. □

We now state the main result of comparing different exhaustive-type policies. The proof is similar to the proof of Theorem 1.

THEOREM 2

Let **f** and **g** be two exhaustive-type policies. Let $U_{\mathbf{f}}(t)$ and $U_{\mathbf{g}}(t)$ be the total amount of unfinished work at time t in the system when the corresponding policy is employed. Under the assumptions of Lemma 2 we have that $U_{\mathbf{f}}(t) \leqslant U_{\mathbf{g}}(t)$ for every $t \geqslant 0$.

An important result, which can be derived using this general framework, is that the pure-exhaustive system is more efficient than any other service policy, as stated in the following theorem:

THEOREM 3

Let **f** be the pure exhaustive policy and let **g** be *any* policy that fulfills the restrictions introduced in section 1. Let $U_{\mathbf{f}}(t)$ and $U_{\mathbf{g}}(t)$ be the total amount of unfinished work at time t in the system when the corresponding policy is employed. Then $U_{\mathbf{f}}(t) \leqslant U_{\mathbf{g}}(t)$ for every $t \geqslant 0$.

The proof of this theorem follows along the same lines as the proof of Lemma 2 and Theorem 2 and the main argument here is that at $t_{\mathbf{f}}^i$ all work arriving during $(0, t_{\mathbf{f}}^i)$ for queue n has been completed. A detailed proof appears in Levy, Sidi and Boxma [14].

Remark 4.3: The formulation presented above allows us to compare between gated-type and exhaustive-type policies. Specifically, if **e** and **g** are exhaustive-type and gated-type policies which utilize the same function sequence **f**, then it can be shown that $U_{\mathbf{e}}(t) \leqslant U_{\mathbf{g}}(t)$ for every $t \geqslant 0$. This result can be proved along the same lines as Lemma 2 and Theorem 2 and using the fact that $C_{\mathbf{g}}^n(\tau_{\mathbf{g}}^i, t) = X_{\mathbf{g}}^n(\tau_{\mathbf{g}}^i)$ while $C_{\mathbf{e}}^n(\tau_{\mathbf{e}}^i, t) = X_{\mathbf{e}}^n(\tau_{\mathbf{e}}^i) + A^n(\tau_{\mathbf{e}}^i, t)$.

5. Constructing a hierarchy of the service policies

In this section we use the framework developed in the previous sections to derive dominance relations between the different service policies. First, in section 5.1 we deal with the common deterministic policies. Second, in section 5.2 we consider the recently introduced stochastic policies. Then, in section 5.3 we discuss policies which require generalization of the framework developed in sections 3 and 4. Lastly (in section 5.4), we outline the dominance relations.

5.1. DETERMINISTIC POLICIES

The most common service policies are deterministic and are the pure-exhaustive, the pure-gated and the limited (gated or exhaustive), which are defined by the following service functions:

1. Pure-gated and pure-exhaustive: $f_i(x) = x$.
2. Limited-gated and limited-exhaustive (with parameter l_i): $f_i(x) = \min\{l_i, x\}$.

Note that in each of the two categories, the gated-type policy and the exhaustive-type policy are defined by the same service function, and the distinction between the two is in the definition of the service candidates ($C_{\mathbf{f}}^n$).

Using the results derived in sections 3 and 4 it is easy to construct dominance relations among these deterministic policies. To this end, note that both $f_i(x) = x$ and $f_i(x) = \min\{l_i, x\}$ are monotonically non-decreasing contractions and thus we can apply Theorems 1 and 2 to them. To conduct the comparison, let **f** and **g** be two *limited-gated* policies, such that $f_i(x) = \min\{l_{\mathbf{f}}^i, x\}$ and $g_i(x) = \min\{l_{\mathbf{g}}^i, x\}$ where $l_{\mathbf{f}}^i$ and $l_{\mathbf{g}}^i$ are the service limits used by **f** and **g** in the ith service period. Note, that if $l_{\mathbf{f}}^i \geqslant l_{\mathbf{g}}^i$, then $f_i(x) \geqslant g_i(x)$ for every $x \geqslant 1$. Thus, we may conclude from Theorem 1 that if $l_{\mathbf{f}}^i \geqslant l_{\mathbf{g}}^i$ for every $i \geqslant 1$, then **f** is more efficient than **g**, in the sense that $U_{\mathbf{f}}(t) \leqslant U_{\mathbf{g}}(t)$ for any $t \geqslant 0$ (where $U_{\mathbf{f}}(t)$ and $U_{\mathbf{g}}(t)$ represent the unfinished work in the two systems at time t).

This result obviously implies that if **f** and **g** are two limited-gated policies which use the limits $l_{\mathbf{f}}^n$ and $l_{\mathbf{g}}^n$ for a visit of queue n and if they use the same polling order, then **f** is more efficient than **g**, if $l_{\mathbf{f}}^n \geqslant l_{\mathbf{g}}^n$ for $1 \leqslant n \leqslant N$.

The same analysis can be repeated to derive similar relations for limited-exhaustive policies. In addition, the pure policies can be viewed as limited policies whose limit is infinity. This directly implies the dominance of the pure policies on any limited (or partially limited) policies.

5.2. STOCHASTIC POLICIES

The stochastic policies can be defined by the following service functions:

(1) Bernoulli-gated and Bernoulli-exhaustive (with parameter p_i): $f_i(x) = \min\{y, x\}$, for $y \geqslant 1$, with probability $(1 - p_i)p_i^{(y-1)}$.

(2) Binomial-gated and binomial-exhaustive (with parameter p_i): $f_i(x) = y$, for $0 \leqslant y \leqslant x$, with probability $\binom{x}{y} p_i^y (1 - p_i)^{(x-y)}$.

Note that, as in the case of the deterministic policies, the exhaustive-type policies and the gated-type policies can be defined by the same service functions and differ only in the definition of the service candidates.

The comparison between the stochastic policies requires some additional analysis. We observe that the functions $f_i(x)$ in all cases above are determined by first performing an infinite sequence of independent Bernoulli experiments (with parameter p_i) and recording them in an infinite vector $\mathbf{v}$, and then by applying a simple *deterministic* algebraic function on the vector. The distinction between the different policies is in the deterministic function utilized.

In the case of the Bernoulli policies the value of the deterministic function, denoted by $d_i(\mathbf{v})$, is the index of the first zero in the vector $\mathbf{v}$. Accordingly, the Bernoulli-gated policy will serve at most $d_i(\mathbf{v})$ customers in a gated fashion (i.e. the number of customers served will be equal to the minimum of $d_i(\mathbf{v})$ and x, where x is the number of customers present at the queue at the polling instant). Similarly, the Bernoulli-exhaustive policy will serve at most $d_i(\mathbf{v})$ customers in an exhaustive fashion.

In the case of the binomial-gated and binomial-exhaustive policies the value of the deterministic function depends on the number of customers (x), and is given by the sum of the first x components of the vector.

The comparison of the stochastic policies, therefore, reduces to a comparison of the Bernoulli vectors and the algebraic functions which operate on them. The comparison of two vectors is conducted component-wise, namely, if $\mathbf{v}$ and $\mathbf{u}$ are vectors whose components are $v(j)$, $j = 1, 2, \ldots$ and $u(j)$, $j = 1, 2, \ldots$, then we say that $\mathbf{v} \geqslant \mathbf{u}$ if $v(j) \geqslant u(j)$ for every $j \geqslant 1$.

Let $\mathbf{f}$ and $\mathbf{g}$ be two stochastic policies which use the parameters $p_\mathbf{f}$ and $p_\mathbf{g}$ respectively to construct the value of their stochastic vectors, $\mathbf{v}_\mathbf{f}$ and $\mathbf{v}_\mathbf{g}$. Let $v_\mathbf{f}(j)$ and $v_\mathbf{g}(j)$ be the jth components of these vectors. The values of $v_\mathbf{f}(j)$ and $v_\mathbf{g}(j)$ are coupled as follows: We randomly (with uniform distribution) select a point $z(j)$ from the interval [0, 1]. We set the value of $v_\mathbf{f}(j)$ to 1 if and only if $z(j) \leqslant p_\mathbf{f}$ and the value of $v_\mathbf{g}(j)$ to 1 if and only if $z(j) \leqslant p_\mathbf{g}$; otherwise these values are set to 0. This guarantees two properties:

(1) In each of the vectors the value of each component is determined by a Bernoulli experiment with the proper parameter. This value is independent of the value of the other components in the vector.

(2) If $p_\mathbf{f} \geqslant p_\mathbf{g}$, then $v_\mathbf{f}(j) \geqslant v_\mathbf{g}(j)$ for every j.

Note that the only stochastic part of this mechanism is the selection of an infinite vector $\mathbf{z}$ (whose components are $z(j)$), which is completely independent of the actual policy. The actual policy can then be viewed as a deterministic policy applied to the vector $\mathbf{z}$.

Once this coupling mechanism is established we may use it to conduct the comparison of different policies. We assume that an infinite vector $\mathbf{z}_i$ is stochastically determined for the ith visit of the system and that each policy now applies

the deterministic function corresponding to the ith visit on the vector $\mathbf{z}_i$ in order to determine the number of customers served during the visit.

To be more specific let us first consider two Bernoulli-gated policies **f** and **g**, which use the parameters $p_{\mathbf{f}}^i$ and $p_{\mathbf{g}}^i$ in their ith visit. From property (2) above, it is clear that if $p_{\mathbf{f}}^i \geqslant p_{\mathbf{g}}^i$ then the Bernoulli vectors determined by the policies, $\mathbf{v}_{\mathbf{f}}^i$ and $\mathbf{v}_{\mathbf{g}}^i$, obey $\mathbf{v}_{\mathbf{f}}^i \geqslant \mathbf{v}_{\mathbf{g}}^i$. For this reason the index of the first zero in $\mathbf{v}_{\mathbf{g}}^i$ is not larger than that of $\mathbf{v}_{\mathbf{f}}^i$, and thus the values of the functions obey: $f_i(x) \geqslant g_i(x)$ for $i \geqslant 1$ and $x \geqslant 1$. Moreover, using this representation, it is also easy to see that both $f_i(x)$ and $g_i(x)$ are monotonically non-decreasing contractions. Thus, all the conditions specified in section 3 are met, and we conclude that if $p_{\mathbf{f}}^i \geqslant p_{\mathbf{g}}^i$ for every $i \geqslant 1$, then $U_{\mathbf{f}}(t) < U_{\mathbf{g}}(t)$ for every $t \geqslant 0$. The comparison of two Bernoulli-exhaustive policies is obviously similar.

Considering the binomial-gated policies, let **f** and **g** be policies which use the parameters $p_{\mathbf{f}}^i$ and $p_{\mathbf{g}}^i$ in their ith visit respectively. Again, if $p_{\mathbf{f}}^i \geqslant p_{\mathbf{g}}^i$, then $\mathbf{v}_{\mathbf{f}}^i \geqslant \mathbf{v}_{\mathbf{g}}^i$ and the sum of the first x components of $\mathbf{v}_{\mathbf{f}}^i$ is greater than or equal to the corresponding sum in $\mathbf{v}_{\mathbf{g}}^i$. Thus $f_i(x) \geqslant g_i(x)$ for every $x \geqslant 1$. Similarly to the Bernoulli policies, one can easily check these functions and verify that $f_i(x)$ and $g_i(x)$ are both monotonically non-decreasing contractions. Thus, applying Theorem 1 we have that if $p_{\mathbf{f}}^i \geqslant p_{\mathbf{g}}^i$ for every $i \geqslant 1$ then $U_{\mathbf{f}}(t) \leqslant U_{\mathbf{g}}(t)$ for every $t \geqslant 0$. The comparison of two binomial-exhaustive policies is obviously similar.

Other stochastic policies can be compared in a similar manner.

5.3. OTHER SERVICE POLICIES: GENERALIZATION OF THE FRAMEWORK

The framework presented in sections 3 and 4 classifies the service policies into two classes: gated-type policies in which the number of service candidates is the number of customers present at the queue at the polling instant, $X_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i)$, and exhaustive-type policies in which the number of service candidates is equal to the sum of $X_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i)$ and $A^n(\tau_{\mathbf{f}}^i, t_{\mathbf{f}}^i)$, the number of arrivals in $(\tau_{\mathbf{f}}^i, t_{\mathbf{f}}^i)$. These categories contain most of the common service policies but not all of them.

A simple generalization of the exhaustive-type class is a class in which the number of service candidates is a general function $C_{\mathbf{f}}^n$ of the variables $X_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i)$ and $A^n(\tau_{\mathbf{f}}^i, t)$ (rather than the simple addition of these two variables). Examples of such policies are the semi-exhaustive and a variation of the binomial-exhaustive policy. The latter policy, proposed by Groenendijk [6], uses the binomial distribution to select the number of customers out of the $X_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i)$ present at the polling instant to serve, and serves them as well as all the customers arriving during the service period $(A^n(\tau_{\mathbf{f}}^i, t_{\mathbf{f}}^i))$. This policy can be defined by $C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i, t) = y + A^n(\tau_{\mathbf{f}}^i, t)$ for $0 \leqslant y \leqslant X_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i)$ with probability

$$\binom{X_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i)}{y} p_i^y (1-p_i)^{X_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i)-y}$$

and $f_i(x) = x$. The semi-exhaustive policy can be defined by $C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i, t) = \max\{1, X_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i)\} + A^n(\tau_{\mathbf{f}}^i, t)$ and $f_i(x) = x$.

The approach used in sections 3 and 4 can be generalized to accommodate these cases as well as others using the broader definition of $C_{\mathbf{f}}^{n}$. We, therefore, may conclude that dominance relations for these classes can be derived similar to the relations derived in sections 5.1 and 5.2 above.

5.4. SUMMARY OF THE DOMINANCE RELATIONS

We summarize the dominance relations established in this paper as follows:

$$U_{\text{EXHAUSTIVE}}(t) \leqslant U_{\text{POLICY}}(t),$$

where "POLICY" is any arbitrary policy.

$$\begin{aligned}
&U_{\text{LIMITED-EXHAUSTIVE-m}}(t) \leqslant U_{\text{LIMITED-EXHAUSTIVE-l}}(t) && \mathbf{m} \geqslant \mathbf{l},\\
&U_{\text{LIMITED-GATED-m}}(t) \leqslant U_{\text{LIMITED-GATED-l}}(t) && \mathbf{m} \geqslant \mathbf{l},\\
&U_{\text{BERNOULLI-EXHAUSTIVE-p}}(t) \leqslant U_{\text{BERNOULLI-EXHAUSTIVE-q}}(t) && \mathbf{p} \geqslant \mathbf{q},\\
&U_{\text{BERNOULLI-GATED-p}}(t) \leqslant U_{\text{BERNOULLI-GATED-q}}(t) && \mathbf{p} \geqslant \mathbf{q},\\
&U_{\text{BINOMIAL-EXHAUSTIVE-p}}(t) \leqslant U_{\text{BINOMIAL-EXHAUSTIVE-q}}(t) && \mathbf{p} \geqslant \mathbf{q},\\
&U_{\text{BINOMIAL-GATED-p}}(t) \leqslant U_{\text{BINOMIAL-GATED-q}}(t) && \mathbf{p} \geqslant \mathbf{q}.
\end{aligned}$$

In all inequalities given above $\mathbf{p}$, $\mathbf{q}$, $\mathbf{l}$ and $\mathbf{m}$ are N-dimensional vectors. In addition, we have that an exhaustive-type policy dominates its gated-type counterpart when they use the same sequence of service functions $\mathbf{f}$.

6. Discussion

In this paper we presented a general framework for the comparison of different service policies in polling systems. The generality of the framework allows us to conduct the comparison for a large variety of service policies, not all of them mentioned above.

First, the dominance relations presented in section 5.1 above trivially extend to mixed systems (e.g., a system in which some of the stations are served according to the limited-exhaustive policy while others are served according to the limited-gated policy) in which the proper conditions hold. Second, newly designed policies can be compared to old ones by simply analyzing and comparing their service functions.

The analysis presented above was conducted under the assumption that the service order within each queue is FCFS. Nonetheless, the stochastic comparison of the different service policies does hold for any service order provided that the order does not depend on the customer service times. The application of the comparison to such policies (e.g., Last-Come-First-Served) is done by coupling the sample paths of the compared systems via proper selection of the actual

service times. Such a comparison will result in *stochastic dominance relations* identical to those derived above. (However, it will not be true any more that the comparison holds for any individual sample path.)

The conditions under which our comparison was conducted are much more general than those required for previous comparisons. The arrival processes are arbitrary, allowing general and correlated interarrival times (where the correlations can be between the arrivals to different queues). Moreover, customer routing (see Sidi and Levy [13]) is allowed. This means that customers need not leave the system after being served in their queue, and can be routed to other queues for further service. The durations of the switch-over periods may depend on each other as can the service times within each queue.

Unfortunately, this framework cannot be extended to compare the waiting times observed in polling systems. The reason is that there seems to exist no general rule by which one can compare different service policies according to a reasonable measure of waiting times. To demonstrate this let us examine the weighted mean waiting time of the system, namely $\sum_{i=1}^{N}\lambda_i E[W_i]/(\sum_{i=1}^{N}\lambda_i)$, where λ_i and W_i are the arrival rate to queue i and the waiting time at queue i, respectively. Consider a two-queue system in which the arrival rates are $\lambda_1 = 1000$, $\lambda_2 = 0.01$ and the mean service times are $b_1 = 10^{-6}$, $b_2 = 90$. Now consider two service regimes: the first serves both queues exhaustively and the second serves queue 1 exhaustively and queue 2 according to the limited-1 policy. The first regime will be superior according to the conservation law criterion, as well as according to the criterion provided in this work. Nevertheless, according to the weighted mean waiting time criterion, the second regime is obviously better (since it gives more attention to queue 1, at which the large majority of customers arrive). Obviously one can change the parameters of the system such that regime 1 will be better according to this criterion, and thus we conclude that such comparison needs to rely on the system parameters and is not a function only of the service strategies.

Finally, we stress the fact that our comparison is restricted to the amount of work in the system. Other performance considerations, like "fairness", may lead to the implementation of a less efficient policy in the actual system.

Appendix

Glossary of notation

The following is a list of the notations frequently used in the paper:

A_l – arrival epoch of the lth customer.
$A^n(t)$ – number of type-n customers arriving to the system by time t.
$A^n(t_1, t_2)$ – number of customers arriving at queue n during (t_1, t_2).
B_l – service time of the lth customer.
Q_l – queue that the lth customer joins.

$I(i)$ – index of the visited queue in the ith visit.
S_i – switch-over period succeeding the ith visit.
$\mathbf{f}, \mathbf{g}$ – service policies.
$\tau_{\mathbf{f}}^i$ – epoch at which the ith service period starts under policy $\mathbf{f}$.
$f_i(x)$ – number of customers served in the ith service period when x customers are present in the visited queue when it is polled.
$t_{\mathbf{f}}^i$ – epoch at which the ith service period ends under policy $\mathbf{f}$.
$L_{\mathbf{f}}^n(t)$ – number of type-n customers already served by time t under policy $\mathbf{f}$.
$C_{\mathbf{f}}^n(t)$ – number of type-n customers in the system at time t under policy $\mathbf{f}$.
$I_{\mathbf{f}}(t)$ – total amount of time the server was idle during $(0, t)$ under policy $\mathbf{f}$.
$U_{\mathbf{f}}(t)$ – total amount of unfinished work at time t in the system under policy $\mathbf{f}$.
$C_{\mathbf{f}}^n(\tau_{\mathbf{f}}^i, t)$ – number of candidates at time t $(\tau_{\mathbf{f}}^i \leqslant t \leqslant t_{\mathbf{f}}^i)$ at queue i.

Acknowledgement

We are indebted to the anonymous referees for several useful comments.

References

[1] J.E. Baker and I. Rubin, Polling with a general-service order table, IEEE Trans. Commun. COM-35, 3 (1987) pp. 283–288.
[2] O.J. Boxma, Workloads and waiting times in single-server systems with multiple customer classes, Draft, to appear in Queueing Systems (1989).
[3] O.J. Boxma and W.P. Groenendijk, Pseudo-conservation laws in cyclic queues, J. Appl. Probab. 24 (1987) 949–964.
[4] M. Eisenberg, Queues with periodic service and changeover time, Oper. Res. 20 (1972) 440–451.
[5] S.W. Fuhrmann and Y.T. Wang, Analysis of cyclic systems with limited service: bounds and approximations, Perform. Eval. 9 (1988) 35–54.
[6] W.P. Groenendijk, private communication (1988).
[7] L. Kleinrock and H. Levy, The analysis of random polling systems, Oper. Res. 36 (1988) 716–732.
[8] J. Keilson and L.D. Servi, Oscillating random walk models for GI/G/1 vacation systems with Bernoulli schedules, J. Appl. Probab. 23 (1986) 790–802.
[9] H. Levy, Analysis of cyclic-polling systems with binomial-gated service, *Performance of Distributed and Parallel Systems*, T. Hasegawa, H. Takagi and Y. Takahashi (eds.) (Elsevier, 1989) to be published.
[10] H. Levy, Optimization of polling systems: the fractional-exhaustive service method, Technical report, Dept. of Computer Science, Tel-Aviv University (July 1988).
[11] L.D. Servi, Average delay approximation of M/G/1 cyclic service queues with Bernoulli schedules, IEEE J. Selected Areas in Commun. SAC-4 (1986) 813–822 (also correction in SAC-5 (April 1987) 547).
[12] L.D. Servi and D.D. Yao, Stochastic bounds for vacation models with limited service, *Performance of Distributed and Parallel Systems*, T. Hasegawa, H. Takagi and Y. Takahashi (eds.) (Elsevier, 1988) to be published, also in Perform. Eval. 9 (1988) 247–261.

[13] M. Sidi and H. Levy, A queueing network with a single cyclically roving server, Technical Report, Dept. of Computer Science, Tel-Aviv University (November 1988).
[14] H. Levy, M. Sidi and O.J. Boxma, Dominance relations in polling systems, Technical Report 117/88, Dept. of Computer Science, Tel-Aviv University (November 1988).
[15] H. Takagi, Mean message waiting time in a symmetric polling system, *Performance '84*, E. Gelenbe (ed.) (Elsevier Science North-Holland, 1985).
[16] H. Takagi, *Analysis of Polling Systems* (MIT Press, April 1986).
[17] Tedijanto, Stochastic comparisons in vacation models, *Int. Workshop on the Analysis of Polling Models*, Kyoto, Japan (December 1988).