Designing a Learning-Automata-Based Controller for Client/Server Systems: A Methodology

G.I.Papadimitriou, A.I.Vakali and A.S.Pomportsis

Abstract— **Polling policies have been introduced to simplify the accessing process in client/server systems by a centralized control access scheme. This paper considers a client/server model which employs a polling policy as its access strategy. We propose a learning-automata-based approach for polling in order to improve the throughput-delay performance of the system. Each client has an associated queue and the server performs selective polling such that the next client to be served is identified by a learning automaton. The learning automaton updates each client's choice probability according to the feedback information. Under the considered approach, a client's choice probability asymptotically tends to be proportional to the probability that this client is ready. Simulation results have shown that the proposed polling policy is beneficial in comparison to the conventional round-robin polling when operating under bursty traffic conditions. The benefits are significant for the delay reduction in the considered client/server system.**

Keywords—**client/server systems, polling policies, learning automata, time-delay, throughput improvement.**

I. Introduction

MANY computer and communication systems have considered polling due to its applicability and effectiveness. A typical polling model considers a number of queues served by a single server that visits the queues in a round-robin fashion to perform servicing of the requests waiting at the clients queues.

Polling models have been studied extensively and their applicability to computer communication networks has been investigated [1]-[3]. More specifically, in [1] asymmetric cyclic polling systems are considered with an arbitrary number of queues in heavy traffic. Closed form expressions are derived which explicitly characterize the complete waiting-time distributions at each of the queues. Furthermore, an extensive overview of the applicability of polling models is given in [3]. A novel approach to queue stability analysis of polling models is introduced in [4], based on a concept queue stability orderings and it is shown that stabilities of any two queues can be compared based on the queue arrival rate. Multiple server polling systems have been also investigated such that each server visits the queues according to its own cyclic schedule [5].

Optimization of polling policies has been studied before in order to minimize clients delays. In [6] a server in a multiple queues polling system is scheduled according to a neural network policy and numerical results have shown that this approach is quite beneficial especially for asymmetric polling systems. However, this approach is based on the assumption that the controller has global knowledge of the states of the clients' queues. An assumption which is difficult to be satisfied, especially when the number of clients is large. Optimal polling in communication networks is studied in [7], where optimal polling algorithms are presented for several classes of graphs. Also the lower bound on the time complexity of any polling algorithm for any graph and polling station is estimated.

The authors are with the Department of Informatics, Aristotle University, 54006 Thessaloniki, Greece. E-mail: gp@csd.auth.gr.

Here, we concentrate on an adaptive polling model which is based on the use of Learning Automata (LA) [8]-[16]. The Learning automata approach has been successfully applied in various topologies and systems, including optical networks [13],[14], broadcast communication systems [10] and multiple disk subsystems [15],[16].

The present paper considers a new polling algorithm in order to improve the accessing process in a client/server environment. Our approach is based on the following key issues:

- the servicing/response time could be improved by reordering of the polling access cycle among clients,
- due to the burstiness of traffic, it is possible to predict the behavior of each client in the near future, based on its behavior in the recent past.
- the system's performance could be improved if each client is being polled with a probability proportional to the probability that its queue is not empty. In this way, each client takes an amount of the system resources (communication bandwidth and service time) proportional to its needs. In this way, the number of unsuccessful polls is decreased and consequently, the performance of the system is improved.

The remainder of the paper is organized as follows. The next section presents the learning automata based model and the performance metrics are defined. In Section III the asymptotic analysis of the proposed model is carried out and theorem proofs are given. Section IV presents the simulation results and discusses the performance improvements due to the presented polling model. Finally, conclusions are summarized and future work areas are suggested in Section V.

II. The Learning-Automata-Based Polling Algorithm (LPA)

Requests arrive to the system randomly, by various independent processes. Some requests arrive while others are being serviced, and so queues are created in each client. Requests arrival rate could be either constant or independent and exponentially distributed or bursty. Overall, a polling system is considered as a system that includes servers, queues and clients. The clients enter the different queues

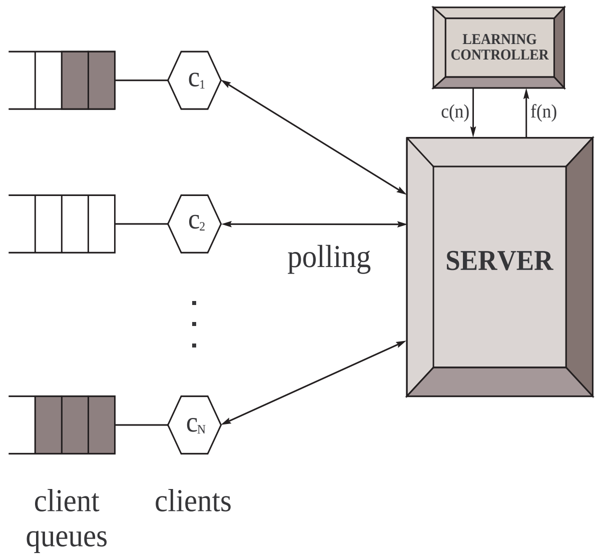

Fig. 1. The Learning-automata-based polling model.

and wait for service. A server polls the clients to find out if there are clients that need service and serves the clients if needed. The order of polling can be pre-determined (cyclic or by table) or random. The most common policies for serving customers in a queue are:

- *Exhaustive :* in which the server serves the queue until it becomes completely empty.
- *Gated :* in which the server serves in a given service period all the customers which were at the queue when the queue was polled.
- *Limited-1 :* in which at most one customer is served in each visit to the queue.

The latter policy is considered in this paper. Under round-robin polling (RRP) clients are visited and questioned by the server in a sequential order.

When the polled client has a non-empty queue it transmits a positive acknowledgement, followed by the first job of its queue. When the job has been serviced, the server sends a polling message to the next client and once all clients have been polled, a cycle is completed and a new polling cycle begins. The time T_P which is required for a successful poll (including the job service time) is given by the relation:

$$T_P = t_{poll} + t_{prop} + t_{proc} + t_{ack} + t_{data} + t_{prop} + t_{service} \quad (1)$$

The time parameters that appear in this relation are defined in Table I.

When the polled client has no jobs in its queue, then it responds with a negative acknowledgement. When the server receives the negative acknowledgement, it continues by polling the next station. The time T_N which is spent in an unsuccessful poll is given by the relation:

$$T_N = t_{poll} + t_{prop} + t_{proc} + t_{nack} + t_{prop} \quad (2)$$

TABLE I
THE BASIC TIME METRICS.

Parameter	Description
t_{poll}	time to transmit a poll
t_{prop}	propagation time
t_{proc}	time to process a poll before acknowledging
t_{ack}	time to transmit a poll acknowledgement
t_{data}	time to transfer the data
$t_{service}$	time to service a client's request
t_{nack}	time to transmit a negative poll acknowledgement

TABLE II
THE BASIC PARAMETERS.

Parameter	Description
B	mean burst length
c_i	identity of client i
N	number of clients
Q	queue capacity
$P_i(n)$	basic choice probability of client i at poll n
d_i	probability that client i is not idle

Here, we propose an adaptive polling policy where the polling order is determined by means of learning automata. We consider a single server polling system of N clients, with each client having an associated queue. The arrivals at the clients' queues are assumed to be bursty.

The server is provided with a learning automaton which determines the basic choice probability $P_i(n)$ of each client c_i $(i = 1, \ldots, N)$, at a certain poll n. Figure 1 depicts the LA based polling process on a client/server system with a single server and N clients. The most important model's parameters are described in Table II.

At each poll n, the client $c(n)$ which grants permission to be served is selected according to the normalized choice probabilities $\Pi_i(n)$ for $i = 1, \ldots, N$, where:

$$\Pi_i(n) = \frac{P_i(n)}{\sum_{k=1}^{N} P_k(n)} \quad (3)$$

If client $c(n) = c_m$ has a non-empty queue, then it transmits the top job of its queue. After the receipt of the job, the server immediately begins servicing it, while the basic choice probability $P_m(n)$ of client c_m is updated. When the job has been serviced the polling process is continued. The server selects a new client $c(n)$ based on the new choice probabilities and sends a poll message to it. On the other hand, if the queue of client $c(n) = c_m$ is empty, then the basic choice probability $P_m(n)$ of the selected client c_m is updated, and a new client is selected according to the new choice probabilities.

The probability updating scheme is based on the feedback information $f(n)$. If the selected client had no job to sent ($f(n) = 0$) then, due to the burstiness of traffic, it is probable that this client will remain idle in the near future. Therefore, its choice probability is decreased. On the other hand, if the selected client responded to the poll by sending a job to be serviced ($f(n) = 1$), then, due to the burstiness of traffic it is probable that this client will remain active in the near future. Therefore, its choice probability is increased. The following probability updating scheme is used [10] (where $L, a \in (0, 1)$):

$$P_i(n+1) = \begin{cases} P_i(n) + L(1 - P_i(n)) & \text{if f(n)=1} \\ P_i(n) - L(P_i(n) - a) & \text{if f(n)=0} \end{cases} \quad (4)$$

Parameter a is introduced to improve the adaptivity of the proposed LPA scheme, since it is used to prevent the client choice probability values to be taken from the neighborhood of 0. This is needed since once a choice probability (let $P_i(n)$) converges to 0 it is highly probable that this client will not be involved in the polling cycle for a long period. Thus, in case that the client c_i has an arrival at its queue again, it will become a poll candidate and the server will need to be notified accordingly by the LA process.

III. Asymptotic Analysis

The learning-automata-based polling algorithm updates the clients choice probabilities by considering their queue status as feedback. In the present section we will prove that in a client server model, the choice probability of polling at a client converges to the probability of having a non-empty queue of messages at this client. The following theorem (presented in [9]) is needed to carry out the asymptotic analysis :

Theorem 1: Let $x(n)_{n\geq 0}$ be a stationary Markov process dependent on a constant parameter $\theta \in [0, 1]$. Each $x(n) \in I$, where I is a subset of the real line. Let $\delta x(n) = x(n+1) - x(n)$. The following are assumed to hold:
(i) I is compact.
(ii) $E[\delta x(n)|x(n) = y] = \theta\ \omega(y) + O(\theta^2)$.
(iii) $E[|\delta x(n)|^2\ \mid x(n) = y] = \theta^2\ b(y) + O(\theta^2)$.
(iv) $E[|\delta x(n)|^3\ \mid x(n) = y] = O(\theta^3)$, where:

$$\sup_{y\in I} \frac{O(\theta^k)}{\theta^k} < \infty \text{ for } k = 2, 3 \text{ and } \sup_{y\in I} \frac{O(\theta^2)}{\theta^2} \to 0 \text{ as } \theta \to 0$$

(v) $\omega(y)$ has a Lipschitz derivative in I.
(vi) $b(y)$ is Lipschitz in I.

If assumptions (i)-(vi) hold, $\omega(y)$ has a unique root y^* in I and $d\omega/dy|_{y=y^*} < 0$, then:
(a) $var[x(n)|x(0) = x] = O(\theta)$ uniformly for all $x \in I$ and $n \geq 0$.
(b) For any $x \in I$ the differential equation $\frac{dy(\tau)}{d\tau} = \omega(y(n))$ has a unique solution $y(\tau) = y(\tau, x)$ with $y(0) = x$ and $E[x(n)|x(0) = x] = y(n\theta) + O(\theta)$ uniformly for all $x \in I$ and $n \geq 0$.
(c) $(x(n) - y(n\theta))/\sqrt{\theta}$ has a normal distribution with zero mean and finite variance as $\theta \to 0$ and $n\theta \to \infty$.

Theorem 2: Under the learning-automata-based polling algorithm in a client server model, the choice probability of polling at client c_i converges to the probability of having a non-empty queue of messages at client c_i. If the learning algorithm (2) is used and d_i is the probability that client c_i is not idle (for $i = 1, \ldots, N$), then for any client c_i:

$$\lim_{n\to\infty, L\to 0, a\to 0} P_i(n) = d_i$$

Proof. We use Theorem 1 to the proof of the current theorem. Here we have to identify $x(n)$ (of Theorem 1) with $P_i(n)$, θ (of Theorem 1) with L and I (of Theorem 1) with $(a, 1)$. We have:

$$\begin{aligned} &E[\delta P_i(n)|P_i(n) = P_i] \\ &= \frac{P_i}{\sum_{k=1}^N P_k}(d_i L(1 - P_i) - (1 - d_i)L(P_i - a)) \\ &= L\frac{P_i}{\sum_{k=1}^N P_k}(-P_i + d_i + a(1 - d_i)) = L\omega(P_i) \end{aligned} \quad (5)$$

$$\begin{aligned} &E[|\delta P_i(n)|^2|P_i(n) = P_i] \\ &= L^2\frac{P_i}{\sum_{k=1}^N P_k}(d_i(1-P_i)^2 + (1-d_i)(P_i-a)^2) = L^2 b(P_i) \end{aligned} \quad (6)$$

$$\begin{aligned} &E[|\delta P_i(n)|^3|P_i(n) = P_i] \\ &= L^3\frac{P_i}{\sum_{k=1}^N P_k}(d_i(1-P_i)^3 + (1-d_i)(P_i-a)^3) = O(L^3) \end{aligned} \quad (7)$$

The functions $\omega(P_i)$ and $b(P_i)$ are defined as follows:

$$\omega(P_i) = \frac{P_i}{\sum_{k=1}^N P_k}(-P_i + d_i + a(1 - d_i)) \quad (8)$$

$$b(P_i) = \frac{P_i}{\sum_{k=1}^N P_k}(d_i(1 - P_i)^2 + (1 - d_i)(P_i - a)^2) \quad (9)$$

It is immediately seen that assumptions (i)-(iv) are satisfied. It can also be proved that $b(P_i)$ and $\omega'(P_i)$ are Lipschitz in (a,1) by showing that their first derivatives ($b'(P_i)$ and $\omega''(P_i)$ correspondingly) are bounded [17] for $P_i \in (a, 1)$.

It remains to show that $\omega(P_i)$ has a unique root P_i^r near the point $P_i^* = d_i$ and that $d\omega(P_i)/dP_i|_{P_i=P_i^r} < 0$. It is immediately seen that $\omega(P_i)$ has a unique root at the point $P_i^r = d_i + a(1 - d_i)$. Since a can be arbitrarily small, it follows that P_i^r is in the neighborhood of the point $P_i^* = d_i$. The derivative of $\omega(P_i)$ at this point is:

$$\frac{d\omega(P_i)}{dP_i}|_{P_i=P_i^r} = \frac{d\left(\frac{P_i}{\sum_{k=1}^N P_k}(-P_i+d_i+a(1-d_i))\right)}{dP_i}|_{P_i=P_i^r}$$

$$= -\frac{1}{1+\frac{\sum_{k=1,k\neq i}^{N} P_k}{P_i^r}} < 0 \qquad (10)$$

It has been shown that $\omega(P_i)$ has a unique root P_i^r in the neighborhood of the point $P_i^* = d_i$ and that the derivative of $\omega(P_i)$ at this point is negative.

If we set $P_i(\tau) = P_i^r$, the differential equation $\frac{dP_i(\tau)}{d\tau} = \omega(P_i(\tau))$ is satisfied (0=0). Thus, $P_i(\tau) = P_i^r$ is a solution of the above differential equation. From Theorem 2, it is derived that this solution is unique, thus all the solutions starting in $(a, 1)$ of the differential equation $\frac{dP_i(\tau)}{d\tau} = \omega(P_i(\tau))$ converge to the point $P_i(\tau) = P_i^r \simeq P_i^* = d_i$. According to Theorem 2, we have:

$$\lim_{n\to\infty, a\to 0} E[P_i(n)] = P_i^* + O(L)$$

and

$$var[P_i(n)] = O(L) \qquad \text{for all } t.$$

Consequently,

$$\lim_{n\to\infty, L\to 0, a\to 0} P_i(n) = d_i \qquad \text{q.e.d.} \qquad (11)$$

The exact values of a and L depend on the environment where the automata operate. When the environment is slowly switching or when the environmental responses have a high variance, a and L must be very close to 0 in order to guarantee a high accuracy. On the other hand, in a rapidly switching environment or when the variance of the environmental responses is low, higher values of a and L can be used, in order to increase the adaptivity of the protocol. Thus, when the burst length is high or the queue length is low, then small values of a and L must be selected. On the other hand, when the burst length is low or when the queue length is high, then a and L can be much higher.

According to Theorem 2, for any two clients c_i and c_j (with $d_j \neq 0$), the learning-automata-based polling algorithm asymptotically tends to satisfy the relation:

$$\frac{P_i}{P_j} = \frac{d_i}{d_j} \qquad (12)$$

This relation also holds for the normalized choice probabilities Π_i and Π_j:

$$\frac{\Pi_i}{\Pi_j} = \frac{\frac{P_i}{\sum_{k=1}^{N} P_k}}{\frac{P_j}{\sum_{k=1}^{N} P_k}} = \frac{P_i}{P_j} = \frac{d_i}{d_j} \qquad (13)$$

IV. Simulation Results

In the following, the proposed Learning-automata-based Polling Algorithm (*LPA*) is compared to the well-known Round-Robin Polling (*RRP*) scheme [18]. The polling schemes which are under comparison were simulated to be applied to four client/server systems (namely, S_1, S_2, S_3 and S_4) under bursty traffic conditions.

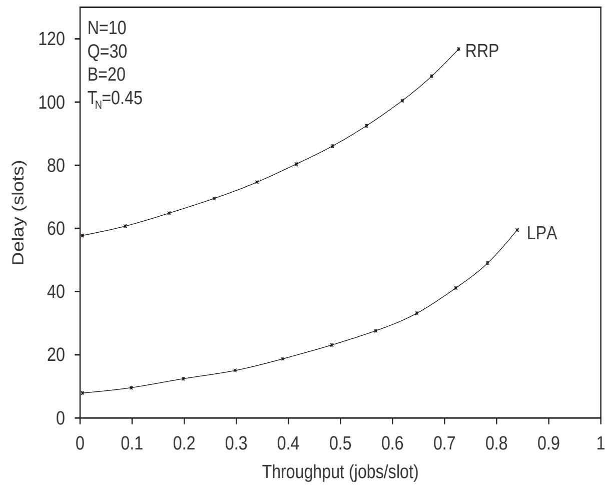

Fig. 2. The Delay versus Throughput characteristics of LPA and RRP when applied to system S_1.

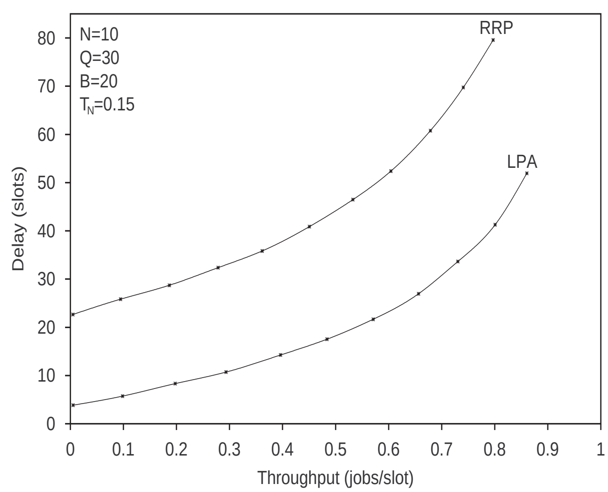

Fig. 3. The Delay versus Throughput characteristics of LPA and RRP when applied to system S_2.

The bursty traffic was modelled in a way similar to the ones presented in [19] and [20]. The time axis of the arrival process is assumed to be slotted with an *arrival slot* size equal to T_P. (Note: The LPA scheme is unslotted. Slots are used only for the modeling of the arrival process. In the rest of the paper "arrival slots" are simply called "slots"). Each client can be in one of two states X_0 and X_1. When a client is in state X_0 then it has no job arrivals. When a client is in state X_1 then, at each slot, it has a job arrival with probability Z. Given a client is in state X_0 at slot t, the probability that this client will transit to state X_1 at the next slot is P_{01}. The transition probability from state X_1 to state X_0 is P_{10}. It can be shown that, when the load offered to the system is R jobs/slot and the mean burst length is B slots, then the transition probabilities are: $P_{10} = 1/B$ and $P_{01} = \frac{R}{B(NZ-R)}$. The total service time of each job (including the job transmission time and the job

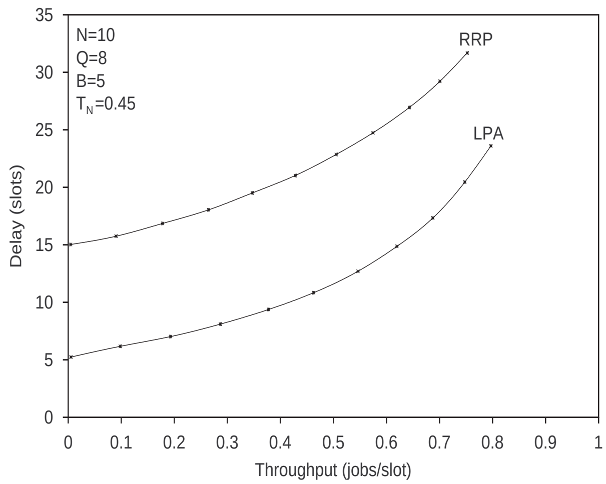

Fig. 4. The Delay versus Throughput characteristics of LPA and RRP when applied to system S_3.

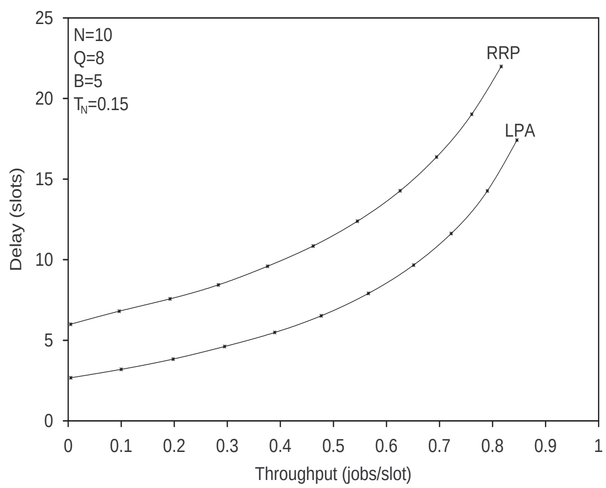

Fig. 5. The Delay versus Throughput characteristics of LPA and RRP when applied to system S_4.

service time) is assumed to be an exponentially distributed random variable with a mean equal to $(T_{data} + T_{service})$.

For all simulated systems the number of clients N was taken to be equal to 10, while the traffic parameter Z was assumed to be equal to 1. The queue capacity Q, the mean burst length B and the time spent on an unsuccessful polling T_N were taken to be as follows:

a) System S_1: Q=30 jobs, B=20 jobs, T_N=0.45 slots.
b) System S_2: Q=30 jobs, B=20 jobs, T_N=0.15 slots.
c) System S_3: Q=8 jobs, B=5 jobs, T_N=0.45 slots.
d) System S_4: Q=8 jobs, B=5 jobs, T_N=0.15 slots.

The parameters of the simulated systems were selected in such a way, that, the reader of the paper can study how the change of parameters B and T_N affects the performance of each polling scheme.

We have used the delay versus throughput characteristics in order to compare the two protocols. The delay versus throughput characteristics of the compared schemes when they are applied to systems S_1, S_2, S_3 and S_4 are appeared at Figures 2, 3, 4 and 5, correspondingly.

The following results can be extracted from the above graphs:

1) The proposed LPA scheme achieves a significantly higher delay-throughput performance than the RRP scheme, when operating under bursty traffic. According to the RRP scheme, all clients are polled in a round-robin fashion. On the other hand, LPA is based on the system feedback information in order to poll clients that are most likely to have at least one job in their queues. In this way, the number of unsuccessful polls is decreased and consequently the performance of the polling system is significantly improved.

2) From a comparison of Figures 2 and 3 with figures 4 and 5, correspondingly, it becomes clear that the performance improvement which is achieved by the use of the learning-automata-based scheme is higher when the offered traffic is more bursty (i.e. when B is high). As the traffic becomes more bursty, the number of idle clients increases. Under these conditions, if the classic RRP scheme is used, the number of unsuccessful polls dramatically increases, resulting to a significant performance degradation. On the other hand, LPA is practically unaffected from the burstiness of the traffic because it is capable of using the system feedback information, instead of blindly selecting the clients which are polled.

3) From a comparison of Figures 2 and 4 with figures 3 and 5, correspondingly, it is derived that the performance advantage of LPA over the RRP scheme is even higher when the time T_N which is spent for an unsuccessful poll is high. When T_N is high, then the performance improvement which is achieved by reducing the number of unsuccessful polls is also high.

Therefore, LPA achieves a significantly higher performance than RRP. The performance advantage of LPA over the RRP scheme is higher when the burst length is high or when the time which is spent for an unsuccessful poll is high.

V. Conclusions - Further Research

This paper has presented a new adaptive polling policy for client/server systems. According to the proposed polling policy the polling cycle is determined by each client's feedback evaluated by means of a learning automata scheme. The server supports the learning automata by an effective probability update scheme adapted to each client's traffic state. The new polling policy is capable of achieving lower delays and a high throughput under bursty traffic conditions.

Future work could extend the proposed polling idea to other systems such as on a client/server system with multiple servers and multiple clients as in [5]. In such a system, we could introduce a polling process such that each of the servers will visit the queues according to a learning-automata-based scheme which will determine each server's polling cycle.

References

[1] R.D. van der Mei,"Distribution of the delay in polling systems in heavy traffic", *Performance Evaluation Journal*, Vol. 38, pp.133-148, 1999.

[2] H.Takagi,"Queueing analysis of polling models: an update", *Stochastic Analysis of Computer and Communication Systems*, pp.267-318, 1990.

[3] H.Levy and M.Sidi,"Polling models: applications, modeling and optimization", *IEEE Transactions on Communications*, Vol. 38, pp.1750-1760, 1991.

[4] R.K.C.Chang and S.Lam, "A novel approach to queue stability analysis of polling models", *Performance Evaluation Journal*, Vol. 40, pp.27-46, 2000.

[5] S.C.Borst and R.D. van der Mei, "Waiting-time approximations for multiple-server polling systems", *Performance Evaluation Journal*, Vol. 31, pp.163-182, 1998.

[6] Y.Matsumoto, "On Optimization of Polling Policy Represented by Neural Network", *ACM SIGCOMM Conference Proceedings*, London, UK, pp.181-190, 1994.

[7] A.A.Rescigno, "Optimal Polling in Communication Networks", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 8, No. 5, pp.449-461, May 1997.

[8] K.Najim and A.S.Poznyak, "*Learning Automata: Theory and Applications*, New York, NY.,Pergamon, 1994.

[9] K.S.Narendra and M.A.L.Thathachar, "On the behavior of a learning automaton in a changing environment with application to telephone traffic routing", *IEEE Transactions on Systems, Man and Cybernetics*, vol.SMC-10, no.5, May 1980, pp.262-269.

[10] G.I.Papadimitriou and A.S.Pomportsis, "Learning-Automata-Based TDMA Protocols for Broadcast Communication Systems", *IEEE Communication Letters*, Vol. 4, No. 3, pp.107-109, March 2000.

[11] G.I.Papadimitriou, "A New Approach to the Design of Reinforcement Schemes for Learning Automata: Stochastic Estimator Learning Algorithms", *IEEE Transactions on Knowledge and Data Engineering*", vol.6, no.4, pp.649-654, August 1994.

[12] G.I.Papadimitriou, "Hierarchical Discretized Pursuit Nonlinear Learning Automata with Rapid Convergence and High Accuracy", *IEEE Transactions on Knowledge and Data Engineering*", vol.6, no.4, pp.654-659, August 1994.

[13] G.I.Papadimitriou and A.S.Pomportsis, "Self-Adaptive TDMA Protocols for WDM Star Networks: A Learning-Automata-Based Approach", *IEEE Photonics Technology Letters*, vol.11, no.10,pp.1322-1324, October 1999.

[14] G.I.Papadimitriou and D.G.Maritsas, "Learning Automata-Based Receiver Conflict Avoidance Algorithms for WDM Broadcast-and-Select Star Networks", *IEEE/ACM Transactions on Networking*, vol.4, no.3, pp.407-412, June 1996.

[15] A.I.Vakali, G.I.Papadimitriou, "An Adaptive Model for Parallel I/O Processing", *Proceedings of the IASTED International Conference, Parallel and Distributed Computing and Systems, PDCS 99*, pp. 139-142, 1999.

[16] A.I.Vakali, G.I.Papadimitriou and A.S.Pomportsis, "A New Approach to the Design of High Performance Multiple Disk Subsystems Dynamic Load Balancing Schemes", Proceedings of the High Performance Computing and Networking (HPCN Europe 2000), to appear in *Lectures Notes Computer Science*, Springer Verlag, May 2000.

[17] S.M.Nikolsky, "*A Course of Mathematical Analysis*", MIR Publishers, Moscow 1977.

[18] I.Frigui and A.S.Alfa, "Analysis of a time-limited polling system", *Computer Communications*, Vol. 21, pp.558-571, 1998.

[19] S.L.Danielsen, C.Joergensen, B.Mikkelsen and K.E.Stubkjaer, "Analysis of a WDM Packet Switch with Improved Performance Under Bursty Traffic Conditions Due to Tuneable Wavelength Converters", *IEEE Journal of Lightwave Technology*, vol.16, no.5, May 1998, pp.729-735.

[20] M.W.McKinnon, G.N.Rouskas and H.G.Perros, "Performance Analysis of a Photonic Single-Hop ATM Switch Architecture with Tunable Transmitters and Fixed Frequency Receivers", *Performance Evaluation*, vol. vol.33, no.5, June 1998.