# On Elevator Polling with Globally Gated Regime *

**Eitan Altman**
INRIA
Centre Sophia Antipolis
06565 Valbonne Cedex, France

**Asad Khamisy**
Electrical Engineering
Technion — Israel Institute of Technology
Haifa, 32000 Israel

**Uri Yechiali**
Department of Statistics
Tel-Aviv University
Tel-Aviv 69978, Israel

Submitted: December 1990
Revised: October 1991

**Abstract**

We consider a polling system consisting of $N$ queues and a single server where polling is performed according to an **Elevator** (scan) scheme. The server first serves queues in the 'up' direction, i.e. in the order 1,2,...,N-1,N, and then serves these queues in the opposite ('down') direction, i.e. visiting them in the order N,N-1,...,2,1. The server then changes direction again, and so on. A globally gating regime is used each time the server changes direction. We show that, for this Elevator scheme, **the expected waiting times in all channels are equal**. This is the only known non-symmetric polling system that exhibits such a *fairness* phenomenon. We then discuss the problem of optimally ordering the queues so as to minimize some measure of variability of the waiting times.
**Keywords:** Polling, Globally Gating, Elevator policy.

## 1 Introduction

We consider a polling system with $N$ independent channels, where channel $i$ ($i = 1, 2, ..., N$) is modeled as an $M/G/1-$type queue. The arrival stream to queue $i$ is Poisson with rate $\lambda_i$, and

*Supported by a Grant from the France-Israel Scientific Cooperation (in Computer Science and Engineering) between the French Ministry of Research and Technology and the Israeli Ministry of Science and Technology, Grant Number 3321190.

service times are distributed as $B_i$, having Laplace-Stieltjes Transform (LST) $b_i^*(s)$, and first and second moments $b_i$ and $b_i^{(2)}$, respectively. We denote by $\rho_i \stackrel{\text{def}}{=} \lambda_i b_i$, and by $\rho \stackrel{\text{def}}{=} \sum_{i=1}^N \rho_i$ the traffic offered to channel $i$, and to the system at large, respectively.

The time it takes the server to move from the $i$th to the $i+1$st queue is called the $i$th *walking time*, and is denoted by $D_i$. We assume that the time it takes for a movement of the server from queue $i$ to queue $i+1$ has the same distribution as the time it takes to move in the opposite direction, i.e. from queue $i+1$ to queue $i$. We assume that the walking times are independent, with LST $d_i^*(s)$, and first and second moments $d_i$ and $d_i^{(2)}$, respectively. Let $D = \sum_{i=1}^{N-1} D_i$ be the total walking time in one direction, and denote by $d$, $d^{(2)}$ and $d^*(s)$ the expectation, second moment and LST of $D$, respectively.

We consider the Elevator (or scan) polling scheme, where the server first visits the queues in one direction, i.e. in the order 1,2,...,N ('up' direction), and then reverses its orientation and visits the queues in the opposite ('down') direction, i.e. going through queues N,N-1,...,2,1. It then changes its direction again, and so on. This type of service discipline is encountered in many applications. e.g. it models a common scheme of addressing a hard disk for writing (or reading) information on (or from) different tracks (see e.g. Tanenbaum [8] pp. 143-146, for a brief discussion of various techniques for head movement in disks).

Compared with Cyclic polling, the Elevator scheme 'saves' the return walking time from queue N to queue 1. This saving may be a significant factor in decreasing cycle duration and expected waiting times, since in many systems the return walking time from $N$ to 1 is considerably large. In the case of hard disks this return walking time represents the movement of the head all the way back from the $N$th track to the first, which means that the time so waisted may be considerably larger than any other walking time. (The main factor in the access time to a disk is the seek time, which is the time it takes to move the head from one track to another).

All service disciplines that have been considered in the literature with relation to cyclic movement (e.g. the Gated, Exhaustive, Limited, Globally Gated) can also be implemented with the Elevator approach (see Coffman and Hofri [4], Swartz [5] and Takagi and Murata [7]). It should be noted that these Elevator service disciplines are special cases of a general polling table (see, e.g. Baker and Rubin [1]).

We shall consider the following version of the Elevator scheme based on the recently introduced Globally Gated discipline (Boxma, Levy and Yechiali [2]). At the start of the up movement *all* gates are closed and the server, while moving from queue 1 to queue N, serves in each queue only those customers that were present at the instant when the gates were closed. When service ends in queue N (at the end of the up movement), all gates are closed once more and service is immediately given again to the customers present at queue $N$ (i.e. to those who arrived during the period that the server was moving from queue 1 to queue N and was serving these queues). The server then goes back through queues $N, N-1, ..., 2, 1$, serving only those customers that were present when the gates were last closed. When this cycle is finished in queue 1, all gates are immediately closed, service starts in queue 1, and the direction is reversed again.

We shall call the period during which the server moves up an 'up cycle', and the period during which the server moves down a 'down cycle'. A 'cycle' will be either an up cycle or a down cycle.

We show for this Elevator scheme that **the expected waiting times in all channels are equal**. This is the only known non-symmetric polling system that exhibits such a *fairness* phenomenon. We then discuss the problem of optimally ordering the queues so as to minimize some measure of variability of the waiting times.

## 2 Performance Measures and Optimization

### Cycle Duration

**Lemma 1** *The distribution of a cycle duration $C$ (in steady state) does not depend on the direction, and*

$$E[C] = \frac{d}{1-\rho} \tag{1}$$

$$E[C^2] = \frac{1}{1-\rho^2}\left(d^{(2)} + 2d\rho E[C] + \sum_{i=1}^{N} \lambda_i b_i^{(2)} E[C]\right). \tag{2}$$

*Denote* $\gamma(s) \stackrel{\text{def}}{=} E\left[e^{-sC}\right]$ *and* $\delta(s) \stackrel{\text{def}}{=} \sum_{i=1}^{N} \lambda_i (1 - b_i^*(s))$*, then*

$$\gamma(s) = d^*(s)\gamma\left(\delta(s)\right) \tag{3}$$

*Define recursively* $\delta^{(0)}(s) \stackrel{\text{def}}{=} s$ *and* $\delta^{(j)}(s) \stackrel{\text{def}}{=} \delta\left(\delta^{(j-1)}(s)\right)$*, then*

$$\gamma(s) = \prod_{j=0}^{\infty} d^*\left(\delta^{(j)}(s)\right) \tag{4}$$

**Proof:** One can conceive an alternative interpretation of the Elevator Globally Gated scheme. It can be assumed, that when completing serving queue $N$ at the end of the up movement, there is an extra walking time of *zero* duration to queue 1. When the server now arrives at queue 1, all gates are closed, the server jumps to queue N in no time, and then goes down visiting queues $N, N-1, ..., 1$, serving all customers marked when the gates were last closed. After serving queue 1 all gates are again closed and the server goes up again, serving all customers present at the moment the gates were closed, etc.

Consider now a polling system with the same arrival, service times and walking time distributions, but where the server moves *cyclically* between the queues. The walking time between queue N back to queue 1 is assumed zero. Assume that at the beginning of each cycle, i.e. each time the server arrives to queue 1, a global gate is closed, and when a queue is attended, then only customers that were present at the moment when the gate was closed are served. This modified model is a special case of the *Cyclic* Globally Gated polling system analyzed by Boxma, Levy and Yechiali [2], for which (1), (2), (3), (4) hold (see (2.8), (2.9), (2.5) and (2.7) in [2]). For the modified model, it can easily be seen that the cycle duration is unchanged if we alter the order of the queues that are served and/or the order of the walking times. This follows from the fact that the number of customers served in each queue is determined at the instant when the global gate is closed, and therefore is not affected by any change of order of service of queues. In particular, the cycle duration remains unchanged if every second cycle one serves the queues in the order N,N-1,... ,2,1 and reverses the order of the walking times. Hence, the distribution of cycle duration in the Elevator Globally Gated scheme is equal to the distribution of cycle duration in the case of Cyclic Globally Gated service discipline with zero walking time from queue N to queue 1. The Lemma then follows. ■

As in [2] we introduce $C_P$ and $C_R$, the past and residual time, respectively, of a cycle. By following similar arguments as in the Lemma above (see [2] eq. (2.11)) we write

$$E\left[e^{-sC_P}\right] = E\left[e^{-sC_R}\right] = \frac{1-\gamma(s)}{sE[C]} \tag{5}$$

$$E\left[C_P\right] = E\left[C_R\right] = \frac{E[C^2]}{2E[C]} = \frac{1}{1-\rho^2}\left((1-\rho)\frac{d^{(2)}}{2d} + d\rho + \frac{1}{2}\sum_{i=1}^{N}\lambda_i b_i^{(2)}\right) \tag{6}$$

## Waiting Times

We can now derive expressions for the expected waiting times in the Elevator Globally Gated system. Consider an arbitrary customer M at queue $k$. As the distributions of the up and down cycles are the same, with probability 0.5 he arrives during an up cycle, and with probability 0.5 he arrives during a down cycle. Thus, denoting the waiting time by $W_k$, we can write

$$E[W_k] = 0.5\left(E\left[W_k \,\middle|\, \begin{matrix} server \\ moves\ up \end{matrix}\right] + E\left[W_k \,\middle|\, \begin{matrix} server \\ moves\ down \end{matrix}\right]\right) \tag{7}$$

The waiting time, when the server moves down, is composed of (i) the residual cycle time $C_R$, (ii) the service times of all customers who arrive at queues $i < k$ during the cycle in which M arrives, (iii) the walking times from queue 1 to queue $k$, (iv) the service times of all customers who arrive at queue $k$ during the past part $C_P$ of the cycle in which M arrives. The four terms are identical to those appearing in the Cyclic Globally Gated discipline. Denoting by $W_k^{(m)}$ the $m$th component of the waiting time of M, from equation (2.17) of Boxma, Levy and Yechiali [2], the sum of the expectations of the four terms is given by

$$\sum_{m=1}^{4} E\left[W_k^{(m)}\right] = E[W_k|down] = (1 + 2\sum_{i=1}^{k-1}\rho_i + \rho_k)E[C_R] + \sum_{i=1}^{k-1} d_i \tag{8}$$

Arguing similarly for the case where M arrives while the server is moving up, we have

$$E[W_k|up] = (1 + 2\sum_{i=k+1}^{N}\rho_i + \rho_k)E[C_R] + \sum_{i=k}^{N-1} d_i \tag{9}$$

Combining (7), (8) and (9) we obtain

$$E[W_k] = (1+\rho)E[C_R] + 0.5d \tag{10}$$

That is, the expected waiting time is *equal for all queues*. This is the only known non-symmetric polling system that exhibits such a *'fairness'* phenomenon. (For further discussion on fairness, the reader is referred to Boxma [3]). An explanation of Equation (10) is the following: $E[W_k] = E[C_R] + 0.5d+$ mean service in same cycle before customer M; the last term follows immediately by adding the 2nd and 4th term in $E[W_k]$ in the decomposition above.

Comparing the Cyclic Globally Gated regime with the corresponding Elevator Globally Gated regime we have

$$E\left[W_k \,\middle|\, \begin{array}{c} Cyclic \\ Globally \\ Gated \end{array}\right] - E\left[W_k \,\middle|\, \begin{array}{c} Elevator \\ Globally \\ Gated \end{array}\right] =$$

$$\left(\sum_{i=1}^{k-1}\rho_i - \sum_{i=k+1}^{N}\rho_i\right)E[C_R] + 0.5\left(\sum_{i=1}^{k-1} d_i - \sum_{i=k}^{N-1} d_i\right) \tag{11}$$

**Optimization in the Elevator Scheme**

Let $a_i \stackrel{\text{def}}{=} 2E[C_R]\rho_i + d_i, \quad (i = 1, 2, ..., N)$. Then,

$$E[W_k|down] = E[C_R](1 + \rho_k) + \sum_{i=1}^{k-1} a_i$$

$$E[W_k|up] = E[C_R](1 + \rho_k) + \sum_{i=k+1}^{N} a_i + d_k$$

It follows that

$$\Delta_k \stackrel{\text{def}}{=} E[W_k|down] - E[W_k|up] = \sum_{i=1}^{k-1} a_i - \sum_{i=k+1}^{N} a_i - d_k$$

and it is clear that $\Delta_k$ is an increasing function of $k$.

One goal is to arrange the channels such that $\max_{1 \le k \le N} \{|\Delta_k|\}$ is as small as possible, as $|\Delta_k|$ is a measure of the variation in waiting times incurred in channel $k$. Let $\pi_0 = \{1, 2, ..., N\}$ be an order of the channels. Now, under any order of queues 2 to N, we have

$$\Delta_1 = -\sum_{i=2}^{N} a_i - d_1 = -A + a_1 - d_1 < 0$$

$$\Delta_N = \sum_{i=1}^{N-1} a_i - d_N = A - a_N - d_N > 0$$

where, $A \stackrel{\text{def}}{=} \sum_{i=1}^{N} a_i$ . Therefore, since $d_N = 0$

$$\max_{1 \le k \le N} \{|\Delta_k|\} = \max(|\Delta_1|, |\Delta_N|) = \max\{A - a_1 + d_1, A - a_N - d_N\}$$

$$= \max\{A - (2E[C_R]\rho_1), A - (2E[C_R]\rho_N)\} \qquad (12)$$

It follows from (12) that $\max_{1 \le k \le N} \{|\Delta_k|\}$ is *minimized* if channel 1 is the one with the *highest* value of $\rho_i$ and channel $N$ is the one with the *second highest* value of $\rho_i$, or vice versa.

A question that still remains open is how to arrange the channels such that the *sum* of the absolute values of $\Delta_k$, i.e. $\sum_{k=1}^{N} |\Delta_k|$, is minimized.

## References

[1] J. Baker and I. Rubin, “Polling with a General-Service Order Table”, *IEEE Trans. on Communications*, **35**, (1987) 283-288.

[2] O. J. Boxma, H. Levy and U. Yechiali, “Cyclic Reservation Schemes for Efficient Operation of Multiple-Queue Single-Server Systems”, to appear in *Annals of Operations Research*, 1992.

[3] O. J. Boxma, "Analysis and Optimization of Polling Systems”, *Queueing, Performance and Control in ATM*, J. W. Cohen and C. D. Pack (eds.), North-Holland (1991) 173-183.

[4] E. G. Coffman and M. Hofri, “On the Expected Performance of Scanning Disks”, *SIAM J. on Computing* **11**, (1982) 60-70.

[5] G. B. Swartz, “Analysis of a SCAN Policy in a Gated Loop System”, *Applied Probability – Computer Science: The Interface, Vol. II*, R. L. Disney and T. J. Ott (eds.), Birkhauser (1982) 241-252.

[6] H. Takagi, *Analysis of Polling Systems*, The MIT Press, 1986.

[7] H. Takagi and M. Murata, “Queueing Analysis of Scan-Type TDM and Polling Systems”, *Computer Networking and Performance Evaluation*, T. Hasegawa et al. (eds.) North-Holland, 1986.

[8] A. S. Tanenbaum , *Operating Systems, Design and Implementation*, Prentice-Hall, 1987.