Exploiting Locality of Demand to Improve the Performance of Wireless Data Broadcasting

P. Nicopolitidis, G. I. Papadimitriou, *Senior Member, IEEE*
and A. S. Pomportsis, *Member, IEEE*

Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece.

Abstract— **With the increasing popularity of wireless networks and mobile computing, data broadcasting has emerged as an efficient way of delivering data to mobile clients having a high degree of commonality in their demand patterns. In many applications clients are grouped into several groups, each one located in a different region, with the members of each group having similar demands. In fixed bit rate wireless broadcast systems, transmission power is set at such a level that guarantees the necessary level of received energy per bit for all clients in the service area, so that they can operate under a predefined BER level. However, as in wireless cellular environments the path loss of wireless signals is typically inverse to the fourth power of the transmitter/receiver distance, there exists an increasing redundancy in the level of received energy per bit for decreasing distances from the server's antenna. This paper proposes a mechanism that exploits locality of demand in order to increase the performance of wireless data dissemination systems. Specifically, it trades the received energy per bit redundancy at distances smaller than the radius of the service area for an increased bit rate for transmission of items demanded by clients at such distances. This results in an increased transmission speed for many items. The bit rate for an item transmission is dynamically determined from the distance between the server's antenna to the group of clients that demand this item. Knowledge of clients' positions is conveyed to the server via a simple feedback from the clients. Simulation results are presented that reveal significant performance improvement over fixed bit rate broadcasting in environments characterized by locality of client demands.**

Keywords— **Adaptive Data Broadcasting, Asymmetric Wireless Environments, Learning Automata, Locality of Demand, Variable Bit Rate.**

Correspondence Address:
Prof. Georgios I. Papadimitriou
Department of Informatics
Aristotle University, Box 888
54124 Thessaloniki, Greece.
e-mail: gp@csd.auth.gr

I. Introduction

Data broadcasting has emerged as an efficient means for the dissemination of information over asymmetric wireless networks [1]. Examples of data broadcasting applications are traffic information, weather information and news distribution systems. In such applications, client needs for data items are usually overlapping. Consequently, broadcasting stands to be an efficient solution, as the broadcast of a single information item will likely satisfy a (possibly large) number of client requests. Moreover, in many applications, such as weather information and news distribution, the locations of clients determine their demands.

Communications asymmetry is due to a number of facts, such as asymmetry in equipment (e.g. lack of client transmission capability, client power limitations), asymmetry in the network system (e.g. small uplink/downlink bandwidth ratio) and application asymmetry (e.g. traffic pattern of client-server applications).

The goal pursued in most of the proposed data delivery approaches is twofold: a) determination of an efficient sequence (broadcast program) for the transmission of the server's data items in a way that the average response time (overall mean access time among the clients) is minimized and b) management and operation of client local memory (cache) so that a client's performance degradation is reduced when mismatches occur between the client's demand pattern and the server's program. This paper focuses on minimization of response time under dynamic and location dependent client demand patterns.

So far, three major approaches have appeared for the server's broadcast program:

- In the pull-based approach (e.g. [2]), the server broadcasts information after explicit requests made by the mobile clients via the uplink channel. This approach is able to adapt to dynamic client demand patterns, however it is inefficient from the point of view of scalability. This is because when the client population becomes too large, the client requests will either collide with each other or saturate the server.

- In the push-based approach (e.g. [3, 4, 5]), the server is assumed to have an a-priori estimate of the demand per-information item and makes item broadcasts according to these estimates. Push systems provide high scalability and client hardware simplicity since clients do not need to include data packet transmission capability. However push systems are unable to operate efficiently in environments with dynamic demand patterns. Nevertheless, with minimal changes to client and server hardware, [6] extends the applicability of the push approach to environments characterized by a-priori unknown and dynamic client demands and presents results that reveal efficient operation in such environments.

- Hybrid approaches (e.g. [14]) try to combine the benefits of the pure-push and pure-pull approaches. However they need to carefully strike a balance between push and pull and manage a number of additional issues (determination and dynamic selection of bandwidth available for push and pull, selection of items to be pushed and those to be pulled, etc).

Information dissemination applications can be characterized by locality of client demands. A possible example of this case could be the case of a museum possessing the necessary infrastructure in order to deliver to the users information regarding the exhibits. Most museums contain several sectors with each sector containing exhibits of a different type (e.g. Egyptian, Greek, etc.). It would be desirable for visitors within a sector to be aided in their tour by receiving information regarding the contents of the sector in their native language. Supposing that the information server broadcasts such information at several languages it can be seen that locality of demand indeed exists, as groups of visitors (which are of the same nationality) tend to be at the same place and many groups are usually present inside the museum at the same time.

In a wireless data dissemination system, the transmission power of the broadcast server determines the service area. Thus, if one wants to provide data dissemination services in an area of radius R, transmission power must be set at such a level that guarantees the necessary energy per bit to noise density per Hz (E_b/N_0) ratio for clients located at the border of the service area. However, in wireless cellular environments the path loss of wireless signals at a distance d is a $1/d^n$ type loss with a typical $n \geq 4$ [15]. This fact creates an increasing redundancy in the E_b/N_0 figure for clients at distances $d < R$ from the antenna.

This paper proposes a mechanism that exploits locality of demand in order to increase the performance of wireless data dissemination systems. Locality of demand means that clients are grouped into groups, each one located at a different place and members of each group have similar demands for information items, different from the demands of clients at other groups. The proposed approach can trade the E_b/N_0 redundancy at clients in groups at distances $d < R$ for an increased bit rate for the broadcasts of the items demanded by these groups. Knowledge of client positions is conveyed to the server via a simple feedback pulse from the clients, a mechanism that was used in [6] in order to provide adaptivity to dynamic client demands. Thus, the proposed approach is presented in the context of the adaptive wireless push system of [6]. It is worth mentioning here that another alternative to transmission bit rate variation could be the use of adaptive modulation. Using adaptive modulation the E_b/N_0 redundancy at client groups close to the antenna could allow a more efficient modulation technique that would carry more bits per symbol than the modulation scheme used for client groups further away. However, changes in the modulation technique would result in discrete changes in transmission speed. On the other hand the bit rate variation will allow for a smooth change in transmission speed that can better match the different distances of various groups of clients from the antenna.

The remainder of this paper is organized as follows: Section II describes research related to the proposed approach. A brief introduction to Learning Automata, which are used in the adaptive wireless push system of [6] on which the proposed method builds, is made in Section III. Section III then presents the proposed variable bit rate adaptive wireless push system. Simulation results, which reveal the performance superiority of the proposed approach to that of the fixed rate adaptive wireless push scheme of [6] in environments with locality of demands, are presented in Section IV. Finally, Section V concludes the paper and highlights our future research in the area.

II. Related research

A. Push systems

Some of the early work relevant to data broadcasting used the flat approach [16], which schedules all items with the same frequency. However, in order to minimize mean access time, research showed that schedules must be periodic [17] and the variance of spacing between consecutive instances of the same item must be reduced [18].

A..1 Broadcast Disks

A method that satisfied both the above constraints was the Broadcast Disks model [3]. It proposed a way of superposition of multiple disks spinning at different frequencies on a single broadcast channel. The most popular data are placed on the faster disks and as a result periodic schedules are produced, with the most popular data being broadcast more frequently. This work also proposed some cache management techniques aiming to reduce performance degradation of those clients with demands largely deviating from the overall demands of the client population. It also proposed prefetching data items in the client's cache to accommodate future client needs. This work was augmented later by dealing with the impact of changes at the values of the data items between successive server broadcasts of the same items [19] and the addition of a backchannel to allow clients to send requests to the server [20].

A drawback of Broadcast Disks is the fact that it is constrained to fixed sized data items and does not present a way of determining neither the optimal number of disks to use nor their relative frequencies. Those numbers are selected empirically and as a result, the server may not broadcast data items with optimal frequencies, even in cases of static client demands. Furthermore, the rigid enforcement of the constraint for minimization of the variance of spacing between consecutive instances of the same item leads to schedules where instances of the same item are equally spaced. This fact can lead to schedules that possibly include empty and thus unused periods (holes). Finally, the Broadcast Disks approach is not adaptive to dynamic client demands, since it is based on the server's a-priori knowledge of static client demands, resulting in pre-determined broadcast schedules.

An interesting paper that builds on the method of Broadcast Disks is [21]. It tries to satisfy client requests in a low time while at the same time achieving a low energy consumption by the client. The contribution of this approach is threefold: a) determination of a method to assign the data items to be broadcast to the various disks so that overall mean access time is reduced, b) determination of the number of disks to use, c) integration of indexing in order to provide energy efficiency.

A..2 The Vaidya-Hameed method

Push-based systems are also proposed in [4]. This method also produces periodic schedules as stated in [22]. According to it the construction of the broadcast when all users are tuned to the same channel is based on the following two arguments:

Argument 1: Broadcast schedules with minimum overall mean access time are produced when the intervals between successive instances of the same item are equal [18].

Argument 2: Under the assumption of equally-spaced instances of the same items, the minimum overall mean access time occurs when the server broadcasts an item i with the spacing between consecutive instances of i being proportional to the factor $\sqrt{\frac{l_i}{p_i}}$, where p_i is the demand probability for item i and l_i is the item's length.

The algorithm operates as follows: Assuming that T is the current time and $R(i)$ is the time when item i was last broadcast, the broadcast scheduler selects to broadcast item i having the largest value of the cost function $G(i) = (T - R(i))^2 \frac{p_i}{l_i}$. For items that haven't been previously broadcast, $R(i)$ is initialized to -1 and if the maximum value of $G(i)$ is given by more than one item, the algorithm selects one of them arbitrarily.

As stated by its authors, the method in [4] has the advantage of automatically using the optimal frequencies for item broadcasts in contrast to [3]. Furthermore, the constraint of equally-spaced instances of the same item is not rigidly enforced, a fact that leads to elimination of empty periods in the broadcast. Finally [4] works with items of different sizes too. This assumption is obviously more realistic compared to that of fixed-length items made in the Broadcast Disks approach. However, the main drawback of the method in [4] remains its lack of adaptivity and therefore its inefficiency in environments with dynamic client demands.

A..3 The Adaptive Push System

The method in [6] proposes a push-based system that is adaptive to dynamic client demands. The system uses a Learning Automaton at the broadcast server in order to provide adaptivity to [4] while maintaining its computational complexity. Using a simple feedback from the clients, the automaton continuously adapts to the overall client population demands in order to reflect the overall popularity of each data item. It is shown in [6] that contrary to the non-adaptive method of [4], the adaptive system provides superior performance in an environment where client demands change over time with the nature of these changes being unknown to the broadcast server. The operation of the adaptive system of [6] is described in the Section III due to the fact that the proposed method builds on top of [6].

B. Pull and hybrid systems

Pull systems have the advantage of being adaptable to dynamic client demands due to extraction of knowledge of these demands via client requests. A representative pull system is that of [2]. This method is shown via simulations to provide close performance to the optimal but not scalable Longest Wait First (LWF) algorithm which selects to transmit the item with the largest aggregate waiting time (the sum of waiting times for all pending requests for that item). A recently proposed pull system is that of [7]. This work also considers the effect on performance of access time, tuning time (the time a client must actively listen to the broadcast before receiving the desired item) and the cost of handling request failures. It proposes a self-adaptive scheduling algorithm which computes a cost function affected by the above three parameters for each data item and uses it for schedule construction. However, in the context of push systems the proposed approach is evaluated by making the assumption that either a) each information item is demanded by the same probability or b) the demand for each information item is known in advance.

In hybrid methods clients can use a backchannel to submit requests for information items to the server. A recent work on such systems is that of [8], which proposes a hybrid system which takes into account user retrials. As far as adaptivity to client demand is concerned, this is made via the actual requests made by the clients. The authors state that even in a hybrid system estimation of items' demands is not a trivial task. A reason that explains this is the fact that in the effort to serve more users via the push mode the system manages to increase the number of such users. Thus, fewer users submit item requests via the pull-mode, resulting to an increasing possibility for less accurate estimation of item demands in the near future.

Another hybrid system recently proposed is that of [9]. It proposes an estimation mechanism and tries to balance the push and pull access times. To estimate item demands, [9] exploits clients' impatience. Periodically, it deliberately generates item misses from the push mode in order to force clients to explicitly demand these items via requests. Thus, by counting requests for each item in the push mode, the server has a measure of item demands among the client population. A similar approach with that of [9] is taken by [10]. It estimates the demand for each item via incoming requests for items that will not appear in the broadcast in the near future and states that without requests for such broadcast missed item demand estimation is impossible.

In [11] a system for dynamic data broadcasting that combines different means of retrieval of items (cache, broadcast channels, pull requests) is proposed. The server is able to work under dynamic client demands which are not a-priori known to it. Every client uses a bit vector in which each bit corresponds to a data item in the broadcast program. At the beginning of each broadcast program the server broadcasts the complete schedule. When clients receive a new program they reset their bit vectors. A bit in the vector of a client is set if the corresponding item will appear in the broadcast. Thus, items that are retrieved either from the client's cache or via pull requests have no influence on the bit vector of the client. When a certain item is not found both in the cache and the broadcast program then a pull request is made for it by the client via the uplink channel and the bit vector of the client is piggybacked at this request. After the pull request the bit vector is reset. At the server, each data item is associated with a counter whose value is increased by one every time this item is either explicitly demanded via a pull request or a bit vector is received with the corresponding bit set. With this mechanism the broadcast server is able to estimate the access patterns of the client population and dynamically determine the broadcast program according to these patterns.

Another interesting approach is that of [12]. This work proposes a system where client requests do not target specific items but can rather be satisfied by items containing information sufficiently close to their demands. Each client maintains a bit vector with each bit in place w in the vector set by a client if an item with a high metric of similarity to the one demanded appears in the w-th position in the broadcast. At any given time, each client sends with a small probability to the server a feedback message that contains its vector. If a client is not satisfied by the broadcast program its feedback will be an explicit request. The frequency of sampling the clients for their demands is determined by statistical techniques. Two such techniques are proposed with one of them being superior as it leads to a smaller proportion of sampled clients.

Another work that targets adaptive data broadcasting is that of [13]. Clients with interests either in the same items with static demands or the same broadcast services form groups. In the beginning of each broadcast cycle each group is assigned a quota of time slots within the cycle's duration. Furthermore, when a group has used its quota, the server can dynamically allocate more slots to the group via loaning of slots from other groups during the broadcast cycle. This loaning mechanism takes place when some slots become available before a broadcast cycle ends. When loaning is not possible due to lack of remaining slots in the broadcast program the server can transmit excess items either via the pull way or via preempting the push mode in the next broadcast cycle. At the end of each cycle the feedback that is available at the server and concerns slot loaning is used in combination with group popularities to adapt the system to the current client demands for the next broadcast cycle.

C. Critique on adaptivity to client demands

The pull and hybrid systems described above share the requirement that clients are able to submit actual packet requests to the server via an uplink channel. In order for these approaches to work in a push environment item demands must be a-priori known ([7], [8]) since in push systems there does not exist the possibility of relaying actual packet requests from the clients to the server. Even in the case of pull and hybrid systems however, a large number of clients can a) congest the uplink channel due to the need to coordinate packet requests with few collisions, b) generate an excessive request arrival rate at the server and thus saturate it.

On the contrary, [6] and the proposed system which builds on it, do not relay actual requests to the server via the uplink channel; rather what the simple uplink channel carries is the sum of clients' feedback pulses. Thus:

- since there is no need to coordinate client responses (pulses) so as not to collide, a large number of clients does not limit the system's scalability by congesting the simple uplink channel,
- what the server receives is not a number of individual packet requests that it must examine and possibly serve; rather it receives the sum of clients' feedback pulses that is used to estimate the demands of the various items.

III. The Variable Bit Rate Adaptive Wireless Push System

A. Learning Automata

The aim of many intelligent systems is to be able to efficiently work in environments with unknown and varying characteristics. A solution to this problem is Learning Automata [23, 24, 25, 26], which are structures that can acquire knowledge regarding the behavior of the environment in which they operate.

A Learning Automaton is an automaton that improves its performance by interacting with the random environment in which it operates. The goal of a Learning Automaton is to find among a set of actions $a_1, a_2, ..., a_M$ the optimal one, such that the average penalty received by the environment is minimized. The operation of a Learning Automaton constitutes a sequence of repetitive cycles which eventually lead to the target of average penalty minimization. The automaton maintains $p_1(n), p_2(n), ..., p_M(n)$, which is a vector representing the probability of selecting action i at cycle n. Obviously, $\sum_{i=1}^{M} p_i(n) = 1$. For each cycle, the automaton chooses an action and receives the environmental response triggered by the selected action. Based on this response the automaton updates the probability vector $p(n)$ to $p(n+1)$and uses it determine the selection of the next action.

There exist different automata types according to the nature of the environmental response. If this takes only the values 0 and 1, indicating only reward or penalty respectively, the automaton is known as a P-model one. However, due to the fact that in many cases a P-model gives only a gross estimation of the environment, schemes where environmental response can be neither completely rewarding or penalizing have been devised. These kind of Learning Automata work with environmental responses which, after normalization, lie in the interval [0..1]. In a Q-model, the environmental response can have more than two, still finite however, possible values in the interval [0..1]. In an S-model environment, the environmental response can take continuous values in [0..1].

Learning Automata have been found to be useful in systems where incomplete knowledge regarding the environment in which those systems operate exists. In the area of data networking Learning Automata have been applied to several problems, including the design of self-adaptive MAC protocols, both for wired and wireless platforms, which efficiently operate in networks with dynamic workloads [27, 28, 29, 30]. Other applications of Learning Automata include queueing systems, task scheduling, image compression, pattern recognition and telephone-traffic routing.

B. The Learning Automaton-based Broadcast Server

In the variable bit rate adaptive wireless push system the server is equipped with an S-model Learning Automaton, which contains the server's estimate p'_i of the demand probability p_i for each data item i among the set of the items the server broadcasts. Clearly, $\sum_{i=1}^{M} p'_i = 1$, where M is the number of items in the server's database.

For each item broadcast, the server selects to transmit the item i that maximizes the cost function $G(i) = (T - R(i))^2 \frac{p'_i}{l_i}, \quad 1 \leq i \leq M$, where T is the current time, $R(i)$ the time when item i was last broadcast and l_i is the length of item i. As mentioned earlier, for items that haven't been previously broadcast, $R(i)$ is initialized to -1 and if the maximum value of $G(i)$ is given by more than one item, 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 broadcasting item i, the algorithm proceeds to select the next item to broadcast. At the adaptive system, after the transmission of item i, the broadcast server awaits for acknowledgment from every client that was waiting item i. Such clients acknowledge reception and each one transmits a feedback pulse. Acknowledging clients' pulses, which are of fixed length, are added at the server, which uses the aggregate received pulse to update the automaton. As far as the feedback pulses are concerned, we assume that the channel is symmetric, meaning that the path loss from the server to a certain client is the same with that from the same client to the server. Moreover, the server and all the clients clients have an agreement on what signal power should be received as a feedback. A feedback pulse of a client, used also in [6], is a simple pulse of very short duration that upon its arrival at the server signifies item reception at the client. The probability distribution vector p' maintained by the automaton estimates the demand probability p'_i (and thus the popularity) of each information item $i, 1 \leq i \leq M$. For the next broadcast, the server chooses which item to transmit by using the updated vector p'.

C. The probability updating scheme

When the transmission of an item i does not satisfy any waiting client, the probabilities of the items do not change. However, following a transmission that satisfies clients, the probability of item i is increased. A Liner Reward-Inaction (L_{R-I}) probability updating scheme [24] is employed after the transmission of each item. Thus, assuming item i is the server's k^{th} transmission, the following probability updating scheme is used:

$$p'_j(k+1) = p'_j(k) - L(1-b(k))(p'_j(k) - a), \forall j \neq i$$

$$\tag{1}$$

$$p'_i(k+1) = p'_i(k) + L(1-b(k)) \textstyle\sum_{i \neq j}(p'_j(k) - a)$$

It holds that $L, a \in (0,1)$ and $p'_i(k) \in (a,1)$ $\forall k$ and $\forall i \in [1..M]$, where M is the number of the information items. L is a parameter that governs the speed of the automaton convergence. The selection procedure for a value of L reflects the classic speed versus accuracy problem. The lower the value of L the more accurate the estimation made by the automaton, a fact however that comes at expense over convergence speed. If the probability estimate p'_i of an item i approaches zero, then $G(i)$ would be very close to zero as well. However, item i, even if unpopular, still needs to be seldomly transmitted since some clients may request it. Additionally, the dynamic nature of client demands might make this item popular in the future. Parameter a prevents the probabilities of non-popular items from taking values in the neighborhood of zero and thus increases the adaptivity of the automaton.

$b(k)$ is the system environmental response that is triggered after the server's k^{th} transmission. Essentially, it is the normalized sum of the received feedback pulses after the server's k^{th} transmission. Thus, there must exist a mechanism that enables the server to possess an estimate of the number of clients under its coverage so as to perform the normalization procedure on the sum of feedback pulses. This can be achieved by the broadcasting of a control packet that notifies all clients to respond with a pulse. The server will use this aggregate received pulse strength to estimate how many clients are within its coverage area and use this value to perform the normalization. The estimation process occurs at regular time intervals [6] after the broadcast of several hundreds of items, with its overhead being the broadcast of a unit-length item. The simulation results in [6] show that such an overhead is negligible due to the superior performance of the proposed approach. A value of $b(k)$ that equals 1 represents the case where no client acknowledgment is received. Consequently, the lower the value of $b(k)$, the more clients were satisfied by the server's k^{th} transmission.

However, the signal strength of each client's pulse at the server depends on its distance from the server's antenna. Moreover, it is of dynamic nature due to the mobility of the clients. Since the path loss is a $1/d^n$ type loss with a typical $n = 4$, the feedback pulse of clients located close to the broadcast server will be orders of magnitude stronger than the pulses of clients further away. In order to prevent those clients that are closer to the server's antenna from dominating the voting, we use a power control mechanism on the returning pulses. Thus, every information item will be broadcast including information regarding the signal power used for its transmission. Due to path loss, clients far from the base station will receive the item and measure a low signal power. Based on the received signal power of the item and the piggybacked information regarding the signal power at which the item was originally transmitted, the client will accordingly set the power of its feedback pulse. Thus, clients acknowledging the receipt of an item will measure the item's signal power and set the power of their feedback pulse to be the inverse of the ratio (power of the received item) / power of the item transmission). For example, assume that the server broadcasts all items with signal power 1. Upon reception by a client of a item with power k ($k \leq 1$), the client will set the power of its feedback pulse at $1/k$. Using this form of power control, the contribution of each client's feedback pulse at the server will be the same order of magnitude and will not depend on client-server distance. For the broadcast of each item, the received feedback signals at the server pass through an operational amplifier [31, 32] that integrates the energy received during a time interval t_p after the broadcast of the item. This time interval is equal to the sum of the feedback duration and the maximum round-trip propagation delay (the round-trip delay from the server to a client located at the border of the coverage area). From the total energy received during this interval the server can conclude about how many stations have transmitted a feedback pulse [32, 33]. Then, the operational amplifier is being reset in order to begin a new integration of duration t_p after the broadcast of the next item.

Using the re-enforcement scheme of Equation (1), the item probabilities estimated by the automaton converge to the actual demand probabilities for each information item. Via simulation, this convergence is shown in Figure 1. Overall client demands for the item are initially unknown to the server. It can also be seen that they are of a dynamic nature as well: At some time instant, the initial overall demand probability for the selected item (solid line) changes to a new one (dashed line). It can be seen that the higher the value of L, the more faster the convergence is at the expense however of estimation accuracy. Estimation accuracy for each value of L is computed in this Figure as the mean distance of the automaton estimates p_i' of the demand probability from the actual demand probability p_i. Its computation starts after the automaton estimates have converged to the actual demand probability and ends when the demand probability for the item changes and thus the automaton estimates begin to converge to the new overall demand probability. To this end we identified convergence as the point where the mean value of p_i' does not differ from p_i by more than 5%. Simulation results in [6] and [34] have demonstrated efficient operation in environments characterized by dynamic and a-priori unknown to the server, client demands.

As far as system complexity is concerned, it can be easily seen that the probability updating scheme of Equation (1) is of O(M) complexity and thus the system maintains the O(M) complexity of the non-adaptive system ([4]). Furthermore, due to the fact that a large number of clients does not overload the server or the uplink channel (since what is carried via the uplink channel is just the sum of the clients' feedback pulses), the system is scalable and maintains its efficiency regardless of the number of clients.

D. The Bit Rate Variation Mechanism

To the authors' knowledge, locality of demand has not been taken into account in related research so far; on the contrary, clients are assumed to be uniformly distributed inside the service area and generally make item requests using the same or similar patterns (e.g. [3, 4, 6]).

In certain cases however, clients are grouped into several groups located at different places with the clients of each group having similar demands, different from those of clients at other groups. In the following discussion all clients are assumed to be positioned outside the near field of the server's antenna.

In a typical data broadcasting application (and generally in wireless cellular systems), service area is an area of certain radius R inside which mobile clients are able to receive information items while experiencing a Bit Error Rate (BER) below or equal to a certain requirement value. The size of the service area depends on a number of parameters, such as the type of modulation that is used, the bit rate, the server's transmission power and noise density per Hz and is determined by a simple rule stating that its border is where the received energy per bit E_b divided by the noise density per Hz N_0 equals a certain constant A. The value of A is determined so that the E_b/N_0 ratio results in a BER below or equal to a set requirement. Thus at the border of the service area it stands that:

$$\frac{E_b}{N_0} = A \Rightarrow E_b = A' \tag{2}$$

where $A' = AN_0$.

Since $E_b = T_b S_R$, where S_R is the received power at distance R from the antenna and T_b is the bit duration, we can rewrite the above relation as:

$$T_b S_R = A' \tag{3}$$

Finally, since in wireless cellular environments the path loss of wireless signals at distance d is a $1/d^n$ type loss (with a typical value of $n \geq 4$), (3) can be expressed as:

$$R^{-n} T_b = A' \tag{4}$$

In fixed bit rate systems, clients inside the service area (at distance $d < R$) experience even lower BERs than those required due to smaller distance from the antenna. Thus, for such clients it holds that $E_b > A'$ and therefore:

$$d^{-n} T_b > A', \forall d < R. \tag{5}$$

Assume that there exists locality of demand, as defined earlier. Then we can exploit the above mentioned redundancy in the received BER by dynamically reducing the T_b parameter for each information item i so that it always holds that $d^{-n}T_b(d) = A'$, where d is the distance of the group of clients that access item i.

The proposed approach works as follows: Each information item comprises a header that contains information (e.g a sequence number) that uniquely identifies the item. All item headers are always broadcast with the default T_b value, while the T_b value for the main item payload can be altered by the server. After the transmission of item i, the server waits for acknowledgment pulses from all mobile clients that were satisfied by this transmission. Since we consider groups of clients having the same interests, acknowledgment pulses for a certain item will be from a group of collocated clients and therefore arrive together at the server. The server monitors the time elapsed from the broadcast of item i until these pulses are received and uses this information to calculate the distance d of the group of clients from the antenna.

The server will use this information to broadcast the payload of the next instance of item i via a bit duration of $T_b(d)$ that satisfies the requirement that:

$$d^{-n}T_b(d) = A', \Leftrightarrow T_b(d) = A'd^n \tag{6}$$

Change of the bit duration is not a problem for the mobile client, as it can be informed of this via piggybacking of the new bit duration in the item header, which is always broadcast with the default T_b value.

We now argue use of Equation 6 minimizes overall mean access time. As shown in [4], the minimum overall mean access time t is given by the following equation:

$$t = \frac{1}{2}\left(\sum_{i=1}^{M}\sqrt{p_i l_i}\right)^2 \tag{7}$$

where l_i and p_i are the length and the demand for item i respectively. However, in the variable rate system the length l_i of i demanded by a group at distance d_i depends on the bit time $T_b(d_i)$ for that item and is thus equal to $T_b(d_i)l_i$. Thus, Equation 7 now becomes:

$$t = \frac{1}{2}\left(\sum_{i=1}^{M}\sqrt{p_i l_i T_b(d_i)}\right)^2 = \frac{1}{2}\left(\sum_{i=1}^{M}\sqrt{p_i l_i A' d_i^n}\right)^2 \tag{8}$$

We have the following possibilities for selecting a distance d according to which we will set the bit time $T_b(d_i)$ for each item i demanded by a group at distance d_i from the antenna:

- $d < d_i$. However, in this case, since $d_i^{-n}T_b(d) = A'\frac{d^n}{d_i^n} < A'$ we have no reception of item i due to an extremely decreased bit time. Thus we cannot use a lower bit time (higher bit rate) than the one given by Equation 6 for $d = d_i$.

- $d \geq d_i$. In this case we have reception since $d_i^{-n}T_b \geq A'$. It can be seen from Equation 8 that by setting $d > d_i$ for any item i, we have $t(d) > t(d_i)$. Thus overall mean access time is minimized when $d = d_i$ for every item i, which is what the proposed bit rate variation system proposes via Equation 6.

As far as acknowledgment pulses are concerned, a client responds to the server via such a pulse if it demands item i and successfully receives i's header. We explain that this provides support for clients that may have broken away from the main group and are located further away from the antenna than the main group. Assume that such a client C, at a distance $d1$ receives only the header of i due to the fact that the main item payload has been transmitted with a bit rate determined by the location of the main group, which is closer to the antenna. In that case the server will receive more than one feedback pulses. The one which arrives last corresponds to C. In order to prevent C from starvation, the server will schedule the broadcast of the next instance of item i according to the feedback pulse of C (thus the client further away). This enables the client further away from the group to successfully receive item i when it is next broadcast. At the next broadcast of item i, C will successfully receive the item. However, this time C will not transmit a feedback pulse so as not to acknowledge twice reception of one instance of item i, a fact that would provide inaccurate information regarding demand for item i to the probability updating scheme.

To better understand the behavior of the system we present the following example. We assume a server with a database of two items of unit lengths, two groups of clients, A and B, with group A always accessing the first item while group B always accesses the second item. The radius of the service area is R. The distances of groups A and B from the antenna are $d_A = R/2$ and $d_B = R/3$ respectively. Two members of group A are away from the main group at distances $R/4$ and $3R/4$. Furthermore, the client at distance $3R/4$ (client C) makes new item requests with non unit probability. Finally, $T_b(R) = 1$, the path loss exponent n is 4 and initially all item payloads will be transmitted via $T_b = 1$. We illustrate the following five example steps of the algorithm:

Step 1: We assume that according to the selection procedure, the server decides to transmit item 2.

- Clients in group B receive item 2 and transmit their feedback pulses.
- From the time elapsed between the broadcast of item 2 and the reception of the response from group B, the server calculates the distance of group B from the antenna and schedules the next instance of item 2 to be broadcast via $T_b(R/3)$ which equals $\frac{T_b(R)}{3^4} = 1/81$.

Step 2: Next, the server broadcasts item 1.

- All clients of group A, except client C, demand and receive item 1 and transmit their feedback pulses.
- The server receives more than one groups of pulses and will schedule the next broadcast of item 1 to take place according to the pulse corresponding to the location further away from the antenna, thus via $T_b(R/2)$ which equals $1/16$.

Step 3: Next, we assume that the server broadcasts again item 1 via $T_b(R/2)$.

- All clients of group A, including client C, demand this item. Obviously due to the increased bit rate the item is received by all members of A except client C who only receives the header of item 1. All members of group A (including client C) acknowledge item 1.
- The server receives three groups of pulses and will schedule the next broadcast of item 1 to take place according to the location of the acknowledging client further away (client C), thus via $T_b(3R/4)$ which equals 0.31.

Step 4: Next, the server decides to transmit item 2 via $T_b(R/3)$ as calculated in step 1. For this broadcast, everything else will be the same with step 1.

Step 5: Next, we assume that the server broadcasts again item 1, this time via $T_b(3R/4)$ as stated in Step 3.

- All members of group A, including client C, demand and receive item 1.
- All members of group A, except for client C, transmit their feedback pulses. C does not transmit a feedback pulse due to the reason explained earlier. Thus, the next broadcast of item 1 will be made via $T_b(R/2)$.

IV. Performance Evaluation

In order to assess the performance increase of the proposed variable rate system, we used simulation to compare it to the fixed bit rate system of [6]. The comparison is made in an environment characterized by client demands that are:

- A-priori unknown to the server.
- Location dependent.

A. Server model

We consider a broadcast server having a database of size Dbs in items. The server is initially unaware of the demand for each item, so initially every item has a probability estimate p'_i of $1/Dbs$. In the fixed bit rate system, the server broadcasts all items with the same bit rate. In the variable rate system however, the server determines the bit rate to use for each item according to the proposed scheme. As far as the item lengths are considered, we compute them similar to [4]: item lengths vary from $L_0 = 1$ to $L_1 = 10$, however, two cases are considered: In the first one, the lengths of items from 1 to Dbs are random integers uniformly distributed in $[L_0..L_1]$. In the second one, the length l_i of a item i is calculated using the formula:

$$l_i = round\left(\left(\frac{L_1 - L_0}{M-1}\right)(i-1) + L_0\right), \quad 1 \leq i \leq Dbs \tag{9}$$

where $round(x)$ returns a rounded version of x. These distributions of item lengths are called ”random” and ”increasing” length distributions respectively, according to the terminology in [4]. Figure 2 shows the lengths of items from 1 to 300 produced by the increasing distribution.

B. Client model

We consider a client population of $ClNum$ clients that have no cache memory, an assumption also made in other similar research (e.g. [4] and [6]). Clients are grouped into G groups each one of which is located at a different distance from the antenna and outside the antenna's near field. Any client belonging to group $g, 1 \leq g \leq G$ is interested in the same subset Sec_g of the server's database. All items outside this subset have a zero demand probability at the clients of the group. Finally, $Sec_i \neq Sec_j, \forall i, j \in [1..G], i \neq j$, 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 according to the Zipf distribution, which is used in other papers dealing with data broadcasting as well ([3], [4], [5], [6]):

$$p_i = c\left(\frac{1}{i}\right)^{\theta}, \qquad where \quad c = 1/\sum_k \left(\frac{1}{k}\right)^{\theta}, \quad k \in [1..Num] \tag{10}$$

where θ is a parameter named access skew coefficient. For $\theta = 0$ the Zipf distribution reduces to a uniform distribution of demand for the items in a subset. For large values of θ, the Zipf distribution produces increasingly skewed demand patterns. The Zipf distribution can thus efficiently model applications that are characterized by a certain amount of commonality in client demands. Figure 3 shows the demand probabilities per item for different values of θ for a database comprising 300 items.

Placement of client groups takes place among LP different distance points outside the antenna's near field, with the maximum distance corresponding to the coverage radius of the system. Members of a group g are not located in the same place. Rather, they are uniformly placed inside a circular area with its center located at a distance Loc_g. The maximum distance between any two clients in this area is Gr_Size distance points. Moreover, to model clients that are completely outside the circular area of the main group we introduce the parameter Dev, which determines for each group the percentage of clients that are located outside the respective circular area. For every client, a coin toss, weighted by Dev, is made. If the outcome of the toss states that the client is to deviate from the location of its group then its position is changed to a new one selected in a uniform manner from the interval [1..LP].

C. The Simulation Environment

We performed our experiments with an event-driven simulator coded in C. The simulator models the $ClNum$ 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. We assume that the broadcast server's antenna is at the center of the circular cell and a path loss model of $1/d^n$. In order to model different group sizes, we calculated the size of each group g via the above mentioned Zipf distribution. Thus the ratio of the number of clients in group g to the number of clients in the entire system is: $c\left(\frac{1}{g}\right)^{\theta 1},\ where\ c = 1/\sum_k \left(\frac{1}{k}\right)^{\theta 1},\ k \in [1..G]$.

Upon completion of a item's broadcast, the following events take place:

1. Clients that demanded the item and received its header correctly respond with a power-controlled feedback pulse. Those that demanded and received the entire item correctly also proceed to calculate the next item to access.

2. The sum of the acknowledging feedback pulses is used by the automaton at the server to update its estimation of the item probabilities.

3. In the variable rate system, the bit rate for the next broadcast of this item is calculated using the received feedback.

Assume that any client located at distance d receives items with $E_b = Th$. In the fixed bit rate system every item being broadcast is assumed to be correctly decoded at the mobile clients. As was mentioned earlier, item headers are broadcast with the default bit rate and are thus always correctly decoded by clients in the variable rate system as well. Item payloads however are correctly decoded by clients in the variable rate system if and only if they arrive at the demanding clients with an E_b figure being at least equal to Th. Thus, the error model is dynamic in terms of the transmission power per bit figure (thus with client-server antenna distance as well), as when this figure is lower from a certain threshold (Th) the item payload is assumed not correctly received at the client.

The simulation is carried out until at least N requests are satisfied at each client, meaning that overall, at least $N * ClNum$ requests have been served. Finally, the overhead due to the duration of the feedback pulse and the signal propagation delay is defined via the parameter Ovh.

D. Simulation results

The simulation results presented in this section were obtained with the following parameters values: $n = 4$, Dbs=300, $ClNum$=10000, $G = 5$, $Sec_1 = [0..119]$, $Sec_2 = [120..209]$, $Sec_3 = [210..239]$, $Sec_4 = [240..269]$, $Sec_5 = [270..299]$, $LP = 100$, N=1000, $Ovh = 0.1$, $Gr\_Size = 10$, L=0.15, a=10^{-4}. Client groups are uniformly distributed inside the service area.

In Figures 4-9 we keep the positions of groups fixed so as to examine the behaviour of the system for varying group sizes. Figures 4-6 display results when the item lengths follow the random distribution, while for the same network parameters, Figures 7-9 display results when the item lengths follow the increasing distribution. These Figures contain results that compare the performance of the fixed bit rate system to that of the adaptive one for different values of Dev in three different network environments, N_1, N_2, N_3. The simulation parameters of these three environments are:

1. *Network* N_1: $Loc_1 = 10, Loc_2 = 30, Loc_3 = 50, Loc_4 = 70, Loc_5 = 90, \theta 1 = 1.0$
2. *Network* N_2: $Loc_1 = 10, Loc_2 = 30, Loc_3 = 50, Loc_4 = 70, Loc_5 = 90, \theta 1 = 0.0$
3. *Network* N_3: $Loc_1 = 90, Loc_2 = 70, Loc_3 = 50, Loc_4 = 30, Loc_5 = 10, \theta 1 = 1.0$

In Figures 10, 11 we present the results of multiple simulations that compare the performance of the proposed approach for various values of Dev to that of the fixed rate system for item lengths following the random (Figure 10) and increasing distributions (Figure 11). What varies here (x-axis) is the positions of the groups. Every experiment is made for a different set of uniformly random selection of group placements. In these experiments we set $\theta = 1$ and $\theta 1 = 0$. Moreover, in order to assess the impact of parameter L and thus estimation accuracy on system performance we present in Figures 12, 13 results of system performance for various values of L. Finally, in Figure 14 we compare the performance of the proposed system to that of the fixed rate one for Networks N_1, N_2, N_3, for item lengths following the random distribution, when there is no locality of demand. Thus, this experiment assumes that demands are randomly distributed among all clients.

The main conclusions that can be drawn from the Figures are:

- The performance of all schemes improves for increasing values of the data skew parameter θ. This is expected behavior [3, 4, 6, 34], as increasing values of θ lead to increased commonality in client demands.
- The performance of the adaptive bit rate system is superior to that of the fixed bit rate one in all cases. This is due to the fact that in the adaptive system bit rate is not fixed but dynamically determined by client distance from the antenna; thus many items are

transmitted much faster than in the fixed bit rate system resulting to the overall performance increase. This fact is supported by the data in Table 1, which shows the mean bit duration for data payload for the two schemes and the various values of Dev for Networks N_1, N_2, N_3 for $\theta = 1.0$ when using the random item length distribution. The mean bit duration is normalized to the bit duration of the fixed-rate system. From Table 1 and Figures 4-6, it can be seen that for a certain Network, a small mean bit duration results in performance improvement over cases where the mean bit duration is higher.

- For increasing values of Dev the performance of the adaptive system declines, remaining however significantly superior to that of the fixed bit rate system. This is due to the fact that as $Dev \neq 0$, not all members of a group are located at the same distance from the antenna; thus in some cases information item broadcasts for a certain client group are also acknowledged by clients further away than the location of the main group. Thus in many cases, it is the feedback pulse of the client that is furthest away that determines the bit rate to be used for the next broadcast of the same information item. This reasoning also explains the fact that for a certain Network, larger mean bit durations are observed in Table 1 for larger values of Dev.

- The performance of the adaptive bit rate system is sometimes better when the sizes of groups that are located closer to the antenna are larger (e.g. Figure 4 compared to 5 and 6 and Figure 7 compared to 8 and 9). This is due to the fact that when the groups close to the antenna are larger, most of the demands of the client population will be made from these groups. Therefore, most of the client population demands will be fulfilled via very small bit durations, leading to the performance improvement over cases where the larger groups are further away. However even in the case where the largest group is the one that is furthest away of the antenna (e.g. Figures 6 and 9), the performance of the adaptive bit rate system is significantly better than that of the fixed bit rate one.

- Another interesting observation is the fact that although in some cases the mean bit duration is essentially the same, there exists a small performance difference for some values of θ. For example, in the cases of random length distribution, the performances for $Dev = 0.15$ and $Dev = 0.30$ in Network N_2 is slightly worse than the corresponding ones in N_1. Such situations are attributed to the different sizes of the various groups that affect demand skewness. Returning to our example, in Network N_2 the fact that $\theta 1 = 0$ means that all groups are of the same size. Thus, the number of clients that make requests for each subset Sec_g of the database is the same. However, in Network N_1 the fact that $\theta 1 = 1$ makes some groups larger than others; thus some items in certain database subsets are now demanded by many more clients than items in other subsets, a fact that obviously increases demand skewness in N_1 and explains its small performance gain over N_2 in cases where the mean bit rates in N_1 and N_2 are nearly the same.

- It can be seen in Figures 10, 11 that the exact amount of performance gain of the proposed system over the fixed rate one is dependent on the actual placements of the groups. In these Figures, several maxima and minima occur for the overall mean access time in the proposed system. This is due to the actual random topologies: maxima occur when the random group placements result to most groups located close to the border of the service area (e.g. in experiment no. 10, groups are in positions 93, 77, 46, 94, 92), whereas minima occur in the opposite case. However, it can be clearly seen from Figures 10, 11 that irrespective of group placements the performance of the proposed approach is always significantly superior to that of the fixed rate system.

- Another interesting observation from Figures 10, 11 is that the performance of the proposed system over the various random topologies starts to smooth for increasing values of Dev. This is due to the reason explained earlier: for increasing Dev the performance of the adaptive system declines. An increasing Dev does not significantly affect performance in topologies where most client groups are close to the border of the service area (e.g. experiment no.10). However it significantly affects topologies where most client groups are closer to the antenna, as in this case an increasing number of clients will be located further away than the main groups and consequently these clients' acknowledgments will increase the bit durations. Thus, the overall mean access time increases and performance over the various random topologies is smoothed for an increasing Dev.

- From Figure 12, we can see that there do not exist significant differences in performance when using different values of parameter L. This is because the improvement of fast convergence in the case of a large L is negated by smaller estimation accuracy and vice versa. There exist only small performance differences in favour of using a smaller L both for the fixed rate system as well as for the adaptive rate one for $Dev > 0$. These differences are due to the effect of smaller estimation accuracy when using a large value of L. To show schematically these differences, the performances for the extreme cases of a small L (0.05) and a large L (0.4) of Figure 12 are plotted together in Figure 13.

- In Figure 14 we compare the performance of the proposed system to that of the fixed rate one for Networks $N_1, N_2, N_3$when demands are randomly distributed among all clients. Item lengths are computed according to the random distribution. We can see that even in that case where there is no locality of demand, the performance of the proposed system (dashed plots) is not worse than that of the fixed rate one (solid plots). We have obtained similar results for the case of the increasing length distribution as well.

V. Conclusions and future work

With the increasing popularity of wireless networks and mobile computing, data broadcasting has emerged as an efficient way of delivering data to mobile clients having a high degree of commonality in their demand patterns. In many cases clients are grouped into several groups, each one in a different location, with the members of each group having similar demands. This paper proposed a mechanism that exploits locality of demand in order to increase the performance of wireless data dissemination systems. Specifically, it trades the E_b redundancy at a distance smaller than the coverage radius, for an increased bit rate for transmission of items demanded by client groups at this distance. Knowledge of client positions is conveyed to the server via a simple feedback from the clients. Simulation results have been presented that reveal significant performance improvement over fixed bit rate systems in environments characterized by locality of client demands.

In the context of the proposed adaptive rate system, the focus of our ongoing research is to use multiple directional antennas instead from a single omnidirectional one. Equipped with one Learning Automaton per antenna, this approach builds on the idea that if a group of interest is known to be in one location, then the corresponding antenna will transmit a beam only in that direction, thus resulting to significant overall performance increase.

ACKNOWLEDGMENTS

The authors wish to thank the Associate Editor Prof. W. Zhuang and the anonymous reviewers for their insightful and useful comments which helped us to improve our work.

REFERENCES

[1] P. Nicopolitidis, M. S. Obaidat, G. I. Papadimitriou and A. S. Pomportsis, ”Wireless Networks”, John Wiley and Sons Ltd., 2003.

[2] D.Aksou and, M.Franlkin, ”RxW: A Scheduling Approach for Large-Scale On-Demand Data Broadcast”, ACM/IEEE Transactions on Networking, vol.7, no.6, pp.846-860, December 1999.

[3] S.Acharya, M.Franklin and S.Zdonik, ”Dissemination-based Data Delivery Using Broadcast Disks”, IEEE Personal Communications. vol.2, no.6, pp 50-60, December 1995.

[4] N.H.Vaidya and S.Hameed, ”Scheduling Data Broadcast In Asymmetric Communication Environments”, Wireless Networks, vol.5, no.3, pp.171-182, May 1999.

[5] C.J.Su and L.Tassiulas, ”Broadcast Scheduling for Information Distribution.”, In Proceedings of the IEEE INFOCOM, pp.109-117, May 1997.

[6] 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.

[7] W.Sun, W.Shi and B.Shi, ”A Cost-Efficient Scheduling Algorithm of On-Demand Broadcasts”, Wireless Networks, vol.9, no.3, pp.239-247, May 2003.

[8] N.Vlajic, C.D.Charalambous and D.Makrakis, ”Performance Aspects of Data Broadcast in Wireless Networks With User Retrials”, IEEE/ACM Transactions on Networking, vol.12, no.4, pp.620-633, August 2004.

[9] C-L.Hu and M-S Chen, ”Adaptive Multichannel Data Dissemination: Support of Dynamic Traffic Awareness and Push-Pull Time Balance”, IEEE Transactions on Vehicular Technology, vol.54, no.2, pp.673-686, March 2005.

[10] T.Sakata and J.Xu Yu, ”Statistical Estimation of Access Frequencies Problems, Solutions and Consistencies”, Wireless Networks, vol.9, no.6, pp.647-657, November 2003.

[11] Q.Hu, D-L. Lee and W-C Lee, ”Dynamic Data Delivery in Wireless Communication Environments”, in Proceedings of Workshop on Mobile Data Access, pp.213-224, November 1998.

[12] W. Wang, and C. Ravishankar, ”Adaptive Data Broadcasting in Asymmetric Communication Environments”, in Proceedings of IEEE International Database Engineering and Applications Symposium, pp.27-36, July 2004.

[13] C-L.Hu and M-S. Chen, ”Adaptive Information Dissemination: An Extended Wireless Data Broadcasting Scheme with Loan-Based feedback Control”, IEEE Transactions on Mobile Computing, vol.2, no.4, pp.332-336, October-December 2003 .

[14] K.Stathatos, N.Roussopulos and J.S.Baras, ”Adaptive Broadcast in Hybrid Networks”, In Proceedings of VLDB 1997, pp.326-335.

[15] J.B.Andersen, T.S.Rappaport and S.Yoshida, ”Propagation Measurements and Models for Wireless Communication Channels”, IEEE Communications Magazine, vol.33, no.1, pp.42-49, January 1995.

[16] T. Bowen, et al, ”The Datacycle Architecture”, Communications of the ACM, vol.35, Communications of the ACM, vol.35, no.12, pp.71-81, December 1992.

[17] M.H.Ammar and J.W.Wong, ”On the Optimality of Cyclic Transmission in Teletext Systems”, IEEE Transactions on Communications, vol.35, no.1, pp.68-73, January 1987.

[18] R.Jain and J.Werth, ”Airdisks and airRAID: Modeling and scheduling periodic wireless data broadcast (extended abstract)”, Technical Report, Rutgers University, 1995.

[19] S.Acharya, M.Franklin and S.Zdonik, ”Disseminating Updates on Broadcast Disks”, In Proceedings of VLDB , pp.354-365, September 1996.

[20] S.Acharya, M.Franklin and S.Zdonik, ”Balancing Push and Pull for Data Broadcast”, In Proceedings of ACM SIGMOD, pp.183-194, May 1997.

[21] W.G.Yee W.G and S.B.Navathe, ”Bridging the Gap between Response Time and Energy Efficiency in Broadcast schedule Design”, in Proceedings of 2002 Extending Database Technology Conference, pp.572-589, March 2002.

[22] N.H.Vaidya and S.Hameed, ”Data Broadcast Scheduling: On-line and off-line algorithms”, Technical Report 96-017, Computer Science Department, Texas A&M University, 1996.

[23] K.Najim and A.S.Poznyak, ”Learning Automata: Theory and Applications”, Pergamon, New York, 1994.

[24] K.S.Narendra and M.A.L.Thathachar, ”Learning Automata: An Introduction”, Prentice Hall, 1989.

[25] 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, vo.6, no.4, pp.649-654, August 1994.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] P.Nicopolitidis, G.I.Papadimitriou and A.S.Pomportsis, ”Learning-Automata-Based Polling Protocols for Wireless LANs”, IEEE Transactions on Communications, vol.51, no.3, pp.453-463, March 2003.

[31] E.W.Kamen, ”Introduction to Signals and Systems”, 2nd edition, Macmillan, 1990.

[32] J. Millman and C. Halkias, ”Integrated Electronics: Analog and Digital Circuits and Systems”, McGraw-Hill, 1971.

[33] W.D.Walden, ”Selecting and Using Transducers for the Measurement of Electric Power, Voltage and Current”, Ohio Semitronics, Inc. August 1998, revised January 7, 2000 Rev. A, available at http://www.ohiosemitronics.com/pdf/tech_papers/Transducer-book-dec-04.pdf

[34] P.Nicopolitidis, G.I.Papadimitriou and A.S.Pomportsis, ”On the Implementation of a Learning Automaton-based Adaptive Wireless Push System”, in Proceedings of Symposium on Performance Evaluation of Computer and Telecommunication Systems 2001, pp.484-491.

	N_1	N_2	N_3
Fixed-rate	1	1	1
Adaptive, Dev=0	0.12	0.25	0.43
Adaptive, Dev=0.15	0.58	0.63	0.68
Adaptive, Dev=0.3	0.75	0.78	0.79

Table 1: Mean bit duration in Networks N_1, N_2, N_3 for the fixed and adaptive rate systems when using the random item length distribution. $\theta = 1.0$.

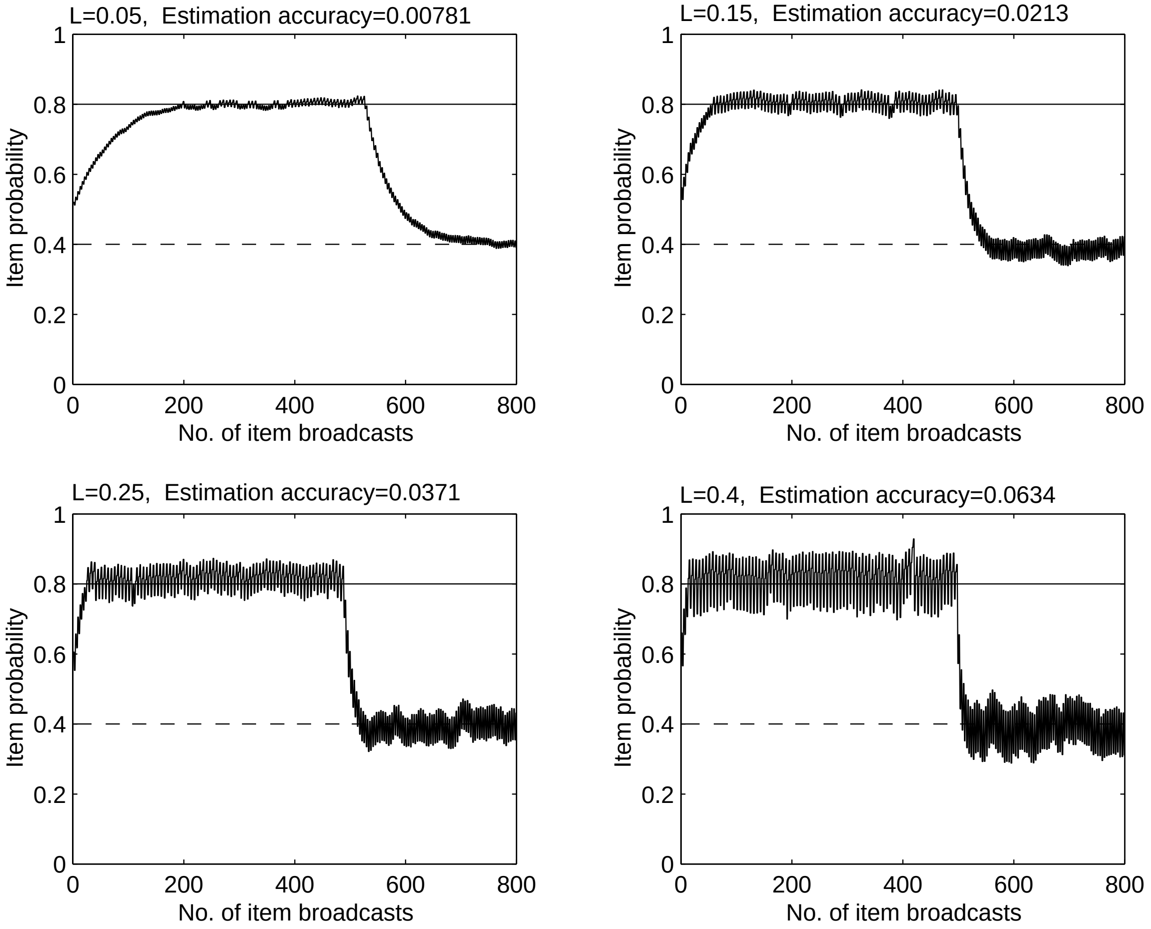

Figure 1: Convergence of automaton estimation of the demand of a data item.

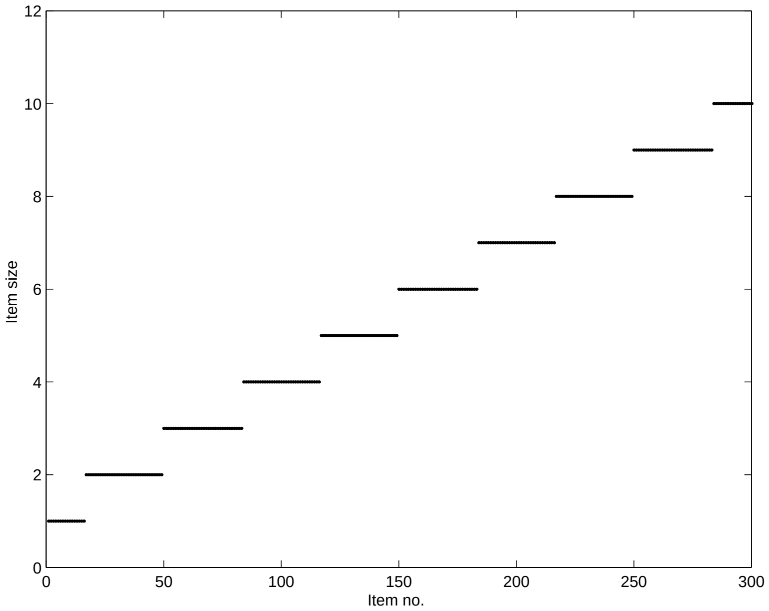

Figure 2: Lengths of server's data items calculated using the increasing length distribution.

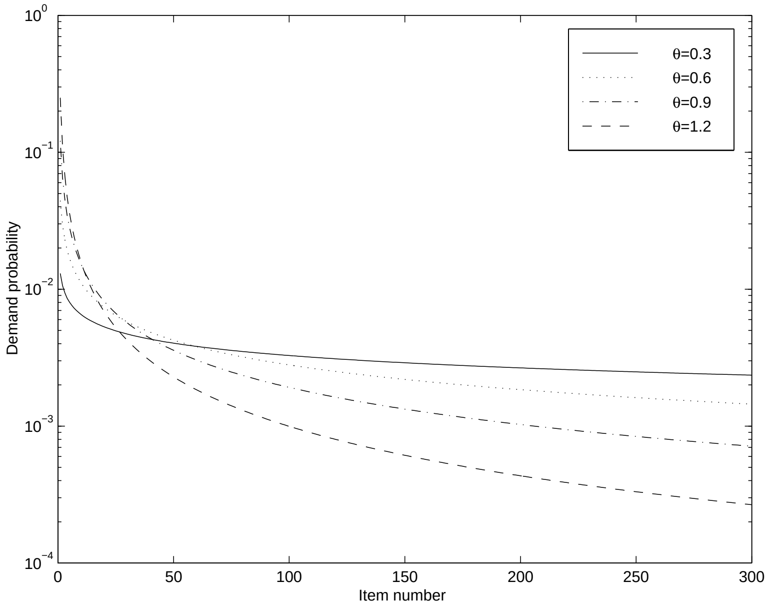

Figure 3: Item demand probabilities produced by the Zipf distribution for different values of θ.

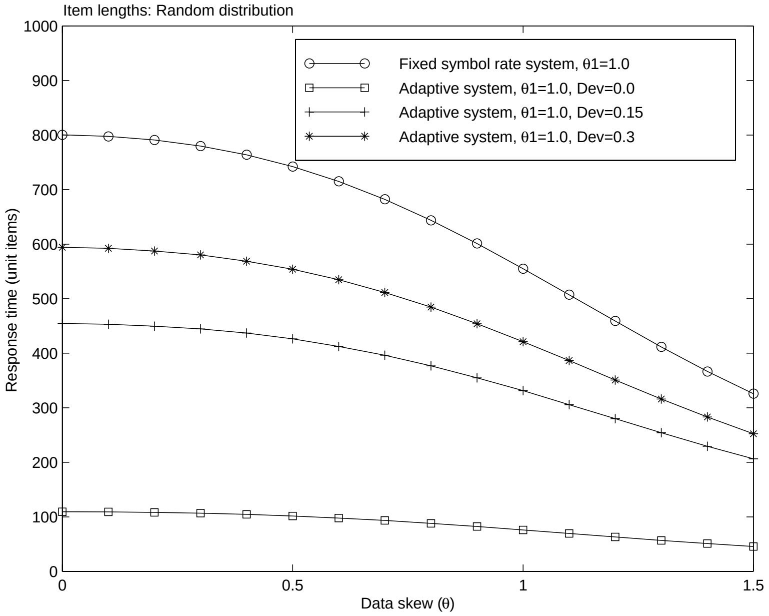

Figure 4: Overall Mean Access Time in unit items versus access skew coefficient θ. Client Groups are in positions 10, 30, 50, 70, 90. Item lengths are calculated with the random distribution. $\theta 1 = 1$.

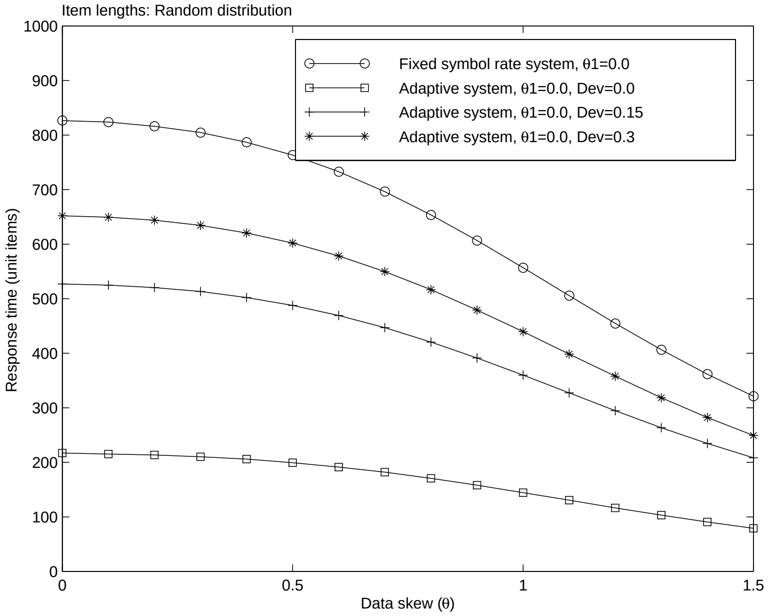

Figure 5: Overall Mean Access Time in unit items versus access skew coefficient θ. Client Groups are in positions 10, 30, 50, 70, 90. Item lengths are calculated with the random distribution. $\theta 1 = 0$.

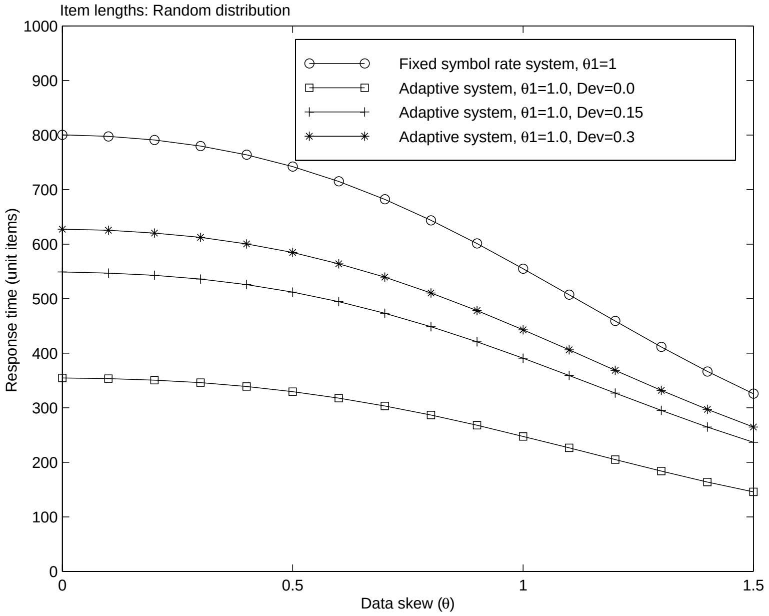

Figure 6: Overall Mean Access Time in unit items versus access skew coefficient θ. Client Groups are in positions 90, 70, 50, 30, 10. Item lengths are calculated with the random distribution. $\theta 1 = 1$.

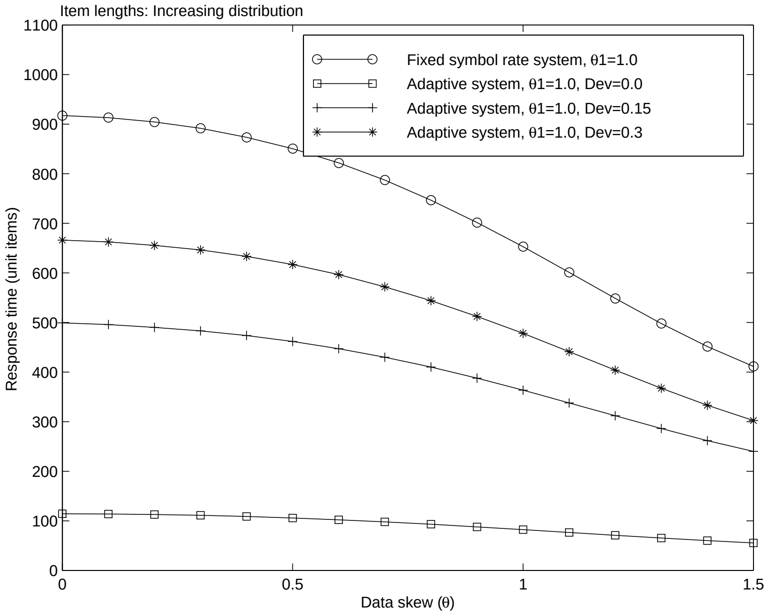

Figure 7: Overall Mean Access Time in unit items versus access skew coefficient θ. Client Groups are in positions 10, 30, 50, 70, 90. Item lengths are calculated with the increasing distribution. $\theta 1 = 1$.

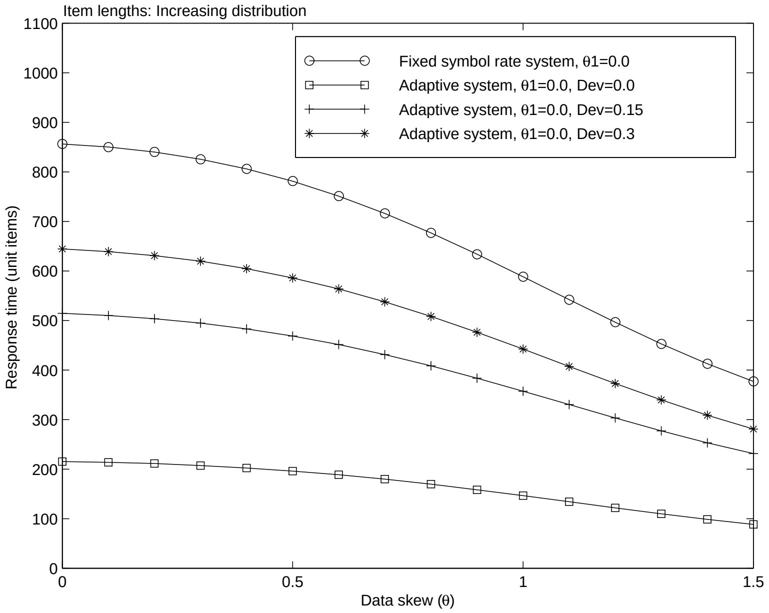

Figure 8: Overall Mean Access Time in unit items versus access skew coefficient θ. Client Groups are in positions 10, 30, 50, 70, 90. Item lengths are calculated with the increasing distribution. $\theta 1 = 0$.

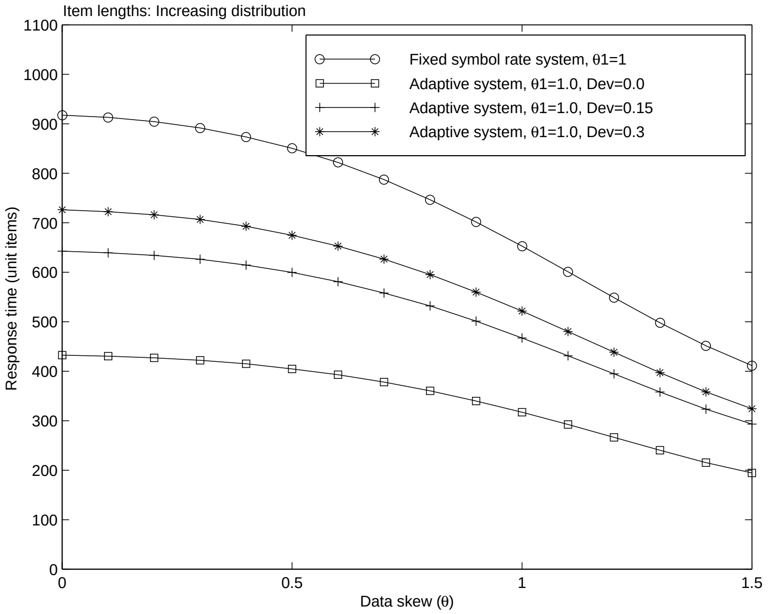

Figure 9: Overall Mean Access Time in unit items versus access skew coefficient θ. Client Groups are in positions 90, 70, 50, 30, 10. Item lengths are calculated with the increasing distribution. $\theta 1 = 1$.

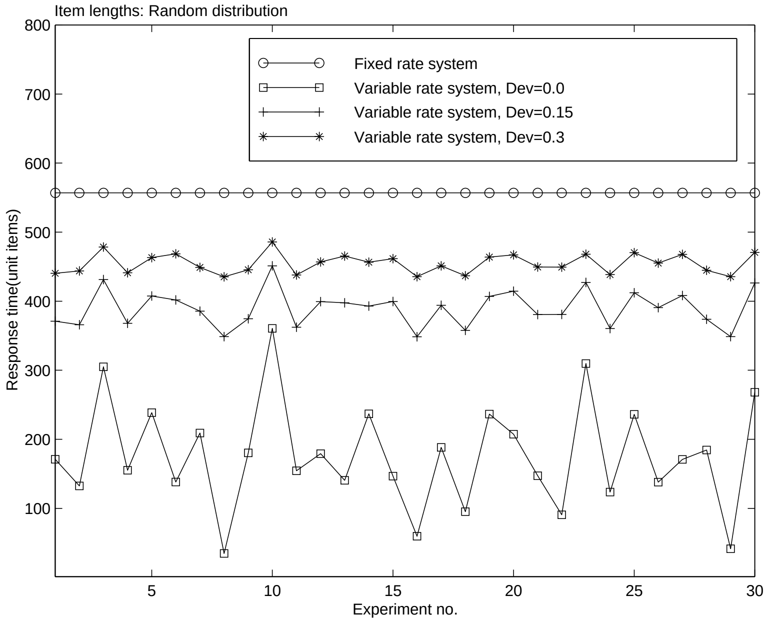

Figure 10: Overall mean access time for various uniformly random placements of groups. $\theta = 1, \theta 1 = 0$. Item lengths are calculated with the random distribution.

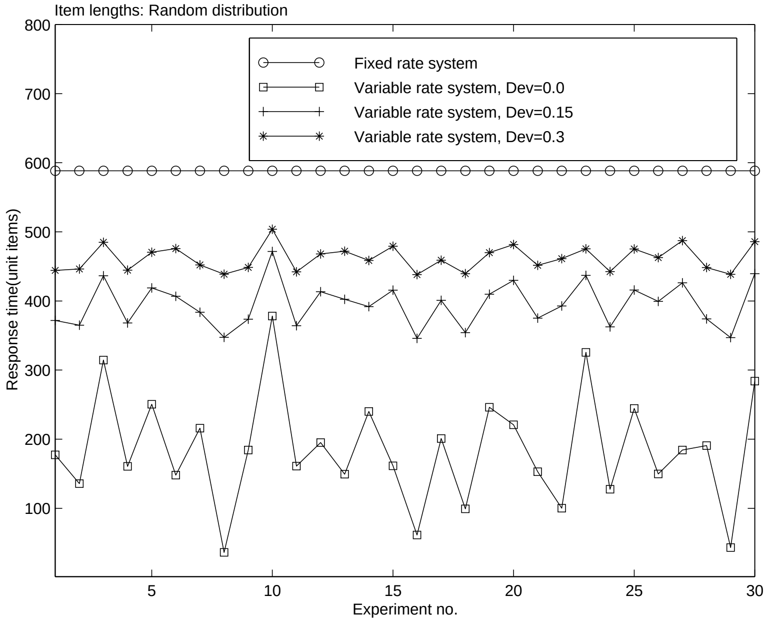

Figure 11: Overall mean access time for various uniformly random placements of groups. $\theta = 1, \theta 1 = 0$. Item lengths are calculated with the increasing distribution.

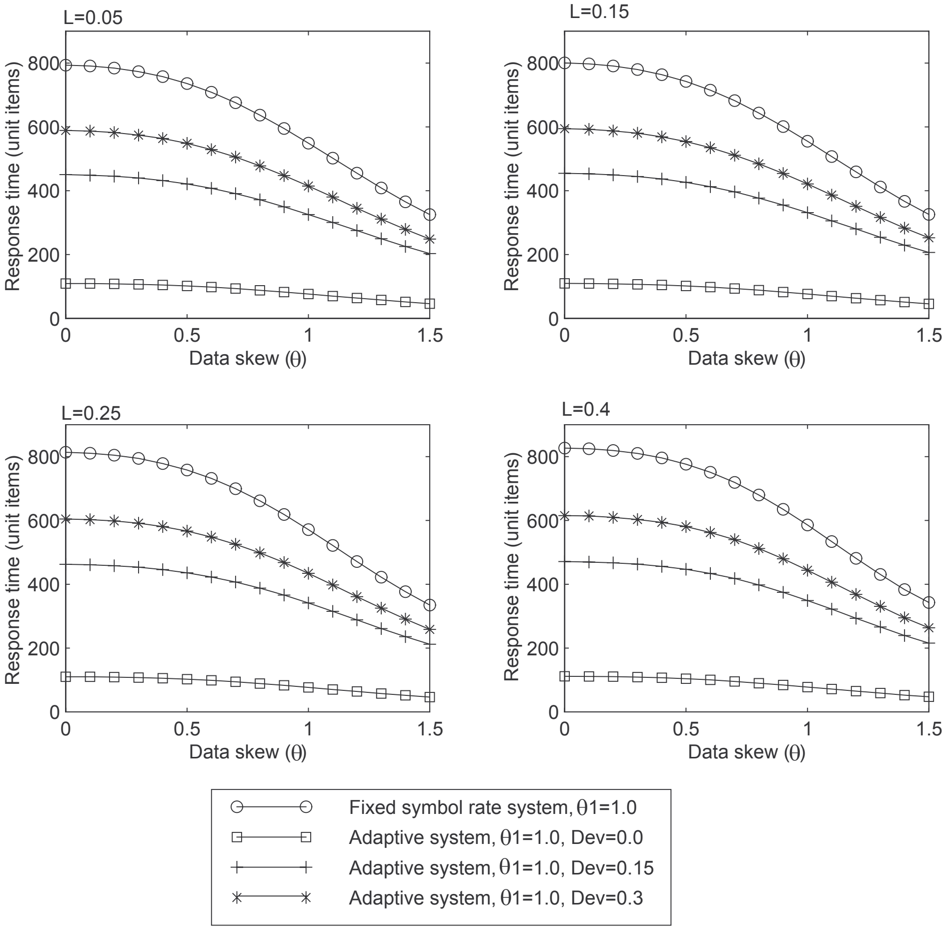

Figure 12: Overall Mean Access Time in unit items versus access skew coefficient θ for various values of L. Client Groups are in positions 10, 30, 50, 70, 90. Item lengths are calculated with the random distribution. $\theta 1 = 1$.

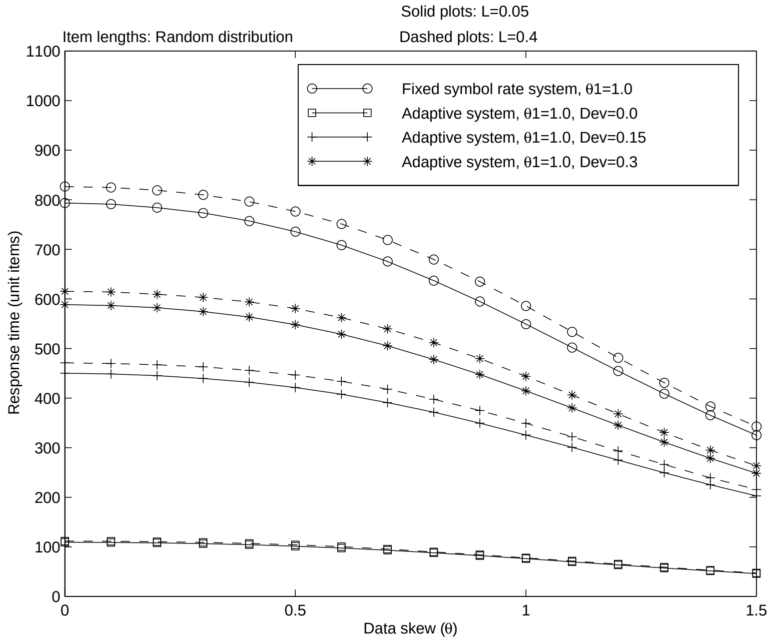

Figure 13: Overall Mean Access Time in unit items versus access skew coefficient θ for $L = 0.05$ (solid plots) and $L = 0.4$ (dashed plots). Client Groups are in positions 10, 30, 50, 70, 90. Item lengths are calculated with the random distribution. $\theta 1 = 1$.

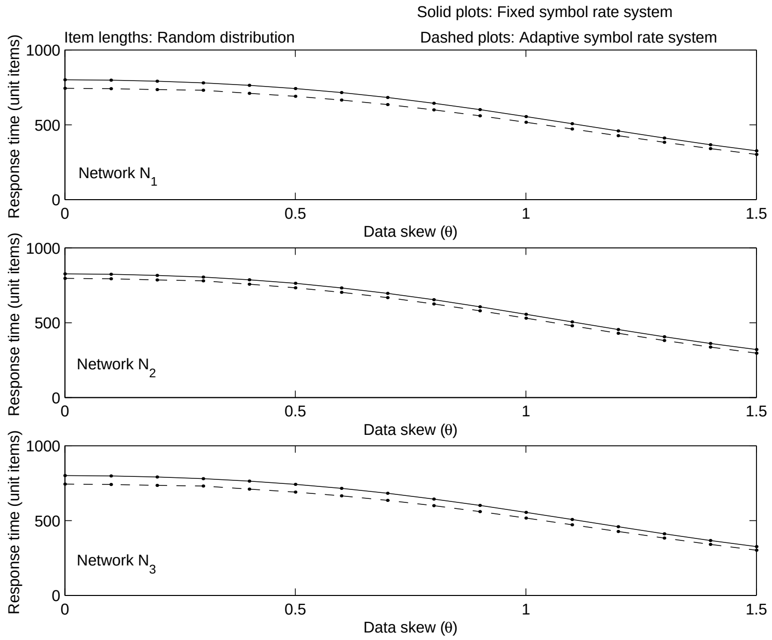

Figure 14: Overall Mean Access Time in unit items versus access skew coefficient θ when demands are randomly distributed among all clients, for Networks N_1, N_2, N_3. Item lengths are calculated with the random distribution.