

Adaptive Balanced Hybrid Data Delivery for Multi-Channel Data Broadcast

Chih-Lin Hu and Ming-Syan Chen

Department of Electrical Engineering

National Taiwan University

Taipei, Taiwan, R.O.C.

Abstract—With the proliferation of wireless information applications and services, data broadcasting known from radio networks has become an important mechanism for mobile data access in wireless communication networks. Two data delivery techniques, push and pull, and their hybrid combination had been introduced to satisfy mobile user demands. However, most previous research efforts were elaborated upon the premise of a single channel with static workload and access pattern. Recently the emergence of a multi-channel broadcast paradigm has attracted much research attention in the data broadcast community. In this paper, we design an adaptive balanced scheme (ABS) which performs a heuristic search in pursuit of a fair balance of access time for a hybrid data delivery in a multi-channel data broadcast environment. The experimental results show that at the balanced point exploited by ABS, the result of channel partition and data item classification is very close to the optimal balance, and consequently, we are able to obtain the lowest mean access time in both push and pull channels simultaneously.

Keywords—Push, pull, data broadcast, mobile computing, wireless network.

I. INTRODUCTION

The wireless communication environment has a feature of bandwidth asymmetry since a downlink channel is of much larger bandwidth than the uplink channel. With the limited bandwidth capacity, it is essentially very difficult for a wireless information server to handle a large population of mobile clients individually and simultaneously. Therefore, the asymmetry imposes constraints and challenges on the behavior between clients and servers as well as on the design and development of wireless information applications and services.

Data broadcasting is a promising mechanism for resolving communication asymmetry against information dissemination in wireless network environments [1][7]. As the broadcast paradigm, a server applies a broadcast cycle, which is a series of interleaved data slots, periodically delivering all data items in the database to clients through a shared medium. The benefits are twofold: (1) in the client side, data broadcasting achieves a total saving of all clients' battery energy by avoiding per access request transmission, (2) an information server can moderately mitigate the inherent performance and scalability problems.

Originally a server can choose a pull- or push-based data delivery to transmit data. However, a push delivery which broadcasts all items periodically can result in an unacceptable access time if the number of data items in the database is huge. In contrast, a pull delivery responding to a client's individual request inevitably incurs a scalability bottleneck with a heavy workload. In view of this, a hybrid delivery is called for to strike a comparison between the trade-offs. Explicitly, data items are classified as hot (or popular) and cold (or unpopular) items according to their access frequencies, and data slots in a broadcast cycle are partitioned into the push and pull sets. Push slots

carry hot items and a cold item is delivered on a pull slot by a request-response way. With the commonality of access interests, a hybrid delivery mode provides an opportunity to make the access time malleable.

The recent advance of a multi-channel broadcast paradigm has a different utilization on downward bandwidth [6][9][10][11]. The downward bandwidth is divided into a number of sub-channels. Then, an adaptive hybrid data delivery should periodically arrange the push and pull items and allocate push and pull channels in response to a traffic change, in order to attain an appropriate access time. However, most prior researches in the adaptation and balance of broadcasting mainly considered a single channel scenario rather than a multi-channel one. In [2][3], the bandwidth assigned to push and pull slots is static and the contents of push and pull sets are constant, leading to no adaptation to various workloads. The work in [5] presents a feedback technique to understand the change of access pattern, assuming that all clients actively report access statistic. [12] presents a probing technique to collect changing access patterns and uses marginal gains to perform adaptation. [4] proposes an optimal cut-off scheme to determine the push and pull item sets. Recently, [9] is an analytical study about the multi-channel partition and [11] is designed for caching. [10] dynamically generates a broadcast disk program to improve energy saving and [6] devises an interpolation search scheme for the adaptation extended from [10]. These works did not aim at the design of an adaptive balanced hybrid data delivery scheme.

Balancing push and pull response time is of vital importance, involving the channel allocation and data classification against traffic changes. In this paper we propose an adaptive balanced scheme (ABS) for a hybrid data delivery in a multi-channel data dissemination environment. ABS performs a deterministic search for a balanced point by adaptively allocating the push and pull channels and classifying the push and pull items in response to dynamic traffic changes. Therefore, ABS is able to discover the lowest balanced point where the mean access time is very close to the optimal one and the difference between the mean push and pull access time is minimized. The characteristic of its dynamic adaptability in a multi-channel scenario distinguishes ABS from the previous works. It is shown by the experimental results that ABS can obtain a balanced point with fairness in reducing both push access and pull response time.

The rest of the paper is organized as follows. Preliminary is given in Section 2. Section 3 presents the adaptive balanced scheme (ABS). Section 4 describes the simulation model and

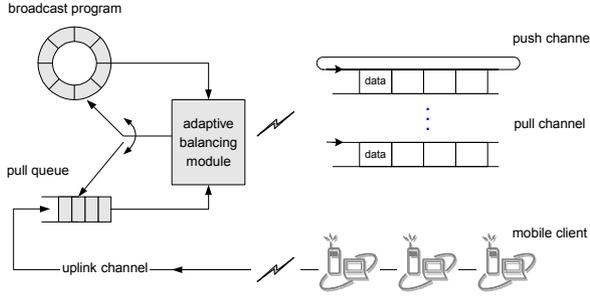


Fig. 1. The integrated data broadcast in an asymmetric communication environment

demonstrates the experimental results. This paper concludes with Section 5.

II. PRELIMINARY

A. Model the Data Dissemination Environment

Fig. 1 illustrates a multi-channel hybrid data delivery in which we consider the adaptation and balance works. A wireless information server interacts with clients in its service coverage over the wireless medium. The broadcast downlink bandwidth is divided into a number of sub-channels of smaller, equal bandwidths. The server contains a database including all broadcast items which are classified as hot or cold ones according to their previous access frequencies. The server broadcasts hot items over push channels periodically and delivers cold items by pull channels in response to clients' explicit access requests. Moreover, a server provides an uplink channel to receive clients' pull requests. The adaptive module attempts to balance mean access and response time of push and pull access respectively under a dynamic nature of broadcast traffic. The assumptions adopted in this study are as follows.

- The length of broadcast cycle is not fixed.
- Broadcast program is dynamic with various traffic factors.
- Each item is self-identified and read-only.
- The uplink channel bandwidth is constant. There is only one uplink channel in this model.
- A downlink channel is either in the pull or push mode, and can alternatively be switched to complement other channels.

Note that the model is similar to the data dissemination environment in [9][10][12], while having a graceful extension for the multi-channel paradigm.

B. Measurement Formulation

Consider a server has a database that contains $|D| = M$ items of equal size s . Assume that the request for each item d_i forms a Poisson process of rate λ_i with an order as $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_M$. Then the total request arrival rate is $\lambda = \sum_{i=1}^M \lambda_i$. The server has a service rate for each channel is $\mu = b/s$. We let $\lambda_{push} = \sum_{i=k+1}^M \lambda_i$ be the push access rate and $\lambda_{pull} = \lambda - \lambda_{push} = \sum_{i=1}^k \lambda_i$ be the pull request arrival rate.

B.1 Pull average response time in on-demand channel

Since there is a single server, the pull delivery system can be modeled as a birth-death process with a birth rate λ_{pull} and a constant death rate $\mu_{pull} = \mu \times c_{pull}$. Then c_{pull} is the number of pull channels as a control factor for dynamic channel allocation. Each pull channel is viewed identical and pull channels are coalesced, not merged, to serve the arrival requests. We then model this case as an M/M/m system [8] where $m = c_{pull}$ is the number of pull channels and $\lambda_{pull} < m \times \mu$ is the stable condition. Consequently, the mean pull response time in the system is

$$\pi_0 = \left[\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \frac{(m\rho)^m}{m!(1-\rho)} \right]^{-1}$$

and

$$\bar{t}_{pull} = \frac{1}{\mu} + \frac{(m\rho)^m}{m!(m\mu - \lambda)(1-\rho)} \pi_0 \quad (1)$$

where $\rho = \lambda/m\mu$ is the *traffic intensity* and π_0 is the steady-state probability that no request waits in the queue.

B.2 Push average response time in broadcast channel

$$\bar{t}_{push} = \frac{sk}{(2bc_{push})} + \frac{s}{b} = \frac{k}{2\mu c_{push}} + \frac{1}{\mu} \quad (2)$$

where k is the number of items which are broadcast periodically, s/b is the inverse of channel service rate and $ks/(bc_{push})$ is the time period of a broadcast cycle b_c .

III. DESIGN AN ADAPTIVE BALANCED SCHEME

The adaptive balanced scheme (ABS) we should devise will improve the adjustment of channel allocation and data grouping under a variety of traffic factors, such as dynamic workload, client access pattern, user behavior, etc., and therefore achieve the fairness in terms of push and pull access/response time for a multi-channel data broadcast.

ABS begins with a broadcast mode that all channels are in the push mode and data in the broadcast program are scheduled in ascending order of access frequencies. ABS firstly tries if altering a push channel as the pull mode will reduce both of \bar{t}_{push} and \bar{t}_{pull} . According to (2), \bar{t}_{push} has a linear progression with the amount of push items, whereas \bar{t}_{pull} is vulnerable to traffic intensity, especially request frequency, as expressed in (1). That is, ABS will do adaptive balancing by following Criterion-1 below. When a push channel is switched to the pull mode, a decrease of \bar{t}_{pull} causes imbalance. Then ABS starts to repeatedly demote an item of $\min\{\lambda_i | d_i \text{ in push item set } U_{push}\}$ into the pull item set U_{pull} , and stops demotion as long as an added item causes \bar{t}_{pull} to be larger than \bar{t}_{push} . Since an increased \bar{t}_{pull} means a decreased \bar{t}_{push} , there exists a cross point where ABS reaches a balanced point by monitoring the value of \bar{t}_{push} . The trial process in a channel switch will stop when ABS detects that a trial \bar{t}_{push} goes up or all channels have been switched as the pull mode. Some criteria of ABS are given below.

Criterion 1 At a local balanced point, \bar{t}_{pull} should be lower than \bar{t}_{push} and have a minimal distance $Min(\bar{t}_{pull}, \bar{t}_{push})$.

Criterion 2 ABS asserts an adaptation when \bar{t}_{push} in a trial is lower than the last \bar{t}_{push} .

Criterion 3 An upward \bar{t}_{push} in a trial indicates a worse balanced point.

Criterion 4 If no push channel is available, that is, a pure pull-based mode, \bar{t}_{pull} will alternatively be used to compare with the last \bar{t}_{push} .

As mentioned above, ABS looks for a new proper balanced point when a traffic change causes imbalance. It is noted that as for the detection of access frequency, one can employ the probing technique in [12] which has been used in [3] too. However, unlike prior works on a single channel paradigm where a data slot is the replacement unit, the switch unit in ABS is a channel as well as a series of item demotion/promotion. For interested readers, the algorithmic form of ABS is given as follows.

Input: $U_{push} = \{D\}, U_{pull} = \{\emptyset\}, c_{down}, b_c, \mu, \lambda, buffer$

Output: $U_{push}, U_{pull}, c_{push}, c_{pull}, \bar{t}_{push}, \bar{t}_{pull}, b_c$

begin

```

1.  $\bar{t}_{1push} \leftarrow \text{evaluatePushTime}(U_{push}, c_{push}, \mu)$ 
2.  $\bar{t}_{1pull} \leftarrow \text{evaluatePullTime}(c_{pull}, \lambda_{pull}, \mu, b)$ 
3. while  $c_{push}$  is not zero and  $U_{push}$  is not empty do
4.    $\bar{t}_{1push} \leftarrow \text{evaluatePushTime}(U_{push}, c_{push}, \mu)$ 
5.    $\bar{t}_{1pull} \leftarrow \text{evaluatePullTime}(c_{pull}, \lambda_{pull}, \mu, b)$ 
6.    $c_{push} \leftarrow c_{push} - 1$ 
7.   if  $c_{push}$  is smaller than zero then
8.     break
9.   endif
10.   $c_{pull} \leftarrow c_{pull} + 1$ 
11.  while  $\lambda_{pull}$  is smaller than  $\mu c_{pull}$  do
12.    if  $U_{push} = \{D\}$  and  $c_{push}$  is larger than zero then
13.       $\bar{t}'_{2push} \leftarrow \text{evaluatePushTime}(U_{push}, c_{push}, \mu)$ 
14.       $\bar{t}'_{2pull} \leftarrow \text{evaluatePullTime}(c_{pull}, \lambda_{pull}, \mu, b)$ 
15.    else
16.       $\lambda_{pull} \leftarrow \lambda_{pull} + \min\{\lambda_i | d_i \in U_{push}\}$ 
17.       $\bar{t}'_{2push} \leftarrow \text{evaluatePushTime}(U_{push}, c_{push}, \mu)$ 
18.       $\bar{t}'_{2pull} \leftarrow \text{evaluatePullTime}(c_{pull}, \lambda_{pull}, \mu, b)$ 
19.       $\bar{t}_{2push}, \bar{t}_{2pull} \leftarrow \text{compareToLastDemotion}(\bar{t}'_{2push}, \bar{t}'_{2pull})$ 
20.    endif
21.  endwhile
22.   $isBetter \leftarrow \text{compareToLastBalance}(\bar{t}_{1push}, \bar{t}_{1pull}, \bar{t}_{2push}, \bar{t}_{2pull})$ 
23.  if  $isBetter$  is True then
24.    return  $U_{push}, U_{pull}, c_{push}, c_{pull}, \bar{t}_{push}, \bar{t}_{pull}, BMCycle$ 
25.  else
26.    continue
27.  endif
28. endwhile
end

```

IV. EXPERIMENTS AND RESULTS

A. Simulation Model

Table 1 lists the simulation parameters. We use a general Zipf distribution expressed as $p_i = (\frac{1}{i})^\theta / \sum_{i=1}^M (\frac{1}{i})^\theta$, where θ is a skew coefficient and $1 \leq i \leq M$.

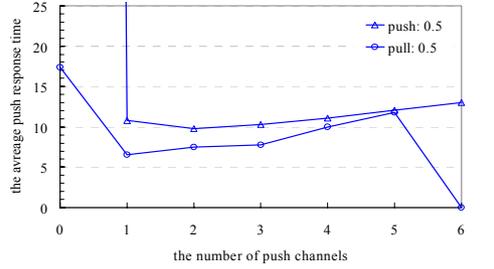
TABLE I
SIMULATION PARAMETERS DESCRIPTION

notation	meaning	value
s	data item size	100 Kbytes
c_{down}	channel number	6
b	downward bandwidth	1200 Kbyte
$ D = M$	number of data items in database	300
λ	workload, request arrival rate by Poisson	8~30
θ	skew coefficient, access pattern by Zipf	-1.0~1.0
μ	service rate (b/s)	2

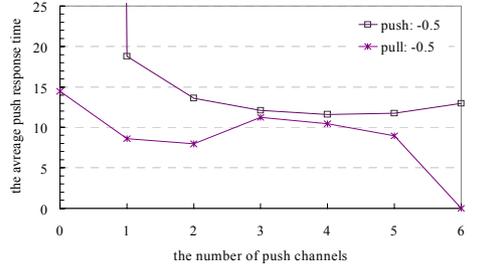
B. Sensitivity to Dynamic Access Frequency

Under a constant workload $\lambda = 15$, Fig. 2(a)-(c) show the curves of the push access and pull response time under a variety of dynamic access patterns. As shown, when a push channel is switched to the pull mode, ABS immediately demotes hot items into push channels so as to look for a new balance where \bar{t}_{pull} is lower than \bar{t}_{push} and the distance between \bar{t}_{pull} and \bar{t}_{push} is minimal. A new \bar{t}_{push} lower than that in the last trial of a channel switch is preferable. Following this trial process, ABS continues until all 6 channels are in the pull mode, showing a pure pull delivery in this case. Because \bar{t}_{push} is not available in this trial, ABS compares \bar{t}_{pull} with \bar{t}_{push} in the last trial. If \bar{t}_{pull} is lower than the last \bar{t}_{push} , ABS employs a pure pull delivery.

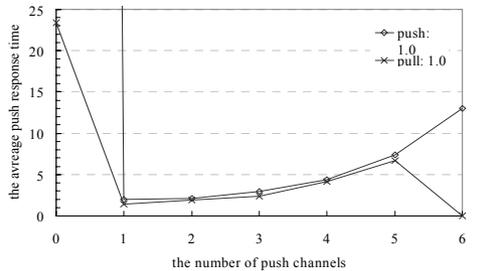
In Fig. 2(a), both push and pull curves go up drastically as the last push channel is switched to the pull mode. The infinite \bar{t}_{push} indicates that some hot items remain in the push set but there is no more push channel because the workload is larger than the pure pull service rate. Fig. 2(b) and (c) show a comparison by the skewed frequency distributions. Using $\theta = -0.5$ generates a Zipf distribution that is close to an uniform distribution, but a distribution with $\theta = 1$ is very skewed. The latter has a much lower \bar{t}_{push} than the former. The reason is that delivering a few items holding a major access frequency by the push mode can significantly reduce the pull traffic load. If a very hot item is incorrectly demoted as a cold one, surely high broadcast misses will result in many pull requests for it. With ABS, we observed that a less skewed frequency distribution will have a higher \bar{t}_{push} , and a hybrid data delivery favors a high skewed access frequency distribution. Moreover, a larger mean access time does not imply a higher access frequency. Even though a static workload, a change of client access interest can result in a higher access time.



(a) dynamic access frequency (Zipf theta=0.5) vs. average access time



(b) dynamic access frequency (Zipf theta=-0.5) vs. average access time

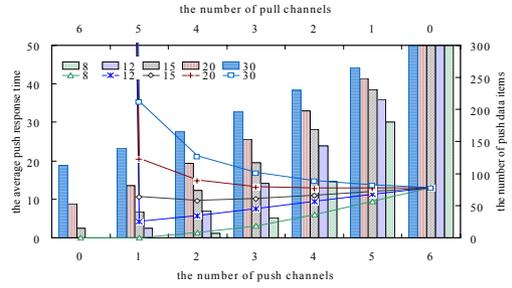


(c) dynamic access frequency (Zipf theta=1.0) vs. average access time

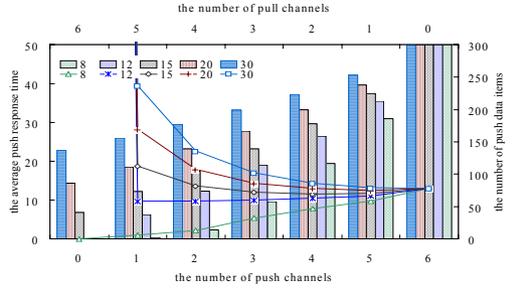
Fig. 2. With dynamic data access frequency, ABS dynamically adjusts channel allocation and data classification for a balanced searching. (λ :15 req./sec)

C. Sensitivity to Dynamic Workload

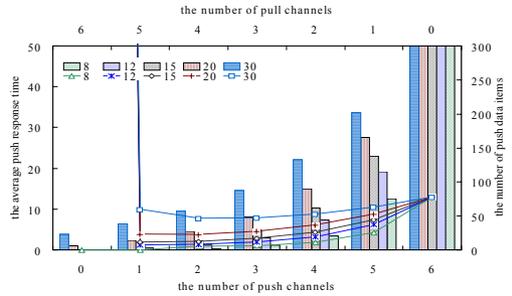
In general, a dynamic workload can be caused by a change of client population. Fig. 3 explicitly depicts the time curves versus adaptive channel allocation, and the bar chart indicates the number of items delivered on push and pull channels respectively by adaptive item promotion/demotion with a channel allocation. Fig. 3(a) illustrates a Zipf distribution with $\theta = 0.5$ under various workloads with a light load $\lambda = 8$ and 12 and a heavy load $\lambda = 15, 20$ and 30 respectively. Initially, all 6 channels are in the push mode and all 300 items are push items. ABS then looks for a *local optimal* balance by iteratively switching a push channel and demoting push items as long as \bar{t}_{pull} is smaller than \bar{t}_{push} and the pull aggregated frequency does not reach the upper limit. If a new local balance is lower than the latest balance point, ABS will use the newly discovered local point as the new balanced point. As shown in the curve of workload $\lambda = 8$, ABS gets the best balance at all 6 channels as the pull mode in this case, showing a pure pull delivery. However with workload $\lambda = 12$, ABS suffers



(a) Zipf distribution with theta = 0.5



(b) Zipf distribution with theta = -0.5



(c) Zipf distribution with theta = 1

Fig. 3. ABS searches a balance under dynamic workloads.

an unstable circumstance when it attempts to switch the last push channel to the pull mode. With 6 pull channels, all items have to be delivered by the pull mode. However, workload $\lambda = 12$ will make \bar{t}_{pull} unpredictable according to an M/M/m queueing model. Hence ABS goes back to the last trial which allocates one push channel to contain 15 most hot items with $\bar{t}_{push} = 4.25$ and 5 pull channels to deliver cold items. When the workload is 15, 20 or 30 respectively, we can find that neither pure push nor pull mode is suitable and ABS performs a hybrid delivery. Even though all channels are the pull mode, we can observe that some items are not on the pull channels yet because the pull workload has reached the upper limit. In this case, ABS will stop its search when it perceives that a channel switch from push to pull does not lead to a better balance.

In Fig. 3(c), most items are cold data and are delivered through 5 pull channels, except the case that workload is 30. Moreover, all the curves are lower than their corresponding curves in Fig. 3(a). This observation shows that given a very high skew access pattern, an increasing workload does not make \bar{t}_{push} go up sharply and \bar{t}_{push} is still lower than \bar{t}_{push}

in a pure push delivery. It is interesting to note from Fig. 3(b) that if access pattern is not relatively skewed, a channel switch does not save too much \bar{t}_{push} when the dynamic workload is heavy. The length of inner histograms with a linear decrease of degree conforms with this observation. In this case, ABS finds that a pure push is preferable when the workload is very heavy.

D. Dynamic Real Traffic Scenario

Fig. 4 presents an attractive scenario considering dynamic access frequency and request arrival rate simultaneously. To simplify a demonstration, we set that access frequency will increase linearly from 10 to 20 per broadcast cycle, and the access pattern is skewed by θ from 1.0 to -0.5. In Fig. 4(a), starting at $\lambda = 10$ and $\theta = 1.0$, ABS finds the best balanced point at $\bar{t}_{push} = 0.78$ when all channels are in the pull mode. In the broadcast ABS detects a traffic change ($\lambda = 12$) and accordingly searches a new balance at $\bar{t}_{push} = 1.75$ with 5 pull channels and 5 push items in one push channel. ABS iteratively repeats the same process, and finally, as $\lambda = 20$ ABS reaches a new balanced point at $\bar{t}_{push} = 12.2$ with 5 push channels. Note that $\bar{t}_{push} = 12.2$ is better than the last two balanced points because its access distribution is near an uniform distribution when θ is close to 0. This agrees with our intuition since a pure broadcast is suitable for a random access traffic.

Fig. 4(b) shows the comparison among pure push, pure pull and ABS delivery modes. Pure push's \bar{t}_{push} is a constant value of 13. Oppositely, pure pull's \bar{t}_{pull} is sensitive to the request arrival rate. When the workload is larger than the available bandwidth, \bar{t}_{pull} becomes unacceptable. For a hybrid delivery mode, the response time is influenced by the access pattern and the request arrival rate. As the case of $\theta = 0.5$ and $\lambda = 20$ that presents a higher and skewed workload, ABS can adaptively determine a better balanced point at $\bar{t}_{push} = 12.2$.

V. CONCLUSIONS

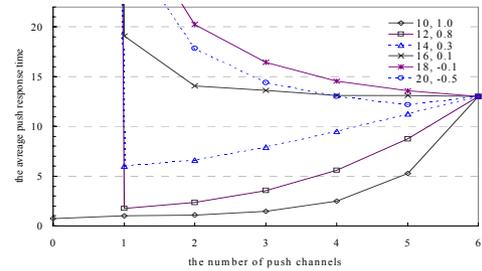
In this paper we propose an adaptive balanced scheme (ABS) which aims at balancing the push and pull mechanisms for a hybrid data delivery in a multi-channel data dissemination environment. We model the adaptation of channel allocation and data classification as a deterministic searching for a local balance from the viewpoint of the access/response fairness. Rather than being dedicated to a specific traffic factor, ABS takes account of multiple traffic factors at a same time. The simulation result has shown the feasibility and the reliability of ABS, especially its adaptiveness to dynamic traffic changes.

ACKNOWLEDGEMENT

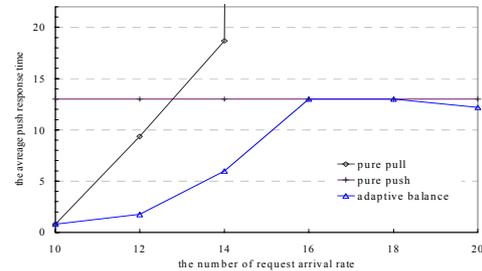
The authors were supported in part by the Ministry of Education Project No. 89-E-FA06-2-4-7 and the National Science Council, Project No. NSC 90-2213-E-002-086, Taiwan, Republic of China.

REFERENCES

[1] S. Acharya, R. Alonso, M. J. Franklin, and S. B. Zdonik. Broadcast disks: Data management for asymmetric communications environments.



(a) ABS searches a balance under a dynamic traffic with multi-factors



(b) ABS balances the mean response time

Fig. 4. ABS is adaptive to both dynamic access pattern and workload simultaneously.

In *Proceedings of the 1995 ACM International Conference on Management of Data*, pages 199–210, May 1995.

[2] S. Acharya, M. J. Franklin, and S. B. Zdonik. Balancing push and pull for data broadcast. In *Proceedings of the 1997 ACM International Conference on Management of Data*, pages 183–194, May 1997.

[3] J. Cai and K.-L. Tan. Tuning integrated dissemination-based information systems. *Data & Knowledge Engineering*, 30(1):1–21, 1999.

[4] Y. Guo, S. K. Das, and C. M. Pinotti. A new hybrid broadcast scheduling algorithm for asymmetric communication systems: Push and pull data based on optimal cut-off point. In *Proceedings of the 4th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile System*, pages 123–130, July 2001.

[5] Q. Hu, D. L. Lee, and W.-C. Lee. Dynamic data delivery in wireless communication environments. In *Proceedings of ER'98 Workshops on Mobile Data Access*, volume 1552 of *Lecture Notes in Computer Science*, pages 218–229. Springer, 1998.

[6] J.-L. Huang, W.-C. Peng, and M.-S. Chen. Binary interpolation search for solution mapping on broadcast and on-demand channels in a mobile computing environment. In *Proceedings of the 10th ACM International Conference on Information and Knowledge Management*, November 2001.

[7] T. Imielinski, S. Viswanathan, and B. R. Badrinath. Data on air: Organization and access. *IEEE Transactions on Knowledge and Data Engineering*, 9(3):353–372, May/June 1997.

[8] K. Kleinrock. *Queueing Systems*, volume I: Theory. Wiley, New York, 1975.

[9] W.-C. Lee, Q. Hu, and D. L. Lee. A study on channel allocation for data dissemination in mobile computing environments. *ACM/Baltzer Mobile Networks and Applications, Special Issue on Resource Management in Wireless Systems*, 4(2):117–129, 1999.

[10] W.-C. Peng and M.-S. Chen. Dynamic generation of data broadcasting programs for a broadcast disk array in a mobile computing environment. In *Proceedings of the 9th ACM International Conference on Information and Knowledge Management*, November 2000.

[11] K. Prabhakara, K. A. Hua, and J. Oh. Multi-level multi-channel air cache designs for broadcasting in a mobile environment. In *Proceedings of the 16th IEEE International Conference on Data Engineering*, pages 167–176, February 2000.

[12] K. Stathatos, N. Roussopoulos, and J. S. Baras. Adaptive data broadcast in hybrid networks. In *Proceedings of the 23rd International Conference on Very Large Data Bases*, pages 326–335, August 1997.