A prediction-based scheme for wireless push systems, using a statistical Hidden Markov Model

V. Kakali, P. G. Sarigiannidis, *Member*, *IEEE*, G. I. Papadimitriou, *Senior Member*, *IEEE*, and A. S. Pomportsis
Department of Informatics, Aristotle University of Thessaloniki
Box 888, 54124 Thessaloniki, Greece

Abstract

In this paper a Hidden Markov Model (HMM) is applied at the broadcast server in order to provide accurate decisions for a wireless push system environment with unknown client demands. Clients are organized into groups and may request a set of items. The server side sends an item per time and then clients respond with a feedback, if the transmitted item is the desirable one. The novel model tries to adapt quickly and precisely to the dynamic changes of the clients' demands. Initially, the suggested scheme learns the intentions of the connected clients for a specific number of broadcasts and then begins to predict the clients' requests. Finally, the server side follows the results of the prediction procedure, by sending the appropriate items. Concurrently, the prediction module keeps a request history, which is updated by the clients' feedbacks. The presented scheme is compared with a learning automata-based scheme and the simulation results indicate that the novel scheme induces an improvement, in terms of mean response time.

1. Introduction

In the field of data broadcasting (e.g. traffic information), push systems [1-3] have appeared to be the main approach that is able to provide high scalability and client hardware simplicity. The "pure" push systems [1] are considered to have an a-priori estimation of the clients' demands and make broadcasts according to these estimations. Thus, they seem unable to operate efficiently in environments with unknown, dynamic client demands. In such environments, a mechanism able to adapt to the clients' demands or predict them would achieve an increased performance of the system. The adaptive push systems of [2] and [3] achieve adaptivity and, thus, significant improvement of the performance in applications characterized by a-priori unknown and dynamic client demands using a Learning Automaton [4] at the broadcast server that is able to adapt to the altering clients' needs. After each item broadcast, every satisfied client sends back to the server a feedback (e.g. one bit), using Code Division Multiple Access (CDMA). In this work a set of HMMs is proposed in order to provide a transmission schedule, predicting the most possible (most desirable to the clients) items. HMMs have widely used for prediction or pattern recognition [5-7].

2. The Prediction-based scheme

Broadcasts are organized into transmission frames. Each frame comprises of two phases, the round robin phase and the prediction phase. During the round robin phase the server side broadcasts all the items (of each group) once per time. Then the prediction phase begins and the transmission items are selected according to the output of the predictor component (Fig. 1). It has been found empirically that the peak system performance is reached when the length of the prediction phase is three times larger than the one of the round robin phase. In other words, the transmission frame is divided into two parts, the first 25% of broadcasts belongs to the round robin phase and the rest 75% is dedicated to the prediction phase.

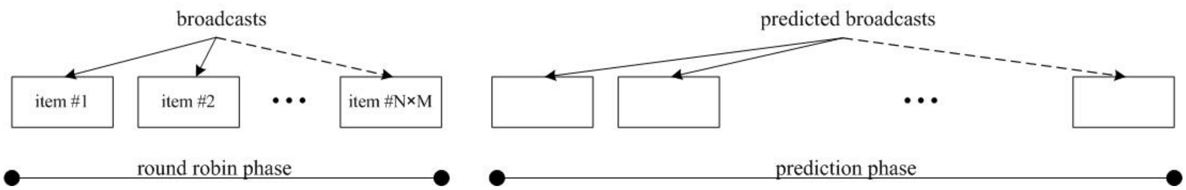

Figure 1. The transmission frame structure.

The purpose of the prediction module is to adapt to the clients' requests as accurately as possible. If we assume that the system comprises of N groups with M possible items per group then NxM parallels HMMs are utilized, i.e., one predictor module per group per item. For each predictor module a set of 20 distinct states are considered, which denote intervals of acceptance:

$S_z^{i,j} = \{S_0^{i,j}, S_1^{i,j}, \ldots, S_{19}^{i,j}\}$, where $1 \le i \le N$, $1 \le j \le M$, $0 \le z \le 19$.

The actual population of each group is unknown, hence estimation about the group population is assumed. If the total population of the connected clients is P then the theoretical proportion is P/N, i.e., P/N clients belong to each group. So, the 20 acceptance intervals cover a range from 0% to 200% of P/N, with equal amount of 10% for each interval. For example, state $S_0^{3,2}$ refers to the second item of the third group and represents an acceptance percentage (satisfied clients) between 0% and 10% of the P/N (theoretical) population of the third group. In the same manner, state $S_1^{3,2}$ represents the acceptance percentage between 10% and 20% of the P/N (theoretical) population and so on. Also, the state transition probability distribution is considered, as follows:

$A_{u,y}^{i,j} = P[f_{t'} = S_y^{i,j} \mid f_t = S_u^{i,j}]$, where $1 \le i \le N$, $1 \le j \le M$, $0 \le u, y \le 19$, while t denotes the current broadcast and t' denoted the next broadcast that the j item of the i group will be transmitted. The HMM structure is depicted in Fig. 2.

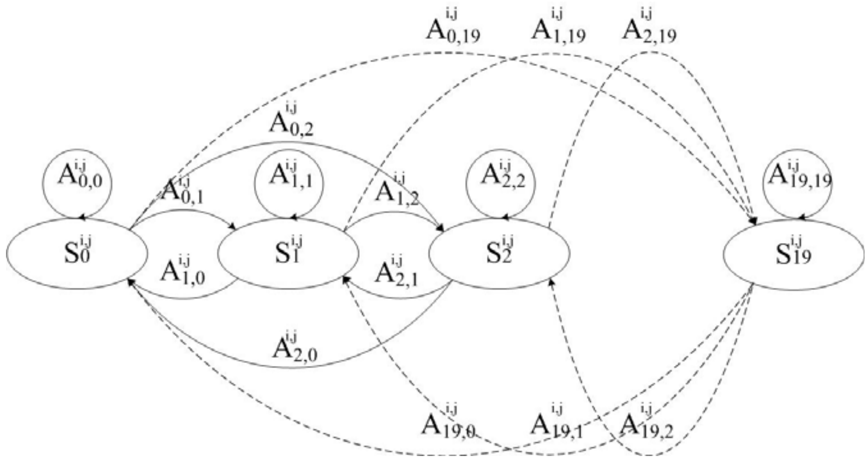

Figure 2. The Hidden Markov Model structure, defined by 20 individual states.

For example, transition probability $A_{4,1}^{3,2}$ refers to the transition from state $S_4^{3,2}$ to $S_1^{3,2}$. Practically, this transition means that item 2 of group 3 has accepted a number of feedbacks equal to 40-50% of the P/N theoretical population of group 3 after the t^{th} broadcast, while the same item has accepted a number of feedbacks equal to 10-20% of P/N after broadcast t'. The output of each predictor is the most probable transition, i.e., the transition with the greatest probability. In order to determine the most probable transition, a set of history vectors are considered for each distinct state to store the past acceptance rates with size equal to V entries:

$$H_z^{i,j}(h) = \{H_z^{i,j}(1), H_z^{i,j}(2), ..., H_z^{i,j}(V)\},$$

where $1 \le i \le N$, $1 \le j \le M$, $1 \le h \le V$, for each z, where $0 \le z \le 19$.

For example, entry $H_4^{3,2}(9) = 3$ means that item 2, of group 3, received feedbacks that indicate the transition from state $S_4^{3,2}$ to $S_3^{3,2}$, 9 entries before. History vectors operate like first in first out queues.

History vectors in conjunction with the actual clients' feedbacks define the predicted request for the following prediction phase. After broadcast t, let the actual acceptance rate of item j (group i) to indicate ph_t state. Also, consider that the actual acceptance rate of the same item after broadcast t' is $ph_{t'}$. Then it holds that the active current state of item j of group i after broadcast t is $S_{ph_t}^{i,j}$. The next step is the history vector update. History vectors regarding group i and item j are updated as follows:

$H_{ph_t}^{i,j}(h) = H_{ph_t}^{i,j}(h-1)$, for each g, where $2 \le h \le V$ and $H_{ph_t}^{i,j}(1) = ph_{t'}$

Then the corresponding transition probabilities are updated, according to history vectors:

$$A^{i,j}_{ph_t,y} = \frac{number\ of\ recorded\ entries\ in\ H^{i,j}_{ph_t}\ vector\ equal\ to\ y}{V},$$

for each y, where 0 ≤ y ≤ 19.

For example, consider that after broadcast 120th the actual acceptance rate for group 3, item 2 equals to the state 4 and after broadcast 125 th the same item accepts feedbacks that indicate state 1. Furthermore, let the size of the history vectors be equal to100 entries. History records are updated, hence $H_4^{3,2}(h) = H_4^{3,2}(h-1)$ for each h, where 2 ≤ h ≤ 100 and $H_4^{3,2}(1) = 1$. Next, the transition probabilities are changed.

Finally, the predictor modules choose the most probable transition state for each item and store it to vector F, as follows:

$F^{i,j}(x) = \arg\max_{0 \le x \le 19}[A^{i,j}_{ph_t,x}]$, for each i, j, where 1 ≤ i ≤ N, 1 ≤ j ≤ M.

At the beginning of the prediction phase vector F is filled with the most probable transition for each group, for each item. This fact means that vector F indicates the estimated feedback rate of each item. The next step is to apply a schedule algorithm, which is utilized in order to construct a schedule of the forthcoming broadcasts. The algorithm is defined as follows:

//begin the prediction phase

Calculate F vector

//a normalization takes place to schedule the 75% of the total broadcasts

Set $F^{i,j}(x) = F^{i,j}(x) \bullet \frac{R}{N}$, for each i, j, x, where 1 ≤ i ≤ N, 1 ≤ j ≤ M, 0 ≤ x ≤ 19

Summarize the F vector and set it to $Sum(F^{i,j})$

Set $F^{i,j}(x) = \frac{F^{i,j}(x)}{Sum(F^{i,j}(x))} \bullet \frac{3}{4} M$

//the selection of the forthcoming items takes place

Step1

Find $\max(x) = \arg\max_{0 \le x \le 19}[F^{i,j}(x)]$ and select x, as the next item to broadcast

Broadcast item x

Update the history

Reduce max(x) by one

If the broadcast reached the ¾ of the total broadcasts the algorithm ends, otherwise return to Step1

//end of schedule algorithm

3. The Simulation Environment

We consider a client population of P. Clients are grouped into N groups each one of which is located at a different location. In order to model groups with different group sizes, we compute the size of each group via the Zipf distribution.

$$c\left(\frac{1}{i}\right)^{\theta}, \textit{where } c = \frac{1}{\sum_k \left(\frac{1}{k}\right)^{\theta}}, k \in [1..N] \qquad (1)$$

where $1 \le i \le N$, θ is a parameter named access skew coefficient.

Any client belonging to group i is interested in the same subset D_i of server's data items. All items outside this subset have a zero demand probability at the clients of the group. Moreover, $D_i \ne D_l, \forall\, i, l \in [1..N], i \ne l$. Each one of the D_i subsets comprises of M items and the client's demand probability is computed via the above mentioned Zipf distribution. The item length, le, is considered to be the same for each item and equal to the unit, le = 1.

The simulation runs until each server broadcasts B items. The simulation results presented in this section are obtained with the following values to the parameters: P = 1000, B = 100000, and V = M.

Table 1 shows the different simulation environments. Fig. 3, Fig. 4, and Fig. 5 depict the mean response time versus the number of groups for N_1, N_2, and N_3 networks respectively.

Network	N	M	θ
N_1	$\in [5...20]$	5	0.9
N_2	10	$\in [5...20]$	0.9
N_3	20	5	$\in [0.0 \ldots 1.0]$

Table 1. The characteristics of the different simulation environments.

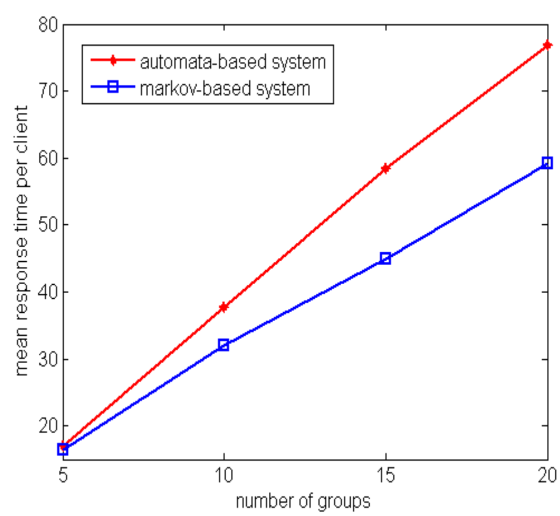

Fig.3. The mean response time versus number of groups for network N_1.

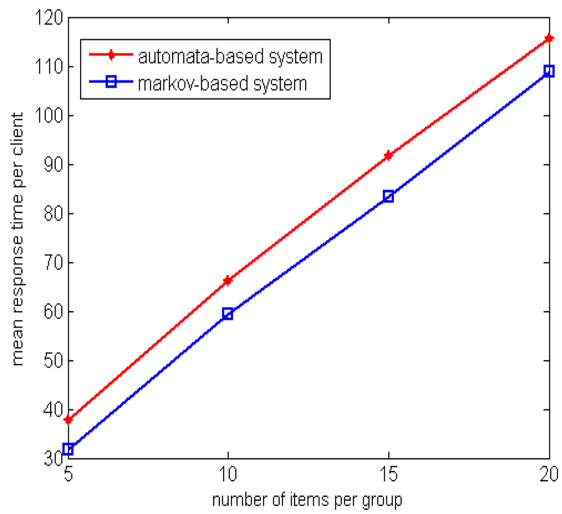

Fig.4. The mean response time versus number of data items for network N_2.

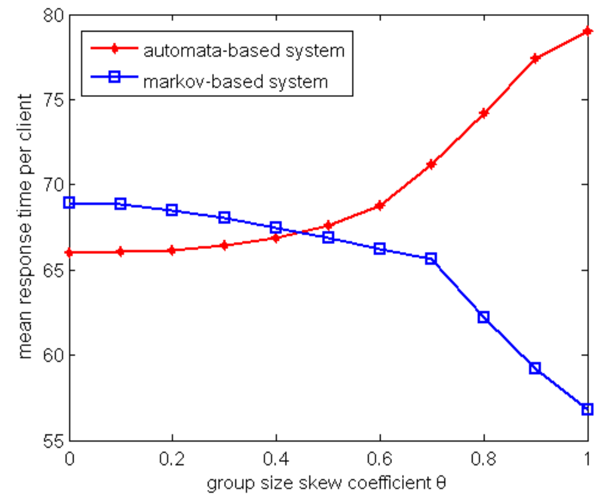

Fig.5. The mean response time versus group size skew coefficient θ for network N_3.

Conclusion

In this work a prediction-based scheme was presented for wireless push-based systems. The suggested model was designed to provide accurate adaptation regarding the requests of the connected clients. Probability states were designed in order to realize the successful rates of the server's broadcasts. Simulation results were presented that reveal the superior performance of the proposed scheme in environments with a priori unknown, dynamic client demands.

REFERENCES

[1] N.H.Vaidya and S.Hameed, "Scheduling Data Broadcast In Asymmetric Communication Environments", Wireless Networks, vol.2, no.3, pp.171-182, May 1999.
[2] P. Nicopolitidis, G. I. Papadimitriou and A. S. Pomportsis, " Using Learning Automata for Adaptive Push - Based Data Broadcasting in Asymmetric Wireless Environments ", IEEE Transactions on Vehicular Technology, vol.51, no.6, pp.1652-1660, November 2002.
[3] P. Nicopolitidis, G. I. Papadimitriou and A. S. Pomportsis, "Multiple Antenna Data Broadcasting for Environments with Locality of Demand", IEEE Transactions on Vehicular Technology, vol.56, no.5, pp.2807-2816, September 2007.
[4] M. A. L. Thathachar, P. S. Sastry, "NETWORKS OF LEARNING AUTOMATA, Techniques for Online Stochastic Optimization", Kluwer Academic Publishers, 2004.
[5] L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proeedings of IEEE, vol. 77, no. 2, pp. 257-289, 1989.
[6] E. Johnson, M. Mishra, and K. Sivalingam, "Scheduling in optical wdm networks using hidden markov chain based traffic prediction," J. Photonic Network Commun., vol. 3, no. 3, pp. 271–286, 2001.
[7] P. Sarigiannidis, G. Papadimitriou, and A. Pomportsis, "A high throughput scheduling technique, with idle timeslot elimination mechanism," IEEE/OSA Journal of Lightwave Technology, vol. 24, no. 12, pp. 4811–4827, 2006.