

# A New Approach to the Fairness of Adaptive Push Systems

V. Kakali, G. I. Papadimitriou, *Senior Member, IEEE*  
P. Nicolitidis, and A. S. Pomportsis

Department of Informatics, Aristotle University of Thessaloniki  
Box 888, 54124 Thessaloniki, Greece (e-mail:vicky198@csd.auth.gr)

**Abstract**– Many push-based systems are characterised by a priori unknown client demands and groups of clients that are located at the same region and have similar demands. However, the main disadvantage of the approaches that deal with such systems is the lack of fairness to the system, as groups with few members have much lower performance than groups with many clients. In other words, the performance per group depends directly on the number of clients per group. In this paper, we propose a fair push-based system suitable for applications with a priori unknown client demands and locality of demand, where the performance of each client can be independent of the total number of clients that are located in the same region without affecting the overall performance of the system.

**Index terms**– Data broadcasting, fairness, group size, learning automata, locality of demand.

## I. INTRODUCTION

Data broadcasting has emerged as an efficient means for the dissemination of information over asymmetric wireless networks. Traffic and weather information are some examples of data broadcasting applications. In such applications, broadcasting is an efficient solution, as the broadcast of a single information item will likely satisfy a large number of client requests. Push systems ( e.g. [1] ) are an approach for data broadcasting that provides scalability and client hardware simplicity. However, push systems, as the one described in [1], are unable to operate efficiently in environments with dynamic client demands. For this reason there has been proposed in [2] a push-based system that can be adapted to dynamic client demands using a Learning Automaton at the broadcast server. After each item broadcast, every satisfied client sends to the server a feedback pulse. The Automaton uses the aggregate feedback pulse in order to modify and adapt server’s broadcasts according to client demands. This adaptive push-based system has also been applied to push-based systems with locality of

demand [3]. Locality of demand means that clients are grouped into groups, each one located at different region and members of each group have similar demands for information items, different from the demands of clients at other groups. For example, clients located around an airport have different information demands from the clients around a theatre.

Fairness is an important issue in data broadcasting and has many different aspects. The approach in [4], concerns a pull system, which provide fairness in the case of items of different size at the broadcast server, while [5] concerns a hybrid system that aims at the fairness between the access time of the “pushed” and the access time of the “pulled” items.

The system, though, described in [2] does not ensure fairness among clients, as the response time of a client as well as the number of server’s broadcasts that concern this client, are affected by the total number of clients that are located in the same place with him.

The aspect of fairness that is considered in this paper, concerns the lack of fairness that is observed due to the unequal size of the clients’ groups. To our knowledge, this is the first approach that deals with this aspect of fairness.

In this paper we propose an adaptive push-based system that achieves fairness in an environment with different group sizes without affecting the overall performance of the system. The remainder of the paper is organised as follows. Section II presents the significance of fairness in such a system and describes the proposal approach. Section III presents the simulation results that reveal the fairness of this system. Section IV concludes this paper.

## II. THE FAIR ADAPTIVE PUSH-SYSTEM

### A. The Significance of Fairness

To indicate the significance of fairness in such a system, we present the situation below. We consider that group A has much fewer members than group B and that the server has broadcast a data item that is demanded by group A. Due to the few members of group A, the server will receive a small aggregate

feedback pulse. Thus, the server will consider that this item is unpopular at the system and will not broadcast it in the near future, even if this item has satisfied the majority of clients at group A. On the contrary, the large aggregate feedback pulse from the large group B indicates to the server to broadcast much frequently data items that concern group B. In this way, the percentage of the server's broadcasts that concern group A is much less than the percentage of the server's broadcasts that concern group B. Thus, the mean response time of each client at group A is much higher (much lower performance) than the mean response time of each client at group B. The lack of fairness is obvious, as clients that have the "bad luck" to belong to a small group are served rarely, while clients of large groups are served at regular times.

### B. The Broadcasting Algorithm and the Probability Updating Scheme

A Learning Automaton [6] is an automaton that improves its performance by interacting with the random environment with unknown and variable characteristics in which it operates. In the area of data networking Learning Automata have been applied to several problems, including the design of self-adaptive MAC protocols for wired and wireless platforms [7].

The broadcast server is equipped with a Learning Automaton, that contains the server's estimate  $p'_i$  of the demand probability

$p_i$  for each data item  $i$  that the server broadcasts. Obviously,  $\sum_{i=1}^M p'_i(n) = 1$  where  $M$  is the number of the data items. For

each broadcast, the server selects to transmit the item  $i$  that maximizes the cost function

$$CF(i) = (T - R(i))^2 \frac{p'_i}{l_i} (1 + E(l_i)) / (1 - E(l_i)), \quad 1 \leq i \leq M \quad (1)$$

where  $T$  is the current time,  $l_i$  is the length of item  $i$ ,  $E(l_i)$  is the probability that an item of length  $l_i$  is received with unrecoverable error, and  $R(i)$  is the time when item  $i$  was last broadcast.  $R(i)$  is initialized to  $-1$  and if the maximum value of  $CF(i)$  is given by more than one items, the algorithm selects one of them arbitrarily. Upon the broadcast of item  $i$  at time  $T$ ,  $R(i)$  is changed so that  $R(i) = T$ . After the transmission of item  $i$ , server waits for acknowledgement from every client that was waiting item  $i$ . In order to achieve fairness to the system, clients of each group transmit pulses with different power, as will be described later on.

After receiving the feedback pulses for the broadcast item  $i$ , server updates the demand probability vector  $p'$ . Vector  $p'$  is

updated in the following way: when there are no satisfied clients, vector  $p'$  does not change. However, when there are satisfied clients the probability of item  $i$  is increased while the probability of all the rest items is decreased.

$$\begin{aligned} p'_j(k+1) &= p'_j(k) - L(1-b(k)) * (p'_j(k) - a), \quad \forall j \neq i \\ p'_i(k+1) &= p'_i(k) + L(1-b(k)) * \sum_{i \neq j} (p'_j(k) - a) \end{aligned} \quad (2)$$

where  $L, a \in (0,1)$  and  $b(k)$  is the normalized sum of the received feedback pulses after the server's  $k^{th}$  transmission.

For the next broadcast, the server chooses which item to transmit by using the updated vector  $p'$ .

### C. The Proposed Feedback Mechanism Scheme to provide Fairness

In order to provide such a system with fairness, the broadcasts must become independent of the groups' size. In order to achieve it, this paper proposes the use of feedback pulses with different power for clients in different groups according to the equation below:

$$F(1) * Size(1) = F(2) * Size(2) = \dots = F(m) * Size(m) \quad (3)$$

where  $m$  is the number of groups,  $Size(1), Size(2), \dots, Size(m)$  are the number of clients at each group and  $F(1), F(2), \dots, F(m)$  is the feedback pulse power of clients in each group measured at the server's antenna. Thus, the feedback that server receives, is independent of the group size and depends only on the item's popularity inside the respective group.

Hence, there must be a mechanism that will once inform clients about the power of their feedback pulse. This mechanism works as follows. We consider clients equipped with GPS receivers. The server, which has a priori knowledge of the geographic locations (co-ordinates) where his services are offered, sends one control packet for each one of these locations, asking of the clients that belong to this location to send back a pulse. The client's GPS receiver detects its co-ordinates and the client sends its pulse as an answer to the control packet that refers to its area. As nowadays many PDAs are typically equipped with GPS receivers, there is no need for complex client equipment. Via the above approach, the server obtains an estimation of the size of each group. Then, server, taking into account the size of each group and equation (3), calculates  $F(1), F(2), \dots, F(m)$  and send a control packet so as

to inform members of each group of the power of their feedback pulses.

### III. PERFORMANCE EVALUATION

Using simulation, we compared the proposed fair push system against the system of [2] with the single power feedback pulse in an environment that is characterized by (a) a-priori unknown client demands to the server, (b) locality of demand, (c) unequal group sizes. The simulation environment is described below.

#### A. Server Model

We consider a server that contains a database of size  $M$  in items. The item length,  $l$ , is considered to be the same for each item and equal to the unit,  $l = 1$ . The server is initially unaware of the demand of each item, so initially every item has a probability estimate  $p'_i$  of  $1/M$ .

#### B. Client Model

We consider a client population of  $NumCl$ . Clients are grouped into  $G$  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. Thus the ratio of the number of clients in group  $g$ ,  $1 \leq g \leq G$ , to the number of clients in the entire system is:

$$c \left( \frac{1}{g} \right)^\theta, \quad \text{where } c = 1 / \sum_k \left( \frac{1}{k} \right)^\theta, \quad k \in [1..G] \quad (4)$$

where  $\theta$  is a parameter named access skew coefficient. For  $\theta = 0$ , the Zipf distribution reduces to a uniform distribution of group sizes. For large values of  $\theta$ , the Zipf distribution produces increasingly skewed patterns.

Any client belonging to group  $g$  is interested in the same subset  $S_g$  of server's data items. All items outside this subset have a zero demand probability at the clients of the group. Moreover,  $S_i \neq S_j, \forall i, j \in [1..G], i \neq j$ , which means that there do not exist common demands between any two clients belonging to different groups.

Assume that such a subset comprises  $Num$  items. The demand probability  $p_i$  for each item in place  $i$  in that subset, is computed via the Zipf distribution. This distribution has also been used to other papers that deal with data broadcasting [1], [2], [3].

$$p_i = c \left( \frac{1}{i} \right)^{\theta_1}, \quad \text{where } c = 1 / \sum_k \left( \frac{1}{k} \right)^{\theta_1}, \quad k \in [1..Num] \quad (5)$$

Finally, we consider that  $NumS_g$ ,  $1 \leq g \leq G$ , is the number of items that constitutes the  $S_g$ . The number of items per subset is (a) the same for each subset  $S_g$  and equal to  $M/G$  or (b) random for each subset with  $NumS_1 + NumS_2 + \dots + NumS_G = M$ .

#### C. The Simulation Environment

We performed our experiments with an event-driven simulator coded in Matlab. The simulator models the  $NumCl$  clients, the broadcast server and the server-client links as separate entities. The server has no a-priori knowledge of item demands, so initially all items have the same demand probability. The broadcast server broadcasts according to the automaton's page probabilities  $p'_i$  and the cost function (1), and updates the  $p'_i$  estimations after receiving the feedback pulses. The broadcasts are subject to reception errors, with unrecoverable errors per instance of an item occurring according to a Poisson process with rate  $\lambda$ , also used in [2], [3]. Thus,  $E = 1 - e^{-\lambda}$  is the probability that a page is received with an unrecoverable error.

The simulation runs until the server broadcasts  $N$  items. The overhead due to the duration of the feedback pulses and the signal propagation delay is defined via the parameter  $Ouh$ .

#### D. Simulation Results

The simulation results presented in this section are obtained with the following values to the parameters:  $NumCl = 5000$ ,  $N = 200000$ ,  $Ouh = 10^{-3}$ ,  $G = 5$ ,  $M = 100$ ,  $\lambda = 0.1$ ,  $L = 0.15$ , and  $\alpha = 10^{-4}$ ,  $\theta_1 = 0.5$ . The values of  $\lambda, L, \alpha$  are the same as the ones used in [2]. The measurement of the response time is considered to be the duration of the broadcast of one item.

The figures below depict the results of our simulations in five different environments:

1. Network  $N_1$ :  $\theta = 0.7$

$$NumS_1 = NumS_2 = NumS_3 = NumS_4 = NumS_5 = 20$$

2. Network  $N_2$ :  $\theta = 0.7$

$$NumS_1 = 17, NumS_2 = 12, NumS_3 = 35, NumS_4 = 14, NumS_5 = 22$$

3. Network  $N_3$ :  $\theta = 0.2$

$$NumS_1 = 17, NumS_2 = 12, NumS_3 = 35, NumS_4 = 14, NumS_5 = 22$$

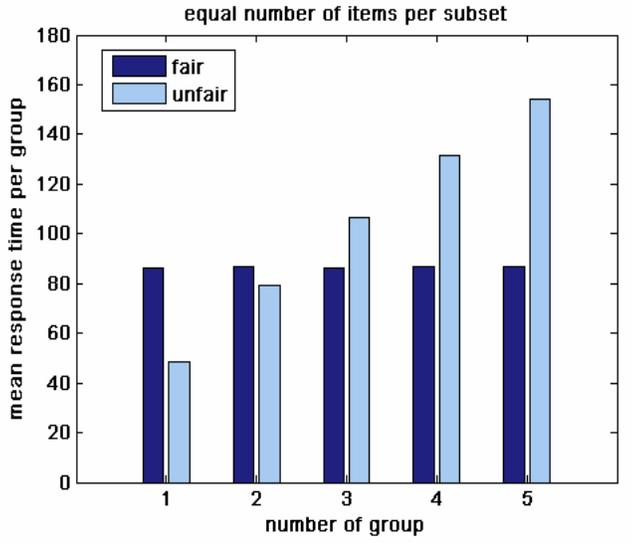


Fig.1. Mean response time per group in network  $N_1$  .

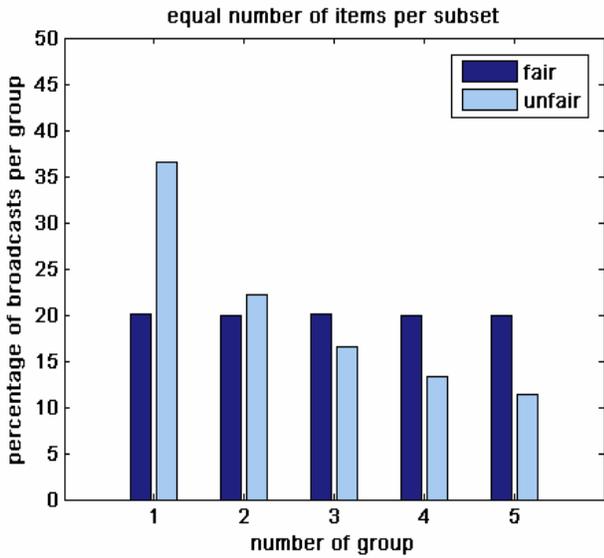


Fig.2. Percentage of broadcasts per group in network  $N_1$  .

4. Network  $N_4$  :  $\theta=0.4$

$$NumS_1 = 17, NumS_2 = 12, NumS_3 = 35, NumS_4 = 14, NumS_5 = 22$$

5. Network  $N_5$  :  $\theta=0.9$

$$NumS_1 = 17, NumS_2 = 12, NumS_3 = 35, NumS_4 = 14, NumS_5 = 22$$

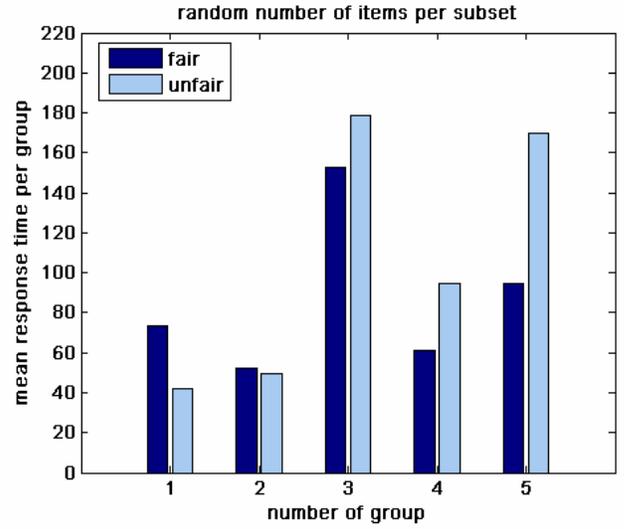


Fig.3. Mean response time per group in network  $N_2$  .

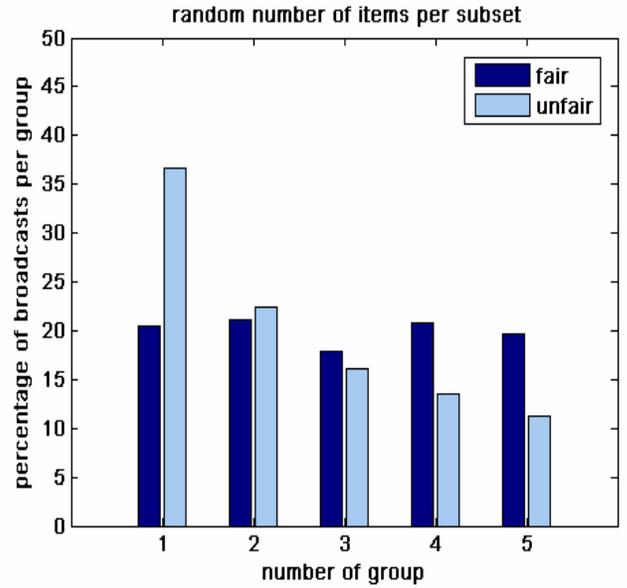
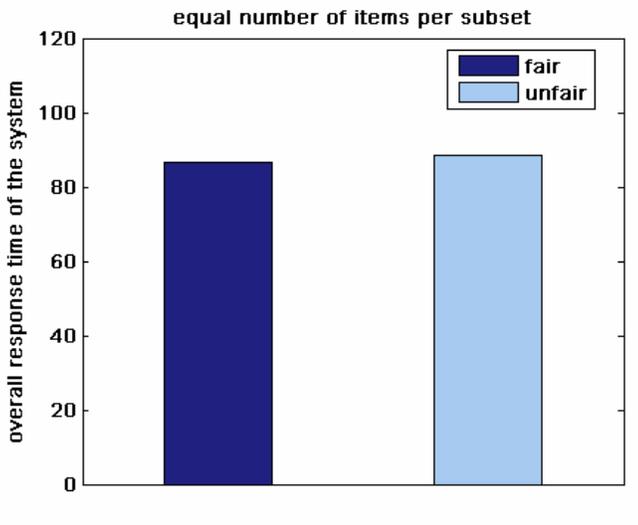


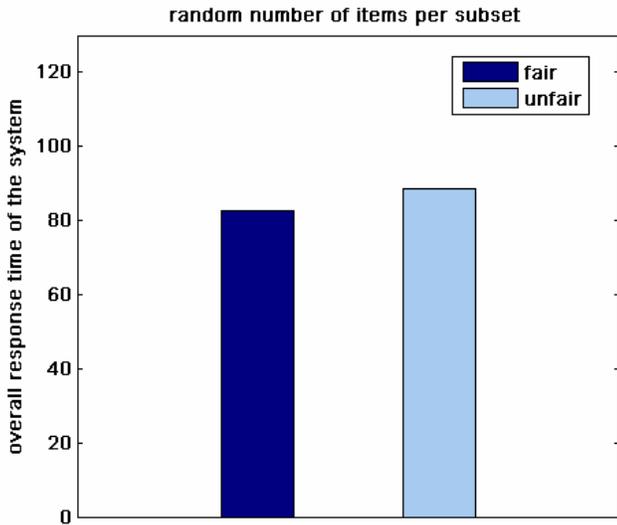
Fig.4. Percentage of broadcasts per group in network  $N_2$

Using different values of  $\theta$ , the number of clients that belong to each group change each time, according to (4).

- For  $\theta = 0.2$ ,  
 $Size(1)=1203, Size(2)=1047, Size(3)=966, Size(4)=912, Size(5)=872$
- For  $\theta = 0.4$ ,  
 $Size(1)=1428, Size(2)=1082, Size(3)=920, Size(4)=820, Size(5)=750$



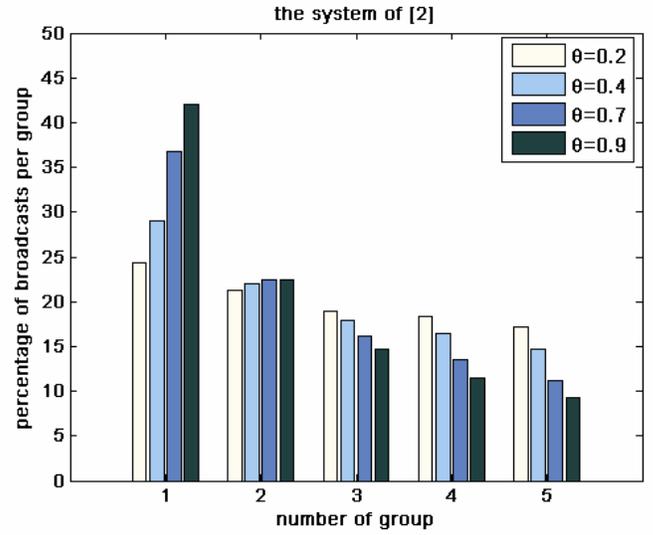
**Fig.5.** Overall response time of the system in network  $N_1$ .



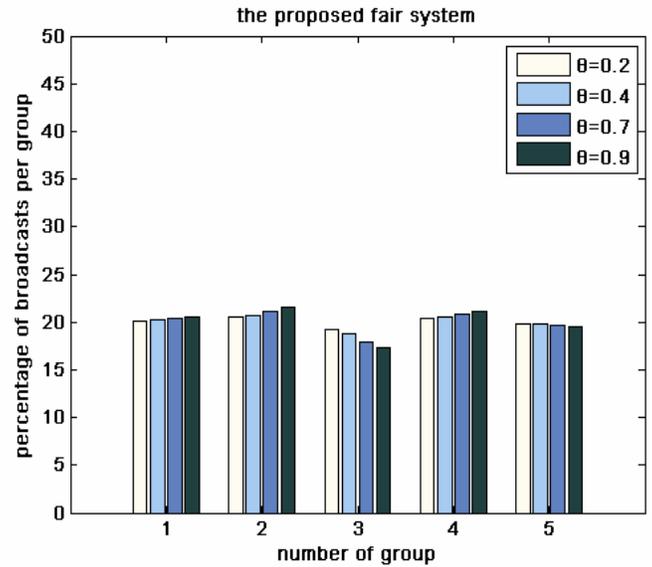
**Fig.6.** Overall response time of the system in network  $N_2$ .

- For  $\theta = 0.7$ ,  
 $Size(1)=1797, Size(2)=1106, Size(3)=833, Size(4)=681, Size(5)=583$
- For  $\theta = 0.9$ ,  
 $Size(1)=2058, Size(2)=1103, Size(3)=766, Size(4)=591, Size(5)=483$

Figures 1, 2 display the mean response time of each group and the percentage of the server's broadcasts for each group upon the total number of server's broadcasts respectively, for network  $N_1$ .



**Fig.7.** Percentage of broadcasts per group, at the push system that does not provide fairness, for four different networks  $N_2, N_3, N_4, N_5$ .



**Fig.8.** Percentage of broadcasts per group, at the proposed fair push system, for four different networks  $N_2, N_3, N_4, N_5$ .

In Figures 3, 4 we present the mean response time of each group upon the total number of server's broadcasts respectively, for network  $N_2$ . Figures 5, 6 display the overall response time of the system, for networks  $N_1$  and  $N_2$  respectively. In each of the Figures described above, we compare the proposed fair push system to the system of [2].

In Figures 7, 8 we depict the percentage of the server's broadcasts for each group, for networks  $N_2, N_3, N_4, N_5$ , at the system of [2] and the proposed fair push system, respectively.

Observing the Figures, we conclude the followings :

- The simulation results in Figures 1, 2 confirm the fact that the proposed system provides fairness. Using equal number of items per subset  $S_g$ , to avoid any dependences of our results

On the number of items per subset  $S_g$ , we can observe that, in the proposed fair push system the mean response time as well as the percentage of broadcasts per group are equal and independent of the number of clients per group, whereas, in the system of [2] these values are not equal.

- Figures 3, 4 show the fairness that this approach achieves in a more realistic environment where the numbers of items per subset  $S_g$  are not equal with each other. In Figure 3, the values which are very low at the system of [2] increase, whereas the ones that are high decrease. Thus, all mean response time values approach to each other and system becomes more fair. It is noticeable that at the fair push system, the percentage of broadcasts per group is almost equal for each group (Figure 4), which does not hold for the system of [2], in which the percentage of broadcasts is not equal for each group and depends on the number of clients of each group.

- At the fair push system, the number of server's broadcasts is almost equally split among the groups of clients, either we use equal number of items per subset (Figure 2) or random number of items per subset (Figure 4).

- As far as the values of the mean response time at the fair push system in network  $N_2$  are concerned (Figure 3), we observe that these values are not equal with each other as they were in  $N_1$  (Figure 1). This is predictable as the clients of a group that demand pages from a subset of small range (e.g. Group 2,  $NumS_2 = 12$ ) have smaller response time than the clients of a group that demand pages from a subset of great rang (e.g. Group 3,  $NumS_3 = 35$ ). This situation does not appear in  $N_1$  because all the subsets of items are equal with each other.

- Figures 5 and 6 confirm that this new approach does not affect the overall response time of the system.

- In the system of [2], the percentage of the broadcasts is not equally split among the groups. In particular, the percentage is high at the large groups at the expense of the percentage of the small ones. It is also important to mention that, as the values of  $\theta$  increase, the system becomes even more unfair for the groups with few members, as their percentage of broadcasts is decreased dramatically (Figure 7).

On the contrast, at the proposed fair push system, for each value of  $\theta$ , the percentage of broadcasts per group is almost the same. It is noticeable the fact that the broadcasts' percentage of each group remains almost the same, independent of the different values of  $\theta$  (Figure 8).

Thus, the system is completely fair as the broadcasts of the server are equally split among the groups, independent of the clients' distribution at the groups.

#### IV. CONCLUSION

This paper proposes an adaptive push system that deals with environments whose clients belong to groups of unequal sizes. By using different power feedback pulses, the server achieves to provide fairness to the response time and the percentage of broadcasts among the groups with different numbers of clients. This is achieved without decreasing the overall performance of the system.

#### 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, "Exploiting Locality of Demand to Improve the Performance of Wireless Data Broadcasting", *IEEE Transactions on Vehicular Technology*, vol.55, no.4, pp.1347-1361, July 2006.
- [4] Xiao Wu and Victor C.S.Lee, "Preemptive Maximum Stretch Optimization Scheduling for Wireless On-Demand Data Broadcast", *IEEE Proceedings of the International Database Engineering and Applications Symposium*, 2004.
- [5] Chih-Lin Hu and Ming-Syan Chen, "Adaptive Balanced Hydric Data Delivery for Multi-Channel Data Broadcast", *IEEE* 2002.
- [6] K.Najim and A.S.Poznyak, "Learning Automata: Theory and Applications", Pergamon, New York, 1994.
- [7] G.I.Papadimitriou and A.S.Pomportsis, "Learning Automata- Based TDMA protocols for Broadcast Communication Systems with Bursty Traffic", *IEEE Communication Letters*, vol.4, no.3, pp. 107-109, March 2000.