Optimization of Polling Systems and Dynamic Vehicle Routing Problems on Networks

Dimitris Bertsimas *Haiping Xu*

OR 283-93 December 1993

Optimization of Polling Systems and Dynamic Vehicle Routing Problems on Networks

Dimitris Bertsimas [1]   Haiping Xu [2]

December 1993

[1]Dimitris Bertsimas, Sloan School of Management and Operations Research Center, MIT, Cambridge, MA 02139. The research of the author was partially supported by a Presidential Young Investigator Award DDM-9158118 with matching funds from Draper Laboratory.
[2]Haiping Xu, Center for Transportation Studies, MIT, Cambridge, MA 02139.

Abstract

We consider the problem of optimizing a polling system, i.e., of optimally sequencing a server in a multi-class queueing system with switch-over times in order to minimize a linear objective function of the waiting times. The problem has important applications in computer, communication, production and transportation networks. We propose nonlinear programming relaxations that provide strong lower bounds to the optimal cost for all static policies. We also obtain lower bounds for dynamic policies as well, which are primarily useful under light traffic conditions and/or small switch-over times. We conjecture that the lower bounds developed in this paper for the class of static policies are also valid for dynamic policies under heavy traffic conditions. We use the information from the lower bound and integer programming techniques to construct static policies that are very close (0-3%) to the lower bounds. We compare numerically our proposed policies with static policies proposed in the literature as well as with dynamic policies and find that the policies we propose outperform all static policies proposed in the literature and at least in heavier traffic outperform dynamic policies as well.

1 Introduction

Polling systems, in which a single server in a multi-class queueing system serves several classes of customers incurring switch-over times when he serves different classes, have important applications in computer, communication, production and transportation networks. In these application areas several users compete for access to a common resource (a central computer in a time sharing computer system, a transmission channel in a communication system, a machine in a manufacturing context or a vehicle in transportation applications). As a result, the problem has attracted the attention of researchers across very different disciplines. The name polling systems comes primarily from the communication literature. Motivated by its important applications, polling systems have a rather large literature, which almost exclusively addresses the performance of specific policies rather than the optimal design of the polling system. For an extensive discussion of the research work on polling systems, we refer to the survey papers by Levy and Sidi [11] and Takagi [15], [16].

Model description

Consider a system consisting of N infinite capacity stations (queues), and a single server which serves them one at a time. The arrival process to station i ($i = 1, 2, \ldots, N$) is assumed to be a Poisson process with rate λ_i. The overall arrival rate to the system is $\lambda = \sum_{i=1}^{N} \lambda_i$. Customers arriving to station i will be referred to as class-i customers and have a random service requirement X_i with finite mean x_i and second moment $x_i^{(2)}$ respectively. The actual service requirement of a specific customer is assumed to be independent of other system variables. The cost of waiting for class-i customers is c_i per unit time. There are switch-over time d_{ij} whenever the server changes from serving class-i customers to class-j customers. The offered traffic load at station i is equal to $\rho_i = \lambda_i x_i$, and the total traffic load is equal to $\rho = \sum_{i=1}^{N} \rho_i$. It is well known (see for example Takagi [15]) that the system is stable if and only if $\rho < 1$. Note that this condition does not depend on the switch-over times.

The natural performance measure in polling systems is the mean delay time between the request for service from a customer and the delivery of the service by the server to that customer. The scheduling problem in polling systems is to decide which customer should be in service at any given time in order to minimize the weighted expected delay of all the classes. Let $\mathcal{U}$ be the class of non-preemptive, non-anticipative and stable policies. Within $\mathcal{U}$ we further distinguish between static ($\mathcal{U}_{static}$) and dynamic ($\mathcal{U}_{dynamic}$) policies.

Static policies at each decision epoch do not take into account information about the state of stations in the system other than the one occupied by the server and are determined a priori or randomly. For example the policy under which the server visits the stations in a predetermined order according to a routing table is a static policy. Dynamic policies take into account information about the current state of the network. For example, a threshold policy or a policy that visits the most loaded station, is a dynamic policy, because the decision on which customer to serve next by the server depends on the current queue lengths at various stations in the system. In certain applications it might be impractical or even impossible to use a dynamic policy. For example in a transportation network, the vehicle might not know the overall state of the network. As a result, although static policies are not optimal, they can often be the only realistic policy. Moreover, when there are no switch-over times, the policy that minimizes the mean weighted delay is a strict priority rule (the $c\mu$ rule), a static policy.

In a transportation system the problem can be seen as a vehicle routing problem operating in a dynamic and stochastic environment. Customers arrive at nodes in a network according to Poisson processes of rate λ_i. A vehicle servicing these requests travels between the nodes in the network. These travel distances are modeled here as switch-over times. The goal is to find a sequencing policy that minimizes the weighted expected delay (see for example Psaraftis [13]). In this context, our work in this paper extends (although using drastically different techniques) the work of Bertsimas and van Ryzin [2], [3] for the dynamic vehicle routing problem in the Euclidean plane.

Despite extensive research efforts on the analysis of polling systems, results on the optimization of polling systems are scarce. Perkins and Kumar [12] develop a lower bound on the optimal objective value for polling systems in which there is no underlying randomness. Browne and Yechiali [7], using Markov decision processes, determine a semi-dynamic policy in which the server, at the beginning of a cycle, chooses a visiting order of the stations for this cycle that minimizes the mean duration of the cycle. Boxma, Levy and Weststrate [6] develop a heuristic approach to obtain a static policy under which the server visits the stations in a predetermined order (routing table). Hofri and Ross [8] find the structure of the optimal dynamic policy (which is of the threshold type) for a two station polling system but do not propose an analytic method to compute the parameters of the optimal policy. Reiman and Wein [14] approximate polling systems in heavy traffic using Brownian motion and construct dynamic policies. To the best of our knowledge, there are no results in the literature that propose a policy and a guarantee for the performance of this policy for polling systems.

Our approach to the problem is to use mathematical programming methodology to obtain lower bounds for the performance of an optimal policy and then construct policies that are provably close to the lower bounds. In particular, our contribution in this paper is as follows.

1. We propose a *nonlinear (but convex) optimization problem*, whose solution provides a lower bound on an arbitrary static policy. In certain cases we provide a lower bound in closed form as well. In this way we are able to assess the suboptimality of proposed static policies. We further conjecture that the bound we obtain is also valid for all dynamic policies, when the system is in heavy traffic. We provide numerical results that support this conjecture.

2. We propose a simple lower bound for all (dynamic and static) policies. The bound is particularly useful under light traffic and/or small switch-over times.

3. Using information from the lower bounds, we construct, using *integer programming*, static policies (routing table policies) that are very close (within 0-3%) to the lower bound. We also show that in special cases our bounds are tight.

4. We investigate numerically the effectiveness of our lower bounds and policies as a function of the switch-over times and the traffic intensity and find that the routing table policies constructed are adequate for practical problems if we optimize over static policies. For dynamic policies, we find that in light traffic the lower bound we obtained is informative, while in heavy traffic the numerical results suggest that the static lower bounds are still valid and therefore, we do not gain a lot in heavy traffic by optimizing over dynamic policies.

The paper is structured as follows: In Section 2, we develop the techniques used to obtain lower bounds on static policies as well as dynamic policies. We also obtain closed form expressions for the lower bounds by relaxing some constraints in the lower bound formulations and make some remarks on the tightness of the bounds. In section 3, we develop near optimal static policies based on integer programming. In Section 4, we compare numerically the upper and lower bounds with static and dynamic policies. Finally, in Section 5 we summarize our conclusions.

2 Lower bounds on achievable performance

We develop in this section lower bounds on the weighted mean waiting time for polling systems for non-preemptive, non-anticipative and stable policies. We call these policies admissible policies. We first focus on the class of static policies $\mathcal{U}_{static}$, in which the server's behavior when in station i is independent of the state of the other stations in the system (i.e., the queue lengths and the interarrival times of the customers). Examples of static policies include randomized policies, in which the next station to be visited by the server is determined by an a priori probability distribution, and routing table policies, in which the next station to be visited is predetermined by a routing table. A special case of the routing table policies is the cyclic policy, where the stations are visited by the server in a cyclic order.

Let $E[W_i]$ be the average waiting time of class-i customers. The goal is to find an admissible static policy $u \in \mathcal{U}_{static}$ to minimize the weighted mean delay $E[W]$ for the polling system:

$$\min_{u \in \mathcal{U}_{static}} E[W] = \frac{1}{\lambda} \sum_{i=1}^{N} c_i \lambda_i E[W_i]. \tag{1}$$

Throughout the paper we use the following notation:

d_{ij} = switch-over time when the server changes from serving class-i customers to class-j customers ($i, j = 1, 2, \ldots, N$). In general we assume that the $N \times N$ matrix $D = [d_{ij}]$ ($i \neq j$) of the switch-over times is asymmetric.

T_i^k = the time between the kth and the $(k+1)$th arrival of the server at station i.

$T_i = \lim_{k \to \infty} T_i^k$ and $E[T_i]$ and $E[T_i^2]$ are the first two moments of the time between two visits to station i by the server.

V_i^k = the time between the kth departure of the server from station i and the next return of the server to station i. Under a static policy, V_i^k is independent of the interarrival times of class-i customers, and, therefore, station i can be viewed as an $M/G/1$ system with server vacations. V_i^k is the kth server vacation observed at station i. Notice, however, that under an arbitrary static policy, successive vacations are not necessarily identically distributed and are not independent.

$v_i = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} V_i^k$, the mean server vacation time at station i.

$v_i^{(2)} = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} (V_i^k)^2$, the second moment of the server vacation time at station i.

$Y_{ij}(t)$ = the total number of visits by the server from station i to station j in the interval $[0, t)$.

$m_{ij} = \lim_{t\to\infty} \frac{Y_{ij}(t)}{t}$, the average number of visits per unit time by the server from station i to station j in steady state.

Q_i = the number of customers waiting in station i and $Q = \sum_{i=1}^{N} Q_i$.

$C_i(t)$ = the number of class-i customers serviced at station i by time t.

$F_i(t)$ = the number of finished server vacations at station i by time t.

$A_i(t)$ = the number of arrivals of class-i customers in $[0, t)$. Clearly, $E[A_i(t)] = \lambda_i t$.

$B_i(t)$ = the proportion of time that the server is busy serving class-i customers in $[0, t)$. For stability reasons, $\lim_{t\to\infty} \frac{B_i(t)}{t} = \rho_i$.

2.1 A lower bound for static policies

We consider the class of all static and admissible policies. Within a particular class, we assume that the server uses a First-In-First-Out (FIFO) discipline.

Proposition 1 *Under any static and admissible policy the expected waiting time of class-i customers decomposes as follows:*

$$E[W_i] = \frac{\lambda_i x_i^{(2)}}{2(1-\rho_i)} + \frac{v_i^{(2)}}{2v_i}, \quad i = 1, 2, \ldots, N. \tag{2}$$

Proof

Let W_i^l be the waiting time of the lth class-i customer in station i. We shall refer to this customer as the "tagged" customer. Upon the arrival of this tagged customer, we denote with R_i^l the remaining time until the service of the class-i customer in service is complete (if the server is busy with class-i customers) or the next return of the server to station i (if the server is on vacation). In the later case, the residual time until the next return of the server to station i can be calculated from V_i^k under the class of static policies $\mathcal{U}_{static}$, because of the independence between the arrival time of this tagged customer and the next return of the server to station i. Let X_i^k be the service requirement of the kth class-i customer, and Q_i^l be the number of class-i customers found waiting in station i by this tagged customer upon his arrival. Since the class of policies is non-preemptive and within class i the service

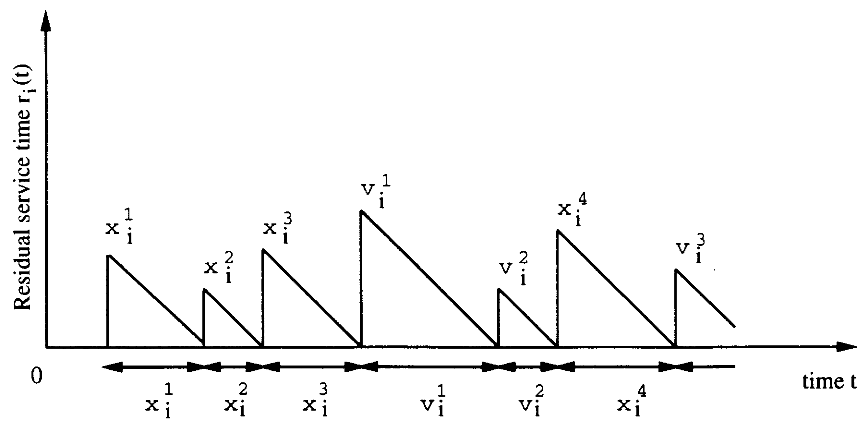

Figure 1: Residual service time for an M/G/1 system with vacations

discipline is FIFO, we obtain:

$$W_i^l = R_i^l + \sum_{k=l-Q_i^l}^{l-1} X_i^k.$$

By taking the expectations we obtain

$$E[W_i^l] = E[R_i^l] + x_i E[Q_i^l].$$

Since the class of policies is stable, the limit $\lim_{l\to\infty} E[W_i^l]$ exists. Because of PASTA, we obtain

$$E[W_i] = E[R_i] + x_i E[Q_i].$$

By Little's Law,

$$E[Q_i] = \lambda_i E[W_i],$$

and by substitution, we obtain

$$E[W_i] = \frac{E[R_i]}{1-\rho_i}. \tag{3}$$

We next calculate the expected residual time $E[R_i]$ using ideas from Bertsekas and Gallager [1]. In Figure 1 the residual service time $r_i(t)$, (i.e., the remaining time for the completion of the class-i customer in service or the remaining time for the server to return to station i to serve class-i customers at time t) is plotted as a function of t. Note that when a new service of duration X_i^k begins, $r_i(t)$ starts at X_i^k and decays linearly for X_i^k time units.

Similarly when a new vacation V_i^k begins, $r_i(t)$ starts at V_i^k and decays linearly for V_i^k time units. We use the assumption here that we only consider static and non-preemptive policies. Note that under a dynamic policy, the server's next return to station i may depend on the arrival time of this tagged customer and therefore the residual time can not be determined from the random variable V_i^k.

Consider a time τ for which $r_i(\tau) = 0$. Then

$$\frac{1}{\tau}\int_0^\tau r_i(t)dt = \frac{1}{\tau}\sum_{k=1}^{C_i(\tau)} \frac{1}{2}(X_i^k)^2 + \frac{1}{\tau}\sum_{k=1}^{F_i(\tau)} \frac{1}{2}(V_i^k)^2, \tag{4}$$

where $C_i(\tau)$ is the number of service completions of class-i customers within $[0,\tau]$, and $F_i(\tau)$ is the number of completed vacations by the server at station i within $[0,\tau]$.
But, in steady state,

$$E[R_i] = \lim_{\tau\to\infty}\frac{1}{\tau}\int_0^\tau r_i(t)dt.$$

Then,

$$E[R_i] = \lim_{\tau\to\infty}\frac{C_i(\tau)}{\tau}\lim_{C_i(\tau)\to\infty}\frac{\sum_{k=1}^{C_i(\tau)}\frac{1}{2}(X_i^k)^2}{C_i(\tau)} + \lim_{\tau\to\infty}\frac{F_i(\tau)}{\tau}\lim_{F_i(\tau)\to\infty}\frac{\sum_{k=1}^{F_i(\tau)}\frac{1}{2}(V_i^k)^2}{F_i(\tau)}.$$

In steady state,

$$\lim_{\tau\to\infty}\frac{C_i(\tau)}{\tau} = \lambda_i,$$

and

$$\lim_{C_i(\tau)\to\infty}\frac{\sum_{k=1}^{C_i(\tau)}\frac{1}{2}(X_i^k)^2}{C_i(\tau)} = \frac{x_i^{(2)}}{2}.$$

Moreover, in the interval $[0,\tau]$ the server will spend some time $B_i(\tau)$ serving class-i customers and time $\sum_{k=1}^{F_i(\tau)} V_i^k$ in vacations, where V_i^k is the duration of the kth vacation. Therefore,

$$B_i(\tau) + \sum_{k=1}^{F_i(\tau)} V_i^k = \tau.$$

Taking limits, and using the fact that in steady state $\lim_{\tau\to\infty}\frac{B_i(\tau)}{\tau} = \rho_i$, we obtain

$$\lim_{\tau\to\infty}\frac{F_i(\tau)}{\tau} = \frac{1-\rho_i}{v_i}. \tag{5}$$

Finally,

$$\lim_{F_i(\tau)\to\infty} \frac{\sum_{k=1}^{F_i(\tau)} \frac{1}{2}(V_i^k)^2}{F_i(\tau)} = \frac{v_i^{(2)}}{2}.$$

Substituting to (3), we obtain (2). □

Our goal is to obtain a lower bound on the optimal performance $E[W]$ in equation (1). Our main result is as follows:

Theorem 1 *The optimal weighted mean delay in a polling system under any static and admissible policy is bounded from below by:*

$$E[W] \geq \max\left\{\frac{1}{\lambda}\left[\sum_{i=1}^{N}\frac{\lambda_i x_i^{(2)}}{2}\right]\left[\sum_{i=1}^{N}\frac{c_i\lambda_i}{(1-\sigma_{i-1})(1-\sigma_i)}\right],\ z_{static}\right\}, \tag{6}$$

where $\sigma_i = \sum_{j=1}^{i}\rho_j$ and z_{static} is the solution of the following convex programming problem:

$$z_{static} = \min \frac{1}{2\lambda}\sum_{i=1}^{N}\frac{c_i\lambda_i^2 x_i^{(2)}}{1-\rho_i} + \frac{1}{2\lambda}\sum_{i=1}^{N}\frac{c_i\lambda_i(1-\rho_i)}{(\sum_{j=1}^{N} m_{ji})} \tag{7}$$

subject to

$$\sum_{j=1}^{N} m_{ij} - \sum_{k=1}^{N} m_{ki} = 0, \qquad i = 1,2,\ldots,N$$

$$\sum_{i,j=1}^{N} d_{ij}m_{ij} \leq 1-\rho$$

$$m_{ij} \geq 0 \qquad \forall\ i,j,$$

$$m_{ii} = 0 \qquad \forall\ i,$$

where m_{ij} $(i,j = 1,2,\ldots,N)$ have the interpretation of the average number of visits per unit time by the server from station i to station j.

Proof

Let $E[W^*(D)]$ be the optimal weighted mean delay under the optimal static policy $u^* \in \mathcal{U}_{static}$ for a polling system with switch-over matrix D. The optimal weighted mean delay is clearly a monotonically increasing function of D, i.e.,

$$E[W^*(D)] \geq E[W^*(0)], \quad \forall\ D \geq 0. \tag{8}$$

But for $D = 0$, the polling system becomes a traditional $M/G/1$ priority queueing system, in which the optimal policy that minimizes the weighted mean delay is the head-of-the-line (HOL) priority policy which gives the highest non-preemptive priority to the station with the largest c_i/x_i. Without loss of generality, we assume that queues are ordered such that $c_1/x_1 \geq c_2/x_2 \geq \ldots \geq c_N/x_N$. Under this priority scheme, the weighted mean delay for class-i customers is equal to (Kleinrock [9])

$$E[W_i] = \frac{\sum_{i=1}^{N} \lambda_i x_i^{(2)}}{2(1-\sigma_{i-1})(1-\sigma_i)}, \qquad i = 1, 2, \ldots, N, \tag{9}$$

where $\sigma_i = \sum_{j=1}^{i} \rho_j$ and $\sum_{i=1}^{N}(\lambda_i x_i^{(2)}/2)$ is the residual service time observed by a random arriving customer to the system. Therefore from equation (1) and (9) above, we obtain

$$E[W^*(0)] = \frac{1}{\lambda}\left[\sum_{i=1}^{N} \frac{\lambda_i x_i^{(2)}}{2}\right]\left[\sum_{i=1}^{N} \frac{c_i \lambda_i}{(1-\sigma_{i-1})(1-\sigma_i)}\right]. \tag{10}$$

Combining equations (8) and (10) we obtain the first part of the lower bound in (6). This part of the lower bound ignores the effects of switch-over times, but it is necessary in the case where all d_{ij}'s are very small; in this case and the polling system can be approximated well by a multi-class M/G/1 queue.

We now turn our attention to the second part of the lower bound. ¿From Proposition 1 the expected waiting time under a static admissible policy for a class-i customer decomposes

$$E[W_i] = \frac{\lambda_i x_i^{(2)}}{2(1-\rho_i)} + \frac{v_i^{(2)}}{2v_i}, \quad i = 1, 2, \ldots, N,$$

where v_i, $v_i^{(2)}$ are the first and second moments of the vacation times observed at station i respectively. Since $v_i^{(2)} \geq v_i^2$, we obtain from (1) and (2):

$$E[W] \geq \frac{1}{\lambda}\sum_{i=1}^{N} c_i \lambda_i \left[\frac{\lambda_i x_i^{(2)}}{2(1-\rho_i)} + \frac{v_i}{2}\right]. \tag{11}$$

¿From (5)

$$\lim_{t\to\infty} \frac{F_i(t)}{t} = \frac{1-\rho_i}{v_i}.$$

But since $F_i(t) = \sum_{j=1}^{N} S_{ji}(t)$, we obtain that

$$\lim_{t\to\infty} \frac{F_i(t)}{t} = \sum_{j=1}^{N} m_{ji},$$

and hence

$$v_i = \frac{1-\rho_i}{\sum_{j=1}^{N} m_{ji}}. \tag{12}$$

¿From (11) and (12) we obtain the objective function in equation (7). Since m_{ij} is the expected number of visits per unit time by the server from station i to j, flow conservation at station i requires that

$$\sum_{j=1}^{N} m_{ij} - \sum_{k=1}^{N} m_{ki} = 0, \qquad i = 1, 2, \ldots, N. \tag{13}$$

If $\overline{d}_i$ is the average switch-over time spend by the server per class-i customer, then the stability condition requires that

$$\sum_{i=1}^{N} \lambda_i (x_i + \overline{d}_i) \leq 1.$$

But

$$\overline{d}_i = \lim_{t\to\infty} \frac{\sum_{j=1}^{N} d_{ji} S_{ji}(t)}{A_i(t)} = \lim_{t\to\infty} \frac{\sum_{j=1}^{N} d_{ji} S_{ji}(t)}{t} \lim_{t\to\infty} (\frac{A_i(t)}{t})^{-1} = \frac{1}{\lambda_i} \sum_{j=1}^{N} d_{ji} m_{ji}.$$

Substituting to the stability condition we obtain

$$\sum_{i=1}^{N} (\rho_i + \sum_{j=1}^{N} d_{ji} m_{ji}) \leq 1,$$

or

$$\sum_{i,j=1}^{N} d_{ij} m_{ij} \leq 1 - \rho. \tag{14}$$

Since m_{ij} can not be negative, we obtain the convex programming formulation in (7) by combining equations (11), (12), (13) and (14). □

Remark:

The key assumption we used in the derivation of the lower bounds for static policies is the independence between the interarrival time of a tagged customer and the server's behavior. Under a dynamic policy, the remaining time until the server returns to station i in general can not be calculated from V_i^k, but the stability condition must still be satisfied

$$\sum_{i,j=1}^{N} d_{ij} m_{ij} \leq 1 - \rho.$$

In heavy traffic, $\rho \to 1$ and thus (assuming $d_{ij} > 0 \ \forall\ i,j$), we see that $m_{ij} \to 0, \ \forall\ i,j$, which implies that the average time duration between two consecutive visits to any station in the polling system should be very large in order for the system to be stable. This implies that for the system to be stable in heavy traffic the server's behavior should not depend significantly on the interarrival time of the tagged customer. In other words, we conjecture that the static lower bounds developed in the previous subsections hold even for dynamic policies under heavy traffic conditions.

2.2 Lower bounds for homogeneous polling systems under exhaustive static policies

We call a polling system *homogeneous* if the costs and service requirements are the same among all different classes of customers (i.e., $c_i = c = 1, \ x_i = x$, for all $i = 1, 2, \ldots, N$). Our goal in this subsection is to improve the lower bound for the special case of homogeneous polling systems under *exhaustive* and static admissible policies, in which the server leaving station i does not leave any class-i customers waiting.

Theorem 2 *If $c_i = c = 1$ and $x_i = x$ for all $i = 1, 2, \ldots, N$, the optimal mean delay $E[W]$ in a homogeneous polling system under exhaustive, static admissible policies is bounded from below by the optimal solution of the following convex program:*

$$E[W] \geq \ z_{static}^{hom} \ = \min \ \frac{\lambda x^{(2)}}{2(1-\rho)} + \frac{1}{2\lambda} \sum_{i=1}^{N} \frac{\lambda_i (1-\rho_i)}{\sum_{j=1}^{N} m_{ji}} \tag{15}$$

subject to

$$\sum_{j=1}^{N} m_{ij} - \sum_{k=1}^{N} m_{ki} = 0, \qquad i = 1, 2, \ldots, N$$

$$\sum_{i,j=1}^{N} d_{ij} m_{ij} \leq 1 - \rho$$

$$m_{ij} \geq 0, \; m_{ii} = 0 \qquad \forall \; i, j,$$

where m_{ij} $(i, j = 1, 2, \ldots, N)$ have the interpretation of the average number of visits per unit time by the server from station i to station j.

Proof

For the homogeneous polling system where the system costs and service requirements are equal for all customer classes, (i.e., $c_i = c = 1$; $x_i = x$; $\forall \; i = 1, 2, \ldots, N$), the mean delay of a random customer in steady state can be written as

$$E[W] = E[R] + xE[Q] + E[S], \tag{16}$$

where $E[R]$ is the mean residual service time of the customer in service upon this random customer's arrival, $E[Q]$ is the average number of customers waiting in the system, and $E[S]$ is the total switch-over times spend by the server during which this random customer must wait before being served.

The residual service time is $E[R] = \lambda x^{(2)}/2$ and by Little's Law, $xE[Q] = \rho E[W]$. We then obtain that

$$E[W] = \frac{\lambda x^{(2)}}{2(1-\rho)} + \frac{E[S]}{1-\rho}. \tag{17}$$

Conditioning on the event U_i that the incoming customer is class-i, we obtain

$$E[S] = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} E[S|U_i]. \tag{18}$$

Conditioning further on the event O_i that the server is at station i, we obtain:

$$E[S|U_i] = \rho_i E[S|U_i, O_i] + (1 - \rho_i) E[S|U_i, O_i^c],$$

where U_i^c is the event that the server is not in station i. Under an exhaustive policy, $E[S|U_i, O_i] = 0$, and thus

$$E[S|U_i] = (1 - \rho_i) E[S|U_i, O_i^c].$$

Equation (18) can then be written as

$$E[S] = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} (1 - \rho_i) E[S|U_i, O_i^c]. \tag{19}$$

Since the server is busy ρ proportion of the time, the server spends $(1-\rho)$ proportion of the time on switch-overs. Note that T_i was defined to be the time between two consecutive arrivals of the server at station i. Under a static admissible policy, given that the server is not at station i, the expected time spent in switch-overs by the server until this incoming class-i customer is served is

$$E[S|U_i, O_i^c] = (1-\rho)\, E[T_i^*] = (1-\rho)\frac{E[T_i^2]}{2E[T_i]} \geq (1-\rho)\frac{E[T_i]}{2}, \tag{20}$$

where T_i^* is the forward recurrence time of T_i. From equations (17), (19) and (20) we obtain:

$$E[W] \geq \frac{\lambda x^{(2)}}{2(1-\rho)} + \sum_{i=1}^{N} \frac{\lambda_i (1-\rho_i) E[T_i]}{2\lambda} \tag{21}$$

Notice that $\sum_{k=1}^{F_i(t)} T_i^k - t = o(t)$, for all stable policies. Taking limits we obtain, that

$$E[T_i] = \frac{1}{\sum_{j=1}^{N} m_{ji}},$$

which combined with equation (21) leads to

$$E[W] \geq \frac{\lambda x^{(2)}}{2(1-\rho)} + \sum_{i=1}^{N} \frac{\lambda_i (1-\rho_i)}{2\lambda \sum_{j=1}^{N} m_{ji}}. \tag{22}$$

Equation (22), together with (13) and (14) gives the lower bound in (15). □

Remark: Notice that the bound of Theorem 2 is stronger than the one in Theorem 1.

2.3 Closed form bounds on static policies

In order to acquire further insight on the bounds of Theorems 1 and 2, the flow conservation constraints (13) are relaxed to obtain a closed form formula for the lower bounds.

Theorem 3

a) For a polling system, the weighted mean delay for all static and admissible policies is bounded from below by:

$$E[W] \;\geq z_{closed} = \max\left\{ \frac{1}{\lambda} \left[\sum_{i=1}^{N} \frac{\lambda_i x_i^{(2)}}{2} \right] \left[\sum_{i=1}^{N} \frac{c_i \lambda_i}{(1-\sigma_{i-1})(1-\sigma_i)} \right] \right.,$$

$$\frac{1}{2\lambda}\sum_{i=1}^{N}\frac{c_i\lambda_i^2 x_i^{(2)}}{1-\rho_i}+\frac{(\sum_{i=1}^{N}\sqrt{c_i\lambda_i(1-\rho_i)d_i^*})^2}{2\lambda(1-\rho)}\Bigg\}, \tag{23}$$

where $\sigma_i = \sum_{j=1}^{i}\rho_j$ *and* $d_i^* = d_{j(i),i} = \min_j\{d_{ji}\}$. *b) For a homogeneous polling system, under exhaustive, static and admissible policies, the weighted mean delay is bounded from below by:*

$$E[W] \geq z_{closed}^{hom} = \frac{\lambda x^2}{2(1-\rho)}+\frac{(\sum_{i=1}^{N}\sqrt{\lambda_i(1-\rho_i)d_i^*})^2}{2\lambda(1-\rho)}. \tag{24}$$

Proof

We relax the flow conservation constraints (13) in the nonlinear formulation (7). Note that the first term in the objective function is a constant. Therefore we need only to solve the following convex program:

$$\min\ \frac{1}{2\lambda}\sum_{i=1}^{N}\frac{c_i\lambda_i(1-\rho_i)}{(\sum_{k=1}^{N}m_{ki})} \tag{25}$$

subject to

$$\sum_{i,j=1}^{N} d_{ij}m_{ij} \leq 1-\rho$$

$$m_{ij} \geq 0 \qquad \forall\ i,j.$$

Since the above program is a convex programming program, the Kuhn-Tucker conditions are necessary and sufficient. The Kuhn-Tucker conditions are:

$$\begin{cases} -c_i\lambda_i(1-\rho_i)/(2\lambda(\sum_{k=1}^{N}m_{ki})^2)+\alpha d_{ji}-\beta_{ji}=0, & \forall\ j,i \\ \alpha \geq 0 & \\ \alpha[\sum_{i,j=1}^{N}d_{ij}m_{ij}-(1-\rho)]=0 & \\ \sum_{i,j=1}^{N}d_{ij}m_{ij}\leq 1-\rho & \\ \beta_{ij}\geq 0 & \forall\ i,j \\ -\beta_{ij}m_{ij}=0 & \forall\ i,j \\ m_{ij}\geq 0 & \forall\ i,j, \end{cases} \tag{26}$$

where α and β_{ij} are dual variables. Let $d_i^* = d_{j(i),i} = \min_j\{d_{ji}\}$. The solution of (26) is

$$\begin{cases} \alpha = (\sum_{k=1}^{N} \sqrt{c_k\lambda_k(1-\rho_k)d_k^*})^2/(2\lambda(1-\rho)^2) \\ \beta_j = \alpha(d_{ji} - d_i^*), \qquad \forall\, j, i \;_i \\ m_{ji} = 0, \qquad \forall\, j, i;\; j \neq j(i) \\ m_{j(i),i} = (1-\rho)\sqrt{c_i\lambda_i(1-\rho_i)/d_i^*}/(\sum_{k=1}^{N} \sqrt{c_k\lambda_k(1-\rho_k)d_k^*}) \end{cases} \tag{27}$$

Substituting m_{ij} given in equation (27) into the objective function in equation (7) we obtain (23) and into the objective function in equation (15) we obtain (24). □

The bounds of Theorem 3 are not as sharp as the bounds of Theorems 1 and 2, since we have relaxed the flow conservation constraints.

If the switch-over times are identical, i.e., we consider the case with $d_{ij} = d,\ \forall\, i, j$. Then equation (23) becomes,

$$\begin{aligned} E[W] \;\geq z_{closed} = \;& \max\left\{ \frac{1}{\lambda}\left[\sum_{i=1}^{N} \frac{\lambda_i x_i^{(2)}}{2}\right]\left[\sum_{i=1}^{N} \frac{c_i\lambda_i}{(1-\sigma_{i-1})(1-\sigma_i)}\right], \right. \\ & \left. \frac{1}{2\lambda}\sum_{i=1}^{N} \frac{c_i\lambda_i^2 x_i^{(2)}}{1-\rho_i} + \frac{(\sum_{i=1}^{N}\sqrt{c_i\lambda_i(1-\rho_i)})^2}{2\lambda(1-\rho)} d \right\} \end{aligned} \tag{28}$$

while, for the homogeneous polling system, equation (24) becomes

$$E[W] \geq z_{closed}^{hom} = \frac{\lambda x^2}{2(1-\rho)} + \frac{(\sum_{i=1}^{N}\sqrt{\lambda_i(1-\rho_i)})^2}{2\lambda(1-\rho)} d. \tag{29}$$

The lower bounds z_{static} in equations (28) and (29) are plotted as a function of switch-over time d in Figure 2a and Figure 2b respectively.

2.4 Lower bounds for homogeneous polling systems for all policies

For a homogeneous polling system, we can divide the total delay of a tagged customer k, W^k, into two parts: the waiting time due to server's switch-over time prior to serving the tagged customer, denoted by $W^k(d)$; and the waiting time due to the on-site service times of customers served prior to the tagged customer, denoted by $W^k(s)$. Thus

$$E[W^k] = E[W^k(d)] + E[W^k(s)].$$

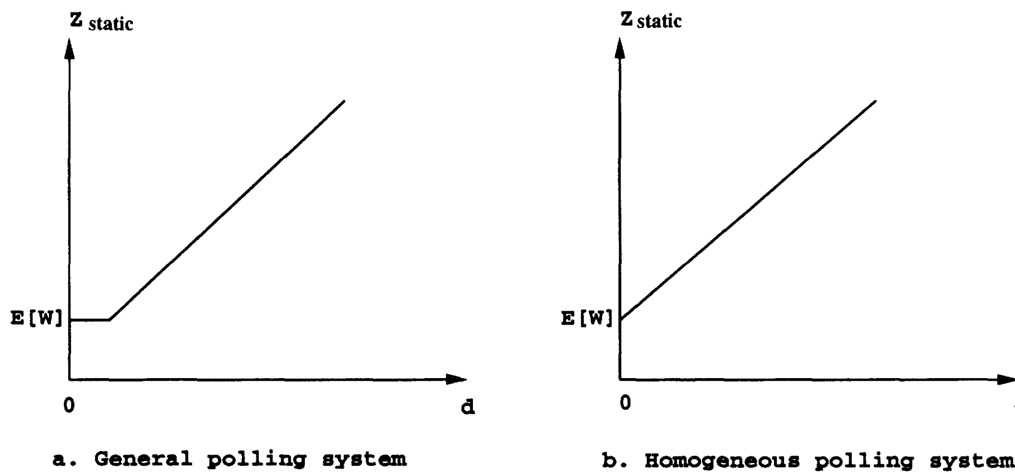

Figure 2: Lower bounds as a function of switch-over time d

Taking expectations and letting $k \to \infty$ gives

$$E[W] = E[W(d)] + E[W(s)],$$

where $E[W(d)] = \lim_{k\to\infty} E[W^k(d)]$ and $E[W(s)] = \lim_{k\to\infty} E[W^k(s)]$. To bound $E[W(d)]$, note that it is at least as large as the switch-over time between the server's location at the time of the tagged customer's arrival and the tagged customer's location. Since the policies are non-anticipative, the server is located at different stations according to some (generally unknown) distribution that depends on the server's policy. Suppose that we have the option of locating the server in the best a priori location that minimize the expected distance to a random arrival. This certainly yields a lower bound on the expected distance between the server and a random arrival. So,

$$E[W(d)] \geq \min_{j\in N}\{\sum_{i=1}^{N} \frac{\lambda_i}{\lambda} d_{ji}\}. \tag{30}$$

To bound $E[W(s)]$, let Q denote the expected number of customers served during a waiting time, since service times are independent, we then have

$$E[W(s)] \geq xE[Q] + \frac{\rho x^{(2)}}{2x},$$

where the second term is the expected residual service time of the customer being served upon the tagged customer's arrival times the probability that the server is busy. Then,

$$E[W(s)] \geq x\lambda E[W] + \frac{\lambda x^{(2)}}{2} = \rho E[W] + \frac{\lambda x^{(2)}}{2}.$$

Since $E[W] = E[W(d)] + E[W(s)]$ we obtain

$$E[W(s)] \geq \frac{\rho}{1-\rho} E[W(d)] + \frac{\lambda x^{(2)}}{2(1-\rho)}. \tag{31}$$

¿From equations (30) and (31) and noticing that the bound is valid for all policies we obtain the following theorem:

Theorem 4 *The mean delay in a homogeneous polling system under any admissible policy is bounded from below by*

$$E[W] \geq z_{dynamic} = \frac{1}{1-\rho} \min_{j \in N} \{\sum_{i=1}^{N} \frac{\lambda_i}{\lambda} d_{ji}\} + \frac{\lambda x^{(2)}}{2(1-\rho)}. \tag{32}$$

The bound $z_{dynamic}$ is primarily useful under light traffic and/or small switch-over times. As already mentioned we conjecture that in heavy traffic the bounds of Theorem 1 remain valid for all policies.

3 Design of effective static policies

In the previous section we determined lower bounds for the optimal static policies. In this section our goal is to use the information contained in the lower bounds of Theorems 1, 2 and 3 to construct near optimal static policies.

In the derivation of the lower bound (7) and (15) we have calculated values of m_{ij} that are interpreted as the steady state average number of visits from station i to station j per unit time. Let $e_{ij} = m_{ij}/\sum_{k,l} m_{kl}$ be the ratio of switch-overs from station i to station j over all switch-overs in the system. $E = [e_{ij}]$ is the corresponding switch-over ratio matrix. Intuitively, in order for the performance of a policy u to be close to the lower bound, it is desirable that the proportion of switch-overs from station i to station j under the policy u is close to e_{ij}. We refer to this requirement as the *closeness* condition. We consider two classes of policies that satisfy the closeness condition approximately.

Randomized policies:

Under this class of policies the server after serving exhaustively all customers at station i moves to station j with probability p_{ij}. Kleinrock and Levy [10] consider randomized policies, in which the next station visited will be station j with probability p_j, independent of the present station. Boxma and Weststrate [5] derive a pseudoconservation law for this class of policies.

Given the values of m_{ij} from the lower bound calculation, we would like to choose the probabilities p_{ij} so that the closeness condition is satisfied. An obvious choice is to pick $p_{ij} = e_{ij}/\sum_{k=1}^{N} e_{ik}$. $P = [p_{ij}]$ is the corresponding switch-over probability matrix. We note, however, that this choice of p_{ij} does not necessarily represent the optimal randomized policy.

Routing table policies

Under this class of policies the server visits stations in an a priori periodic sequence. For example the server visits a three station system using the cyclic sequence (1,2,3,1,2,3,...) or the sequence (1,2,1,2,3,1,2,1,2,3,...), i.e., stations 1 and 2 are visited twice as often as station 3. Boxma et. al. [6] use heuristic rules to construct routing table policies.

We use integer programming methods to construct routing tables that satisfy the closeness condition. Let h_{ij} be the number of switch-overs from station i to station j in an optimal routing table. $H = [h_{ij}]$ is the switch-over matrix. Note that unlike m_{ij}, h_{ij} should be integers. Notice that $\sum_{i,j} h_{ij}$ is the length of the routing table, i.e., the total number of switch-overs in the periodic sequence. Moreover, $e_{ij}\sum_{k,l} h_{kl}$ is the desired number of switch-overs from station i to station j in the routing table under the closeness condition. In order to satisfy the closeness condition, a possible objective in selecting a routing table is to minimize the maximum difference between the number of switch-overs h_{ij} from station i to station j in the optimal routing table and the desired number of switch-overs determined by $e_{ij}\sum_{k,l} h_{kl}$. i.e.,

$$\min_h\{\max_{i,j}\{|h_{ij} - e_{ij}\sum_{k,l} h_{kl}|\}\}. \tag{33}$$

In addition, the flow conservation at each station requires that

$$\sum_{j=1}^{N} h_{ij} - \sum_{k=1}^{N} h_{ki} = 0, \qquad i = 1, 2, \ldots, N, \tag{34}$$

i.e., the number of visits by the server to station i should equal to the number of visits by the server from station i to other stations. The h_{ij}'s should also form an Eulerian tour. Let I be the set of all stations and G be the subset of stations in the network. Since the Eulerian tour should be connected, we require that for all subsets G of the stations

$$\sum_{i \in G,\ j \in \bar{G}} h_{ij} \geq 1, \ \ \forall\ G \subset I,\ G \neq \phi. \tag{35}$$

In summary, the problem becomes

$$\begin{array}{lll} (P_{Eulerian}) & \min_h \{\max_{i,j}\{|h_{ij} - e_{ij} \sum_{k,l} h_{kl}|\}\} & \\ \text{subject to:} & \sum_{j=1}^{N} h_{ij} - \sum_{k=1}^{N} h_{ki} = 0, & i = 1, 2, \ldots, N \\ & \sum_{i \in G,\ j \in \bar{G}} h_{ij} \geq 1, & \forall\ G \subset I,\ G \neq \phi \\ & h_{ij} \geq 0,\ integer, & i, j = 1, 2, \ldots, N \end{array} \tag{36}$$

Equation (36) can be easily converted to a pure integer programming problem. Since our goal is only to obtain an approximate solution, we approximate the problem by relaxing the connectivity constraints in equation (35). But if equation (35) is relaxed, $h_{ij} = 0,\ \forall\ i, j$ will be a feasible solution and will minimize the objective function in (36). In order to exclude this infeasible solution to (36), we impose a lower limit on the length of the routing table. Since each of the stations should be visited at least once in any feasible routing table, the length of any routing table should be at least N. Moreover, we place an upper bound L_{max} on the length of the routing table to make the integer programming solvable:

$$\begin{array}{lll} (P_{approx}) & \min_h \{\max_{i,j}\{|h_{ij} - e_{ij} \sum_{k,l} h_{kl}|\}\} & \\ \text{subject to} & \sum_{j=1}^{N} h_{ij} - \sum_{k=1}^{N} h_{ki} = 0, & i = 1, 2, \ldots, N \\ & N \leq \sum_{i,j} h_{ij} \leq L_{max} & \\ & h_{ij} \geq 0,\ integer, & i, j = 1, 2, \ldots, N \end{array}$$

The previous formulation can be reformulated as a pure integer programming problem as follows:

$$
\begin{array}{lll}
(P_{approx}) & \min\ y & \\
\text{subject to} & y - h_{ij} + e_{ij} \sum_{k,l} h_{kl} \geq 0, & i,j = 1,2,\ldots,N \\
 & z + h_{ij} - e_{ij} \sum_{k,l} h_{kl} \geq 0, & i,j = 1,2,\ldots,N \\
 & \sum_{j=1}^{N} h_{ij} - \sum_{k=1}^{N} h_{ki} = 0, & i = 1,2,\ldots,N \\
 & N \leq \sum_{i,j} h_{ij} \leq L_{max} & \\
 & h_{ij} \geq 0,\ integer, & i,j = 1,2,\ldots,N
\end{array}
\tag{37}
$$

Note that there are many feasible routing tables that will be consistent with the h_{ij}'s obtained from the solution of (P_{approx}). We will select a Eulerian tour that spaces the visits to the stations as equally as possible. Although it is possible to formulate this requirement precisely as another integer programming problem, we found numerically that the added effort is not justified from the results it produces.

4 On the performance of static and dynamic policies

Using the lower bounds developed in Section 2, we are able to assess the performance of the policies constructed in Section 3. The main questions we address are:

1. Are the lower bounds we obtain tight or are they further improvable?
2. What is the degree of suboptimality of randomized policies ?
3. What is the degree of suboptimality of routing table policies ?
4. How the routing table policies proposed in Section 3 compare with those proposed in Boxma et. al. [6] ?
5. How much is the performance improvement if dynamic policies are used ?

In order to obtain some preliminary insights about the tightness of our lower bounds we consider first the case of a completely symmetric polling system.

4.1 Tightness of the lower bounds in symmetric and homogeneous polling systems

We consider a completely symmetric and homogeneous polling system, i.e., $\lambda_i = \lambda/N$, $c_i = c = 1$ and $x_i = x$ for all $i = 1,\ldots,N$; $d_{ij} = d$ for all $i,j = 1,\ldots,N$. The service discipline at all queues are exhaustive.

Cyclic policies

The simplest and the most commonly studied static policy is the cyclic polling policy. In such cyclic polling policies the server visits the stations in a cyclic order (namely, $1, 2, \ldots, N$, $1, 2, \ldots, N, \ldots$). Using the lower bounds developed in Section 2, we are able to show that the cyclic policy is in fact optimal among all static policies for a completely symmetric and homogeneous polling system.

Theorem 5 *The exhaustive cyclic policy is optimal among all static admissible policies for a completely symmetric and homogeneous polling system.*

Proof

Under the exhaustive cyclic policy, the average waiting time of the system is given in [1],

$$E[W_{cyclic}] = \frac{\lambda x^{(2)}}{2(1-\rho)} + \frac{(N-\rho)d}{2(1-\rho)},$$

which equals exactly the lower bound z_{static}^{hom} given in equation (29). Therefore,

$$\frac{E[W_{cyclic}]}{z_{static}^{hom}} = 1. \qquad \square$$

Although the previous result is intuitively obvious it shows that the lower bound given in (29) is indeed tight in this special case.

Randomized policies

We next show that the performance of a randomized policy in the symmetric and homogeneous polling system is not very attractive.

Theorem 6 *For a completely symmetric and homogeneous polling system, the average waiting time under a randomized polling policy with routing probabilities $p_i = 1/N$ is within a factor of 2 from the optimal static solution.*

Proof

Using the results from the pseudoconservation law of Boxma and Weststrate [5], we obtain that the average waiting time with routing probabilities $p_i = 1/N$ is

$$E[W_{random}] = \frac{\sum_{i=1}^{N} \lambda_i x^{(2)}}{2(1-\rho)} + \frac{(N-\rho)d}{(1-\rho)} - d = \frac{\lambda x^{(2)}}{2(1-\rho)} + \frac{(N-1)d}{(1-\rho)}$$

and from equation (29), the optimal mean waiting time is bounded from below by

$$z_{static}^{hom} = \frac{\lambda x^{(2)}}{2(1-\rho)} + \frac{(N-\rho)d}{2(1-\rho)},$$

which immediately shows the theorem. Note we have shown that the lower bound can be achieved by a cyclic policy for this special case and that as N or d increase the ratio $E[W_{random}]/z_{static}^{hom}$ can become arbitrarily close to 2. □

4.2 Suboptimality of routing table and randomized policies

In this subsection we perform several simulation experiments in order to assess the performance of the routing table and randomized policies. Although pseudoconservation laws have been developed for these two classes of policies [4], [5], there are no simple expressions available for the mean waiting times. For this reason we used simulation to assess the performance of proposed policies. All simulation results reported are based on 10 replication runs and the simulation time is set to 1 million units to allow for steady state to be reached. The initial 50,000 units of time are discarded as transient period and the following 950,000 units of time are used to collect statistics for various policies. We use the exhaustive service discipline exclusively.

A three station example

Boxma et. al. [6] developed a heuristic approach to determine an optimal routing table for polling systems. The network used in their study is used here to facilitate comparisons. The polling system is shown in Figure 3. It consists of three stations. All arrival processes are Poisson with arrival rates $\lambda_1 = 0.54$, $\lambda_2 = 0.24$ and $\lambda_3 = 0.06$; all service time distributions are exponential with service rates equal to 1 unit at all three stations, i.e., the traffic intensity is $\rho = 0.84$. The switch-over times are equal to 1 among all stations. We set $d_{ii} = 5000$ for all stations to reflect the fact that self-transitions are not allowed. The best routing table obtained in their study is table1=121213.

As an example of how our methodology proceeds we give some details of our calculation for both the lower and upper bounds. The lower bound obtained numerically from (15)

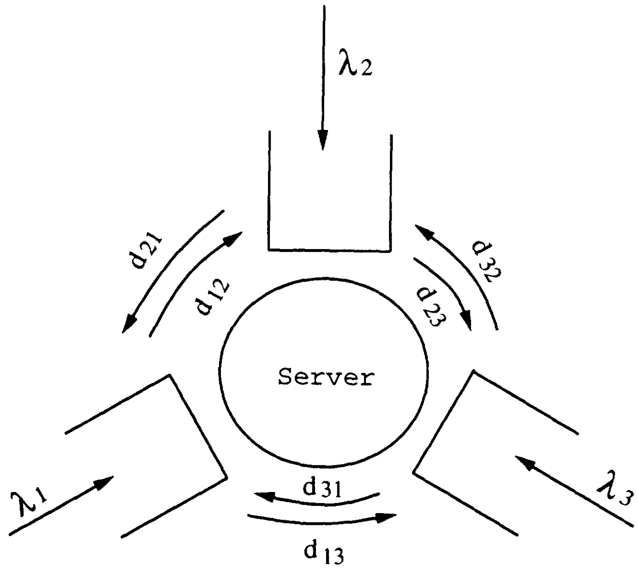

Figure 3: A 3 queue polling system

with the flow conservation constraint is 10.282. The matrix for the optimal m_{ij}'s is

$$M = \begin{bmatrix} 0.0 & 0.059 & 0.010 \\ 0.036 & 0.0 & 0.023 \\ 0.033 & 0.0 & 0.0 \end{bmatrix}$$

Using $e_{ij} = m_{ij} / \sum_{i,j}^{N} m_{ij}$, the normalized ratio matrix is

$$E = \begin{bmatrix} 0.0 & 0.367 & 0.061 \\ 0.224 & 0.0 & 0.143 \\ 0.204 & 0.0 & 0.0 \end{bmatrix}$$

We set the initial limit on the size of the routing table L_{max} to 1000. Solving (P_{approx}) in equation (37) we find that the optimal h_{ij}'s are

$$H = \begin{bmatrix} 0 & 18 & 3 \\ 11 & 0 & 7 \\ 10 & 0 & 0 \end{bmatrix}$$

with a table length of 42 and the maximum discrepancy of 0.0069.

The table size of 42 might be too large for a small network of three stations. Therefore, we would like to explore other routing tables with comparable values of discrepancy but smaller table sizes. This can be done easily by setting L_{max} to the previous value of the optimal table size minus 1. In this example, we change L_{max} to 41, and solved (37) again. This time the optimal matrix for h_{ij}'s becomes

$$H' = \begin{bmatrix} 0 & 5 & 1 \\ 3 & 0 & 2 \\ 3 & 0 & 0 \end{bmatrix} \tag{38}$$

with a table length of 14 and discrepancy 0.1415. A routing table consistent with this matrix H' is

$$table2 = 1 \to 2 \to 1 \to 2 \to 3 \to 1 \to 2 \to 1 \to 2 \to 3 \to 1 \to 2 \to 1 \to 3$$

We simulated both routing table $table2$ proposed above and the routing table $table1 =$ 121213 proposed in [6]. For switch-over time of $d = 1$, the lower bound is $z_{static} = 10.282$, while the simulation results $E[W_{table1}] = 10.642$ and $E[W_{table2}] = 10.505$. Therefore, for this example

$$\frac{W_{table1}}{z_{static}} = 1.035, \quad \frac{W_{table2}}{z_{static}} = 1.022,$$

which shows that routing table $table2$ is better and near optimal. We next examine the closeness of the lower bounds more systematically.

Effect of switch-over times

In Table 1, we present simulation results for routing table policies $table1$ and $table2$ along with dynamic, cyclic and randomized policies for the previous example polling system shown in Figure 3 as d varies from 0.01 to 1000.

Under the cyclic policy, the server visits stations in a cyclic order; Under the randomized policy, the routing probabilities are obtained from the H' matrix shown in equation (38), i.e.,

$$P = \begin{bmatrix} 0 & \frac{5}{6} & \frac{1}{6} \\ \frac{3}{5} & 0 & \frac{2}{5} \\ 1 & 0 & 0 \end{bmatrix};$$

Table 1: Performance as a function of the change-over time d

d	z_{static}	$\frac{W_{table1}}{z_{static}}$	$\frac{W_{table2}}{z_{static}}$	$\frac{W_{cyclic}}{z_{static}}$	$\frac{W_{random}}{z_{static}}$	$\frac{W_{dynamic}}{z_{static}}$	$z_{dynamic}$	$\frac{W_{dynamic}}{z_{dynamic}}$
0.01	5.300	1.004	1.004	1.004	1.011	0.998	5.272	1.003
0.10	5.753	1.009	1.006	1.011	1.032	0.976	5.473	1.026
0.50	7.766	1.023	1.016	1.029	1.071	0.951	6.366	1.160
1.00	10.282	1.035	1.022	1.042	1.097	0.955	7.482	1.312
3.00	20.345	1.049	1.031	1.060	1.136	1.001	11.946	1.705
5.00	30.408	1.054	1.033	1.067	1.148	1.006	16.411	1.864
10.00	55.566	1.058	1.036	1.072	1.156	1.026	27.571	2.068
50.00	256.830	1.062	1.038	1.078	1.172	1.045	116.857	2.297
100.00	508.410	1.063	1.039	1.079	1.171	1.053	228.464	2.343
500.00	2521.048	1.064	1.037	1.078	1.157	1.061	1121.321	2.385
1000.00	5036.847	1.061	1.035	1.077	1.140	1.061	2237.393	2.388
Average		1.044	1.026	1.054	1.117	1.012		

Under the dynamic policy, the server serves the stations exhaustively. When the current station becomes empty, the server moves to the most loaded queue, while if the system is empty, the server remains idle at the current station. Note that our static lower bounds do not hold for dynamic policies as expected for small values of switch-over times.

Both the static lower bounds z_{static} and the dynamic lower bounds $z_{dynamic}$ are listed in Table 1 for comparison. The simulation results on the performance of various policies are listed as a ratio over the lower bound. The last line in Table 1 reports the average suboptimality averaged over all values of d.

The following observations can be made:

1. The dynamic policy is within a few percentage of the lower bound on average, which is an interesting result, as it shows that although the static lower bound does not hold for dynamic policies in general, at least for a higher values of ρ (in the example $\rho = 0.84$), the dynamic policies do not improve performance significantly. It is interesting to observe that as d increases the policy we proposed based on the solution of (P_{approx}) performs better than all policies, including the dynamic policy.

2. Among static policies, the policy we propose based on the solution of (P_{approx}) is a clear winner especially as d increases.

3. The randomized policies are significantly worse than the routing table policies.

4. The simulation results suggest a linear dependence on d for all policies consistent with Figure 2.

5. At a lower switch-over time of $d = 0.01$, the light traffic lower bound is very tight, which should not be surprising because when $d \to 0$, the polling becomes a M/G/1 queueing system and we know the bound is exact. Even as d increases the lower bound $z_{dynamic}$ is somewhat informative.

Effect of asymmetry in switch-over times

When the switch-over times are different, the lower bounds given in equations (23) and (24) no longer provide the tightest possible bounds. We need to solve for the lower bound with the flow conservation constraints in equation (13). Using the same network shown in Figure 3 we keep all parameters unchanged except the switch-over times. The new asymmetric switch-over time matrix is changed to

$$D = \begin{bmatrix} 5000.0 & 1.4 & 1.3 \\ 1.1 & 5000.0 & 1.0 \\ 1.2 & 1.0 & 5000.0 \end{bmatrix}$$

where d_{ii} are set to a large number (5000) to reflect the fact that self-transitions are not allowed. For this new switch-over time matrix, $d^* = \{1.1, 1.0, 1.0\}$. The closed form lower bound from equation (24) is 10.494. The lower bound obtained numerically from equation (15) with flow conservation constraints is 11.185, which improves the previous bound by 6.58%. For comparison $z_{dynamic} = 8.33$. Solving the lower bound formulation in equation (15) for this problem we find that the optimal matrix for m_{ij}'s is

$$M = \begin{bmatrix} 0.0 & 0.021 & 0.032 \\ 0.053 & 0.0 & 0.0 \\ 0.0 & 0.032 & 0.0 \end{bmatrix}$$

Using $e_{ij} = m_{ij} / \sum_{i,j=1}^{N} m_{ij}$, the normalized ratio matrix E is

$$E = \begin{bmatrix} 0.0 & 0.152 & 0.232 \\ 0.384 & 0.0 & 0.0 \\ 0.0 & 0.232 & 0.0 \end{bmatrix}$$

Let the initial limit on the size of the routing table L_{max} be 1000 and solving problem (P_{approx}) in (37) we find that the optimal h_{ij}'s are:

$$H = \begin{bmatrix} 0 & 19 & 29 \\ 48 & 0 & 0 \\ 0 & 29 & 0 \end{bmatrix}$$

with a table length of 125 and discrepancy 2×10^{-16}. We change L_{max} to 125 - 1 = 124, and solve (37) again. This time the optimal matrix for h_{ij} becomes

$$H' = \begin{bmatrix} 0 & 2 & 3 \\ 5 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix}$$

with a table length of 13 and discrepancy 0.024. A possible routing table consistent with this H' matrix is:

$$\textit{routing table} = 1 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 2$$

Using this routing table, the simulation results show that the average waiting time is equal to 11.372, which is within 2 % of the static lower bound, showing that in this case the static policy we construct is close to optimal.

For the cyclic policy, due to the asymmetric switch-over times, the best cyclic routing is $1 \rightarrow 3 \rightarrow 2$. The average waiting time under this cyclic policy is 11.449.

Under the optimal randomized policy, the server switches from station i to station j with probability $p_{ij} = h'_{ij} / \sum_{k=1}^{N} h'_{ik}$. Thus in our example, randomization takes place only when the server is in station 1. The server switches from station 1 to station 2 with probability $2/5 = 0.4$ and to station 3 with probability $3/5 = 0.6$, and it switches from station 2 to station 1 and from station 3 to station 2 with probability 1 respectively. The switch-over matrix is:

$$P = \begin{bmatrix} 0 & 0.4 & 0.6 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

and the average waiting time under this randomized policy is 11.718.

Table 2: Effects of asymmetry switch-over times on performance for a 3 queue network

$z_{dynamic}$	z_{closed}	z_{static}	$W_{routing\ table}$	W_{cyclic}	$W_{randomized}$
8.33	10.494	11.185	11.372	11.449	11.718

The simulation results are shown in Table 2.

Effects of system utilization

Different levels of system utilization will result in different optimal routing tables. In order to simplify the implementation of the optimal routing table while testing the effect of system utilization ρ on the lower and upper bounds developed in this paper, we use a four station polling system shown in Figure 4, where all arrival process are Poisson and the service requirements are exponential with mean 1 unit of time at all stations. The switch-over times are 1 unit of time between all stations. We would like to keep $\lambda_1 = \lambda_3$ and $\lambda_2 = \lambda_4$ and force the best routing table be: *routing table* $= 1 \to 2 \to 3 \to 1 \to 4 \to 3$, i.e., the frequency of visits to stations 1 and 3 is two times higher than that to stations 2 and 4. We want to select values for λ_1, λ_2 as a function of ρ. Since $\lambda_1 = \lambda_3$ and $\lambda_2 = \lambda_4$, we obtain from equation (27) that

$$\begin{aligned} &\tfrac{m_{j(1),1}}{m_{j(2),2}} = \tfrac{\sqrt{\lambda_1(1-\lambda_1)}}{\sqrt{\lambda_2(1-\lambda_2)}} = 2 \\ &2(\lambda_1 + \lambda_2) = \rho \end{aligned} \tag{39}$$

Solving for λ_1 and λ_2 we obtain

$$\begin{aligned} &\lambda_1 = \lambda_3 = \tfrac{\sqrt{(5-4\rho)^2+12\rho(2-\rho)}-(5-4\rho)}{6} \\ &\lambda_2 = \lambda_4 = \tfrac{\rho}{2} - \lambda_1 \end{aligned} \tag{40}$$

This example is constructed in such a way that the best possible performance of a routing table policy from the solution of (P_{approx}) can be demonstrated.

The simulation results for various values of system utilization ρ are shown in Table 3. Under the cyclic policies server visits stations in a cyclic order; under the randomized policy, the server will next go to station 1 and station 3 with probability 1/3 and to station 2 and station 4 with probability of 1/6, independent of the current station; under the dynamic policy, after serve exhaustively at one station, the server will serve next the most loaded

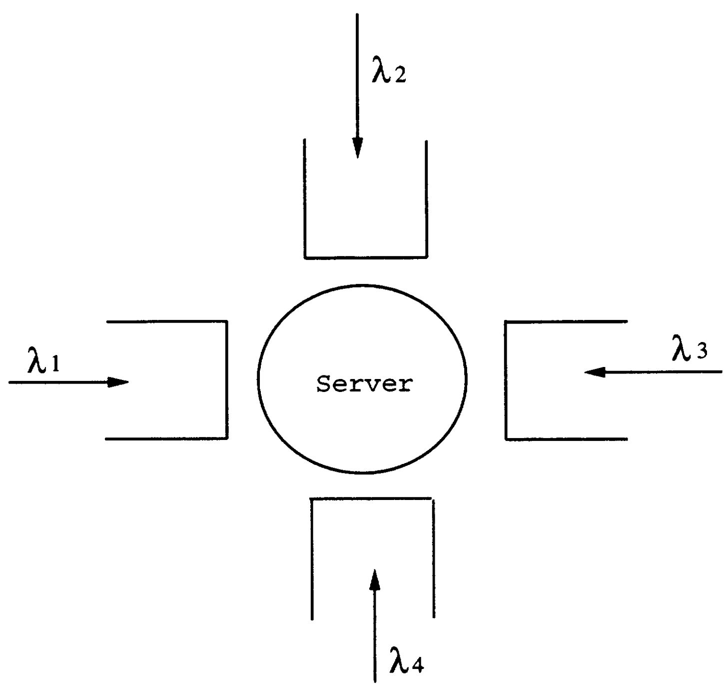

Figure 4: A 4 queue polling system

Table 3: Performance as a function of the system utilization ρ

ρ	$\lambda_1,\ \lambda_3$	$\lambda_2,\ \lambda_4$	z_{static}	$\frac{W_{123143}}{z_{static}}$	$\frac{W_{cyclic}}{z_{static}}$	$\frac{W_{random}}{z_{static}}$	$\frac{W_{dynamic}}{z_{static}}$	$z_{dynamic}$	$\frac{W_{dynamic}}{z_{dynamic}}$
0.05	0.020	0.005	1.915	1.000	1.109	1.707	0.400	0.684	1.120
0.20	0.081	0.019	2.344	1.000	1.099	1.679	0.501	0.994	1.181
0.40	0.164	0.036	3.242	1.000	1.090	1.638	0.646	1.650	1.269
0.60	0.251	0.049	5.021	1.002	1.081	1.605	0.791	2.954	1.344
0.80	0.340	0.060	10.314	1.003	1.072	1.573	0.925	6.875	1.388
0.90	0.387	0.063	20.858	1.003	1.067	1.571	0.989	14.700	1.403
0.98	0.425	0.065	105.094	1.004	1.063	1.532	1.002	77.316	1.362

station. The server will stay idle if there is no customer in the system;

Based on these simulation results the following observations can be made:

1. The routing table policy is almost identical to the lower bound. This should not be surprising as we constructed the arrival rates so that the policy will be optimal. The routing table policy outperforms both the cyclic and the randomized policy significantly. Moreover, as we have not optimized the parameters of the randomized policy, the policy is rather weak.

2. For lower system utilizations the dynamic policy outperforms the static policy significantly. Again, the light traffic lower bound works quite well for low system utilizations. For example, when $\rho = 0.05$, the performance of the dynamic policy is within 12% of the lower bound $z_{dynamic}$.

3. As ρ increases, the optimal static policy becomes closer and closer to the dynamic policy, and for very heavy traffic the static policy is actually better. This confirms our conjecture that the lower bounds developed for static policies in this paper are also valid for dynamic policies in heavy traffic.

5 Conclusions

We proposed methods for proving strong lower bounds on arbitrary static policies based on nonlinear optimization. Moreover, we developed a lower bound for dynamic policies as well. We then used our methods to propose effective routing table policies using integer programming. The simulation experiments we conducted illustrate the following points:

1. The routing policies we construct outperform all other static routing policies and they are at most within 3% from the lower bounds for the cases studied.
2. The performance of the routing table policy improves compared against other static routing policies as the change-over times or the system utilization increases.
3. For lower change-over times and system utilizations dynamic policies outperform static policies by a significant margin. But as the change-over times or system utilization increase, static policies are equally or more effective. This is a rather interesting fact, since at least in the cases that optimal dynamic policies are known (two stations), they are rather complicated threshold class policies (Hofri and Ross [8]).
4. Based on our intuition from the proof of the static lower bound and the numerical results, we conjecture that the static lower bounds developed in this paper are valid for dynamic policies also under heavy traffic conditions.

Our results suggest that as far as static policies are concerned the routing table policies constructed in the paper are adequate for practical problems. As far as dynamic policies are concerned the dynamic lower bound does provide information in lighter traffic, while we believe that in heavier traffic, the routing table policies we construct are adequate for practical problems. We believe that dynamic policies would be very useful in light and intermediate traffic. Having techniques to generate provably near optimal dynamic policies is indeed an important open problem.

References

[1] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, Inc., New Jersey, 1987.

[2] D. Bertsimas and G. van Ryzin, "A stochastic and dynamic vehicle routing problem in the Euclidean plane",*Operations Research*, vol. 39, 4, 601-615, 1991.

[3] D. Bertsimas and G. van Ryzin, "Stochastic and Dynamic vehicle routing in the Euclidean Plane: the multiple-server, capacitated vehicle case", *Operations Research* , 41, 60-76, 1993.

[4] O.J. Boxma, W.K. Groenendijk and J.A. Weststrate, "Pseudoconservation Law for Service systems with a polling table", *IEEE Trans. Commun.*, to appear.

[5] O.J. Boxma and J.A. Weststrate, "Waiting Times in Polling Systems with Markovian Server Routing", technical report, to appear.

[6] O.J. Boxma, H. Levy and J.A. Weststrate, "Optimization of Polling Systems", technical report, to appear.

[7] S. Browne and U. Yechiali, "Dynamic Priority Rules for Cyclic-type Queues", *Adv. Appl. Prob.*, Vol 21, 432-450, 1989.

[8] M. Hofri and K.W. Ross, "On the optimal control of two queues with server set-up times and its analysis", *SIAM J. on Computing*, 16, 399-419.

[9] L. Kleinrock, *Queueing systems, Volume II: Computer applications*, John Willey, New York, 1976.

[10] L. Kleinrock and H. Levy, "The Analysis of random polling systems". Operations Research, 36, 716-732, 1988.

[11] H. Levy and M. Sidi, "Polling Systems: Applications, Modelling and Optimization", to appear in *Queueing systems and applications.*

[12] J. Perkins and P.R. Kumar "Stable, Distributed, Real-Time Scheduling of Flexible Manufacturing/ Assembly/Disassembly System", *IEEE Transactions on Automatic Control*, Vol 34, No 2, 1989.

[13] H. Psaraftis, "Dynamic vehicle routing: status and prospects", working paper, National Technical University of Athens, Greece, 1993.

[14] M. Reiman and L. Wein, personal communication, 1993.

[15] H. Takagi, *Analysis of Polling Systems*, MIT press, 1986.

[16] H. Takagi, *Queueing Analysis of Polling Systems*, *ACM Comput. Surveys*, 20, 5-28, 1988.