

# Cooperation Strategy for Broadcast Scheduling and Base Station Caching in Hybrid Wireless Broadcast Environment

JING CAI,<sup>†</sup> TSUTOMU TERADA,<sup>†</sup> TAKAHIRO HARA<sup>†</sup>  
and SHOJIRO NISHIO<sup>†</sup>

Recent advances in computer and wireless communication technologies have increased interest in combinations of various techniques such as pull-based data dissemination, data broadcasting, and data caching, as well as investigations into data management in hybrid data delivery systems. This paper proposes cooperative data management based on the Hybrid Wireless Broadcast (HWB) model to provide more efficient data delivery to mobile clients. Our cooperative strategy integrates broadcast scheduling with cache management of the base station, taking heterogeneous and homogeneous access of clients into consideration, by making effective use of HWB data dissemination, i.e., push-based broadcast, pull-based broadcast, and pull-based point-to-point wireless communication. Our simulation results confirmed that the strategy we propose can improve the performance of the system even further.

## 1. Introduction

### 1.1 Background

Various techniques have been developed to improve the performance of wireless information services, such as pull-based data dissemination, data broadcasting, and data caching. Recent advances in computer and wireless communication technologies have increased interest in combinations of these techniques.

There has been some research on hybrid data delivery augmenting pure push-based broadcast, integrated with pull-based broadcast or with pull-based point-to-point data dissemination. Meanwhile, data caching has been widely used as an important technique by many broadcasting systems.

Our previous study proposed a Hybrid Wireless Broadcast (HWB) model<sup>3)</sup>, which can provide flexible and complementary information services at different bandwidths and service ranges, as well as improving the responsibility and scalability of the system. However, it is necessary to investigate cooperative data management so that information can be disseminated more efficiently to mobile clients.

Cooperative broadcast scheduling and cache management have been confirmed to deliver data more efficiently<sup>1),4),8),10),14),15)</sup>. However, most research has investigated integrated control based on push-based broadcast, and cache management has usually been conducted on the

client side. So far, there have been no studies discussing cooperative data management based on hybrid broadcasting environments.

In this paper, we aimed at constructing an effective cooperative strategy under the HWB environment by integrating broadcast scheduling and cache management of the base station, and considering heterogeneous and homogeneous access of clients.

### 1.2 HWB Model

This section briefly introduces the HWB model and its features.

As **Fig. 1** shows, the HWB model is comprised of a broadcast server, numerous base stations, and a large number of mobile clients. The broadcast server broadcasts information over a broad range to global clients with two distinct bandwidths, i.e., a broadband main channel and a relatively narrowband on-demand sub channel. Here, the global clients are those for the entire system. Each base station also has a cache, connects with the broadcast server through the Internet, and serves local clients via the narrowband wireless channel. Here, the local clients are those covered by the services of a base station, which are a subset of global clients. We distinguish between mobile clients as global and local clients to describe the different service ranges of the broadcast server and base stations. In fact, their operations are not different. Mobile clients in the HWB system equipped with a portable terminal such as a PDA can access information from the broadcast server and base stations. Clients send requests to their local base station, while the base sta-

<sup>†</sup> Graduate School of Information Science and Technology, Osaka University

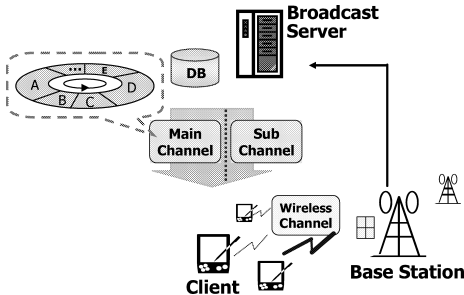


Fig. 1 HWB model.

tions process queries, respond to these by themselves, or transfer them to the broadcast server.

It is worth noting that the three modes of data delivery for the HWB model have different features. The response for the push-based broadcast of the main channel is not affected by the load, i.e., the number of clients queries. However, clients may not be able to acquire a reply quickly even though there is a small number of total queries, since the average response time depends on the volume of broadcast.

The pull-based broadcast of the sub channel can meet the individual requirements of global clients; at the same time, any on-demand response can be accessed by an arbitrary number of clients. Therefore, it is more efficient to respond to queries that require the same data. However, access latency depends on the number of different requests.

In contrast, pull-based wireless communication can meet the individual requirements of local clients by using the base station cache. However, unlike the on-demand sub channel, an on-demand response via the wireless channel cannot be shared due to the point-to-point communication. Hence, access latency depends on the number of queries, no matter whether they require the same item or not.

It would be helpful to determine a more effective system control from the three modes of data delivery for the HWB model, if minimizing the impact of defects in one mode by exploiting the advantages of the others were taken into account.

### 1.3 Broadcast Scheduling and Caching

The server usually broadcasts different items at the same frequency, i.e., flat broadcast, regardless of their relative importance to the clients. By comparison, the server in multi-disks broadcast<sup>1)</sup>, can broadcast items at different frequencies, i.e., important items can be broadcast more often than others. Hot items

are normally placed on fast disks.

Caching, on the other hand, is used to improve performance by keeping frequently accessed or expensive data in memory. The basic problem with cache management is to determine which items should be cached to yield the best response to the clients' requests. This is typically achieved in pull-based systems by caching items with the highest access probability, i.e., probability-based caching. Cost-based PIX caching is more efficient than probability-based caching in push-based multi-disks systems. PIX caching is management that keeps data items with the highest PIX values in the cache, where these values represent the ratio of access probability to the broadcast frequency for the item. Caching strategy in such systems takes the cost of re-acquisition along with the access probability of the items into account. However, we need to clarify whether probability-based caching or PIX caching is more suited to the HWB system.

In addition, caches are generally managed by the client side, where the role of the cache is to reduce response latency for the client itself. In contrast, the HWB system employs a base station cache, which serves the local client community rather than an individual client. Therefore, cache management of the base station aims at improving the average response to the local client community.

It is necessary to take the special role of the base station and cooperative cache strategy with broadcast scheduling into consideration to construct more effective data management of the HWB system,

### 1.4 Overview of the Paper

We propose a cooperative strategy in this paper that takes the complementary features of the HWB data dissemination, and the integration of the base station caching and the broadcast scheduling into account. We investigated the following interrelated issues in an HWB environment by evaluating the performance of the system under different strategies:

- Which would be the optimal way of processing data items with the highest probability of access? Keeping them in the base station cache or broadcasting them at high broadcast frequency?
- Would it be necessary to assign all items in the database to broadcasting? How about only broadcasting some hot items or items with a no Cache-resident?

- What would the most suitable strategy be to attain integrated control of the HWB system?

All the work in this paper was assumed to be based on the HWB model. We also made these additional assumptions:

- Data were read-only, and clients' access patterns were static. We set aside the issue of dynamic access and data updates for future study.
- Clients' access patterns varied according to the base station they belonged to, e.g., location-dependent services.
- The broadcast server and base station had knowledge of the clients' access patterns, which could be precompiled based on past accesses or their profiles.
- The broadcast server was aware of each base station's cache, which could be informed by the base station via the Internet.
- The client had no capabilities for caching. Although this assumption might be too assertive in a real environment, we made it to remove the impact of the client cache, since the aim of this paper was to study the cooperative broadcast scheduling and cache management of the base station.

The rest of this paper is organized as follows. Section 2 presents our method of cooperative control and puts forward several elementary cooperative strategies for the HWB model. In Section 3, we show the evaluation results for the simulation study and give a further discussion on the integrated control of the HWB system. Section 4 introduces related work. Finally, we summarize this paper in Section 5.

## 2. Cooperative Control in HWB

### 2.1 Access Probability

Our cooperative control of the HWB system depends on the clients' access probability, which is distinguished as the Local Access Probability (LAP) of the base station and the Global Access Probability (GAP) of the system. LAP plays an important role in base station caching; while GAP has much more of a relation with broadcast scheduling of the server.

The requests for item  $i$  from base station  $b$  generate the local access probability,  $P_b(i)$ , where  $1 \leq i \leq N$  and  $\sum_{i=1}^N P_b(i) = 1$ . Here,  $N$  denotes the total number of data items in the database. Moreover, the requests for item  $i$  by all base stations are aggregated into the global access probability,  $P(i)$ , which is calculated by

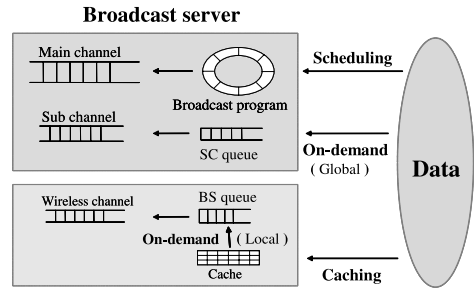


Fig. 2 Control model.

$$P(i) = \frac{1}{M} \sum_{b=1}^M P_b(i),$$

where  $M$  denotes the total number of base stations, and  $\sum_{i=1}^N P(i) = 1$  is also satisfied.

Since clients in different regions normally have heterogeneous access, while clients in the same geographical area often have similar interests in local information, such as location-based information, some items may be hot in one area but cold in others. Taking these into account, we assumed that all the items in the database would be divided into several data groups in accordance with the number of base stations. Requests in base stations have a great tendency ( $k$ ) to issue skewed queries, and a low tendency to send uniform queries. The skewed queries follow Gaussian distribution  $Gau(i, \mu, \sigma)$  with  $\mu$  being the center of hot spot and  $\sigma$  the deviation. The value of  $\mu$  is set as the center of each data group, while the value of  $\sigma$  can be varied to reflect the different skews of the clients' queries. Let  $F_b(i)$  denote the query mode of base station  $b$  for item  $i$ , which is given by

$$F_b(i) = k \times Gau(i, \mu, \sigma) + (1 - k)/N.$$

Hence, the local access probability is estimated by

$$P_b(i) = F_b(i) / \sum_{j=1}^N F_b(j).$$

### 2.2 Cooperative Control

Figure 2 outlines the control model for the HWB system. Generally speaking, data items in the database can be assigned to the schedule of the main channel to be broadcast, be placed in the on-demand sub channel to respond to queries by global clients, or be kept in the base station cache and transmitted on-demand by the wireless channel to respond to queries by local clients. Although it is essential to choose

the optimal means of processing, it is unrealistic to assume the data organized in the system will comply with the individual access patterns of all clients. Therefore, we focused on processing items with high access probability (i.e., hot items), for which to choose the optimal means.

Base stations, on one hand, provide services to the local client community, which normally pays more attention to local information. We tend to increase the cache hits ratio of base stations to provide a low-latency response to local clients. Probability-based caching is good at increasing cache hits ratio if the hottest local items are kept in the base station cache, since it is more suited to the pull-based system, while the HWB model provides two pull-based modes of data delivery. Our work concentrated on skewed access of clients, while multi-disks scheduling is a good program for non-uniform access. Consequently, we employed probability-based caching and multi-disks broadcasting as the basic processing mode in our cooperative control of the HWB system.

The broadcast server, on the other hand, serves clients in global areas. If the hottest local items are retained in the base station's cache, as clients that reside in different regions normally have heterogeneous access interests, these items should be pushed onto a slower disk. Moreover, chopping off part of the broadcast data effectively increases the available bandwidth. We attempted to remove the cached items from the broadcast program to shorten the broadcast cycle, i.e., only the items with no Cache-resident, no\_Ca items, were to be broadcast. It was because items that are disappeared in the main channel in the HWB environment can be pulled via an on-demand wireless channel if there is the required cache at the local base station, or they can be pulled through the on-demand sub channel.

Furthermore, the broadcast server needs to be aware of the base station cache to cooperate with broadcast scheduling and carry out joint control. We considered modifying the LAP and GAP values to reflect the base station cache by means of the two-phase access probability of the HWB system, since multi-disks scheduling is conducted based on the clients' access probability. We therefore designed the following procedure to implement our method of cooperative control.

- (1) Retain the hottest items with the largest estimated LAP values in the base station

cache,

- (2) Modify the LAP values of all the items in the base station cache,
- (3) Aggregate the GAP by using the modified LAP of each base station, and
- (4) Construct a multi-disks broadcast program with the modified GAP.

Note that the modified LAP/GAP values are different from the estimated LAP/GAP values; the latter denotes the clients' access probability, while the former indicates the distribution for the base stations' cache along with client access. As a result, once an item is cached, its LAP and GAP values will be modified. The cached items are impossible to push to the fastest disk by greatly decreasing the corresponding LAP and GAP values. Moreover, if the modified GAP values of some items become zero, these items will no longer be broadcast. Multi-disks scheduling with our cooperative control ranks all the items with the modified GAP values instead of those estimated to generate a multi-disks broadcast program.

### 2.3 Cooperation Strategies

To find what the most efficient cooperative control for the HWB system would be, we considered several different processes for base station caching and broadcast scheduling. We considered three patterns for base station caching: the items with the largest GAP values or those with the largest LAP values were given the highest priority to be cached in the base station, or PIX caching was employed (abbr. as GAP/LAP/PIX caching, respectively). We considered three patterns for scheduling of multi-disks broadcast: all the items in the database to be broadcast, or only part of the items, hot items or items with no Cache-resident, to be broadcast (abbr. as all/hot/no\_Ca scheduling respectively). We present the following six cooperation strategies by combining the processes of base station caching and broadcast scheduling.

#### GAP-no\_Ca

The hottest global items with the largest GAP values are kept in the cache of each base station with this strategy; hence all base stations have an identical cache. In addition, the LAP values for all the cached items are set to zero, i.e.,

$$P_b(i) = 0 \mid i \in \text{Cache}, b = 1 \sim M,$$

where  $\text{Cache} = \text{Cache}_1 = \dots = \text{Cache}_b = \text{Cache}_M$ . Here  $\text{Cache}_b$  denotes the set of items kept in the cache of the base station  $b$  ( $1 \leq$

$b \leq M$ ), while *Cache* denotes the total set of items kept in all base station caches. Therefore, the GAP values for those cached items also become zero. That is to say, all the items identically cached at each base station will no longer broadcast.

#### GAP-all

The cache management for this strategy is the same as that for GAP-no.Ca; in short, all the base stations identically cache the hottest global items. However, all the items in the database will be broadcast, no matter whether they are cached or not.

#### LAP-no\_Ca

All base stations keep the hottest local items with the largest LAP values in their cache with this strategy. Distinct from the above two strategies, cache of each base station is different, because heterogenous access of clients in different regions results in dissimilar LAP. The LAP values, on the other hand, for all items in any base station cache are set to zero, i.e.,

$$P_b(i) = 0 \mid i \in \text{Cache}_b, b = 1 \sim M,$$

where  $\text{Cache} = \text{Cache}_1 \cup \dots \cup \text{Cache}_b \cup \text{Cache}_M$ . Therefore, the GAP values for those items in any base station also become zero. In other words, once items are cached in any base station, they will no longer be broadcast. Items, which disappeared from the main channel and are requested by clients from a base station without the required cache-resident, will be pulled via the on-demand sub channel.

#### LAP-all

This strategy has the same cache management as LAP-no.Ca, and also lets each base station keep the hottest local items in its cache. However, it only modifies the LAP values for items residing in its own base station cache to zero, i.e.,

$$P_b(i) = 0 \mid i \in \text{Cache}_b, b = 1 \sim M.$$

Hence, although the GAP values of the cached items are greatly decreased, they do not become zero. All the items in the database will be broadcast.

These four strategies can be generated according to the procedure described in the Section 2.2. In contrast, the following two strategies first generate a multi-disks broadcast program with the estimated GAP, and then PIX caching is adopted. The reason for this is that PIX caching calculates the PIX value for each item by using the determined broadcast frequency. Here, the PIX value for an item is the ratio of its local access probability, i.e., the LAP

value to its broadcast frequency.

#### PIX-all

This strategy follows the traditional ‘‘Broadcast Disks’’ technique. All items in the database are broadcast with an adequate frequency; the hottest global items with the largest GAP values are pushed to the fastest disk. Items with the highest PIX values, on the other hand, are cached in the base station.

#### PIX-hot

The broadcast scheduling and base station caching with this strategy are similar to PIX-all. However, only part of the items in the database are broadcast. The cold items with the smallest GAP values are removed from the broadcast program, i.e., only the hot items for the system are broadcast.

### 3. Simulation Experiments

We conducted a set of simulation experiments to examine what influence these stated strategies had on the performance of the system.

**Table 1** summarizes the default parameter settings for the experiments. The bandwidth for the main channel, sub channel, and wireless channel were assumed to be 100 Mbps, 10 Mbps, and 5 Mbps, due to a satellite communication system with a high data rate above 100 Mbps having been developed, and the data rate for the 3G mobile service being a few hundreds of Kbps and several Mbps. In addition, there were 10,000 items in the database and all these had an equal size of 100 KB. There were 10 base stations and each had a cache with 200 items.

There were three disks for multi-disks scheduling and their broadcast frequency was four, two, and one. The size of the fastest disk was identically set to 500 items, while

**Table 1** Parameter settings.

Parameters	Values
Database Size [Data Items]	10,000
Data Item Size [KB]	100
Number of Base Stations	10
Cache Size of BS [Data Items]	200
Main Channel Bandwidth [Mbps]	100
Sub Channel Bandwidth [Mbps]	10
Wireless Bandwidth [Mbps]	5
Time Slot [D/Bm]	100,000
Query Interval [ms]	50~1,500
Data Group Size [Data Items]	1,000
Query Tendency [%]	80
Deviation for Gaussain	150
Number of Disks	3
Broadcast Frequency of Disk <sub>1,2,3</sub>	4; 2; 1
Size of Disk <sub>1,2</sub> [Data Items]	500; 1,000

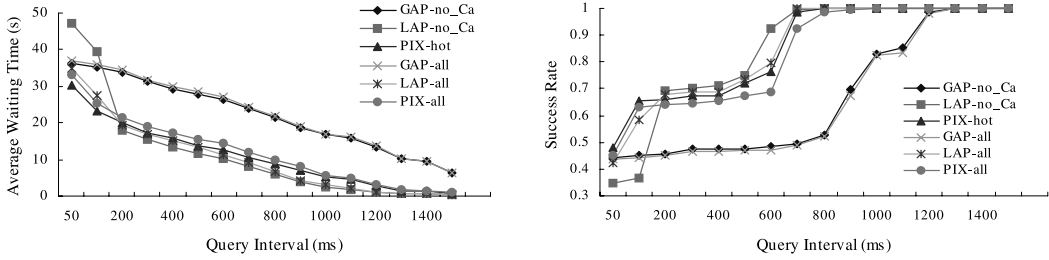


Fig. 3 Impact of query interval.

the medium disk was set to 1,000 items to enable objective evaluation. The size of the slowest disk and the broadcast cycle, on the other hand, depended on the strategy. Comparing the broadcast cycle for all strategies, LAP-no\_Ca, together with PIX-hot, was the shortest.

Our experiments evaluated the performance of all strategies by varying some important factors: the average intervals of queries, the access patterns of clients' requests, the sizes of base station caches, and the bandwidths for data delivery. The performance metrics were the average waiting time and the success rate of queries.

### 3.1 Impact of Query Interval

We evaluated the performance of the system under different query intervals in the first experiment. Figure 3 shows that as the query interval increases, the performance of all the strategies increases in terms of the average waiting time as well in the success rate of the queries, due to the decreasing access load. Meanwhile, LAP-no\_Ca and LAP-all outperform the other strategies in the main phase of the evaluation (i.e. at query intervals longer than 200 ms); in particular, LAP-no\_Ca always performs the best.

The reasons for all these are as follows. These two strategies all maintain the hottest local items in the base station cache, which is more suited to providing low-latency response to local clients who normally have homogeneous access to local information. LAP-no\_Ca, moreover, removes all cached items from the broadcast, and utilizes the on-demand channel to pull items that have disappeared from the main channel. Therefore, LAP-no\_Ca greatly shortens the broadcast cycle and makes more efficient use of the three bandwidths and thus performs the best.

However, a heavy access load (e.g., query intervals of less than 200 ms) causes a dramatic drop in the performance of LAP-no\_Ca. The reason for this is that a great number of queries

is responded to by point-to-point wireless communication, which is far beyond the bandwidth of the wireless channel. By comparison, there is less impact on the performance of LAP-all, in which all items are broadcast, while cached items with a lower broadcast frequency, such as those in a congested wireless channel can be relieved to some degree by the broadcast in the main channel.

PIX-hot and PIX-all, on the other hand, both perform better under heavy query loads. The best explanation is that numerous queries can be responded to better by frequently broadcasting the hottest items. PIX-hot, in particular, performs the best by cutting some cold items from the broadcast to shorten the broadcast cycle.

GAP-no\_Ca and GAP-all always perform the worst, due to the lowest cache hits ratio. Only when the access load is extremely heavy, does it perform better. This is because a low cache hits ratio cannot impose a greater load on the wireless channel, while most cache-missed hot items are frequently broadcast. Similarly, GAP-no\_Ca is a little better than GAP-all due to the shorter broadcast cycle.

### 3.2 Impact of Access Pattern

Next, let us evaluate the impact of access patterns through two interrelated factors: access tendency and the query deviation of skewed access. The greater the access tendency and the smaller the query deviation in queries, the more skewed the access. We conducted experiments on these.

Figure 4 plots the access tendency under normal query deviations and system conditions. The results reveal that the performance of each strategy is proportional to the access tendency, but they have different slopes. Generally speaking, strategies with LAP caching perform better than those with PIX caching, and both of them are drastically better than the strategies with GAP caching. Strategies broadcast-

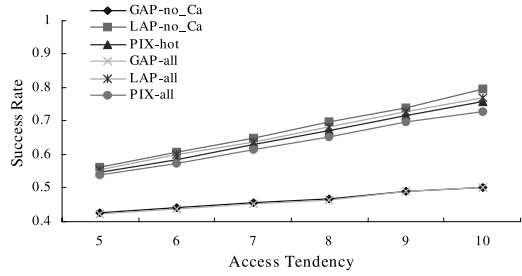
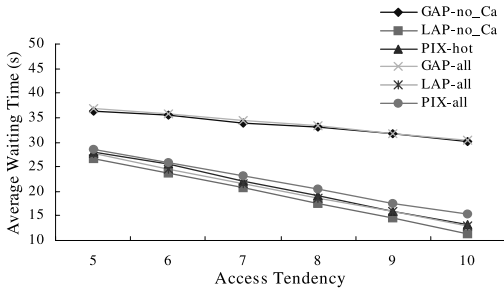


Fig. 4 Impact of access tendency.

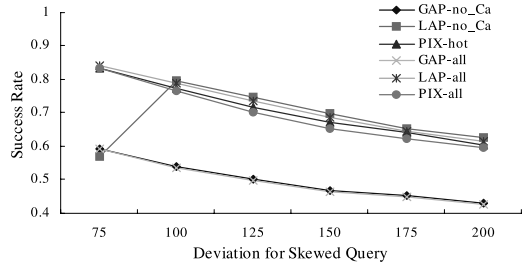
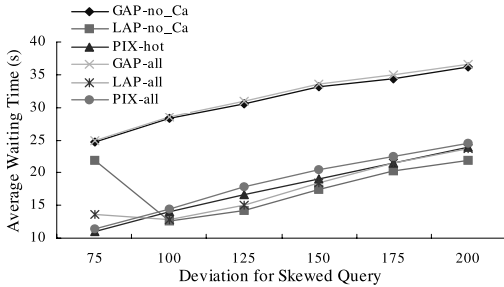


Fig. 5 Impact of query deviation.

ing part of the items (i.e., no\_Ca scheduling and hot scheduling), on the other hand, perform relatively better than corresponding ones with all scheduling. Therefore, it is reasonable to conclude that LAP-no\_Ca has the best performance by its adopting LAP caching and no\_Ca scheduling. This is because under normal access conditions, a base station with LAP caching can provide more efficient local services to clients; moreover, no\_Ca scheduling can not only shorten the broadcast cycle, but can improve the efficiency of bandwidth use.

In addition, Fig. 5 plots the query deviations under default access tendencies. We can see that LAP-no\_Ca outperforms the other strategies under ordinarily skewed access (i.e., query deviations above 100), by taking advantage of a higher cache hits ratio and more efficient bandwidth use.

The performance of LAP-no\_Ca rapidly decreases when the query deviations become extremely skewed (e.g., only about 75 items), i.e., a large number of queries concentrate on a quite small number of the hottest items. This is because most of the hottest items are only stored in the base station cache, such that a large number of point-to-point responses causes the wireless channel to become very congested. Strategies with PIX caching, on the contrary, broadcast the required hottest items more of-

ten, where frequent broadcasts can be shared by an arbitrary number of clients; hence they perform better, and PIX-hot with the shortest broadcast cycle performs the best.

### 3.3 Impact of BS Cache Size

Let us now examine the impact of cache size at the base station. As we can see from Fig. 6, when there are less than 300 items in a base station cache, i.e., an amount relatively smaller than that of the access range of local clients, the performance of all strategies markedly increases as the size of the cache increases. This is because the number of cache hits increases with the cache size. Meanwhile, LAP-no\_Ca has the highest cache hits and the shortest broadcast cycle, and thus works the best.

However, when the size of the base station cache approaches 300 items, the differences in performance between almost all strategies are extremely small. Moreover, when there are more than 300 items in the cache, the performance of LAP-no\_Ca drops sharply, whereas this has little influence on the other strategies. The reason is that LAP-no\_Ca stores the hottest local items in the base station cache, and once these items are kept in the cache they can no longer be broadcast. When the cache at the base station is so large as to store most of the hottest local items, the load imposed on the wireless channel becomes too heavy. Mean-

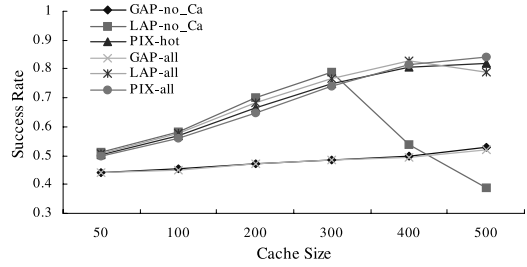
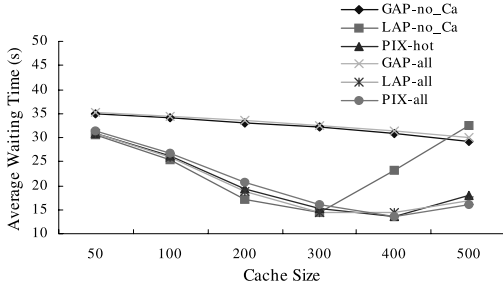


Fig. 6 Impact of base station cache size.

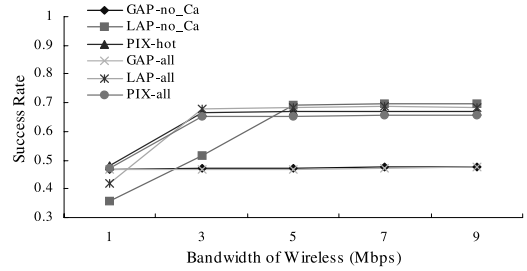
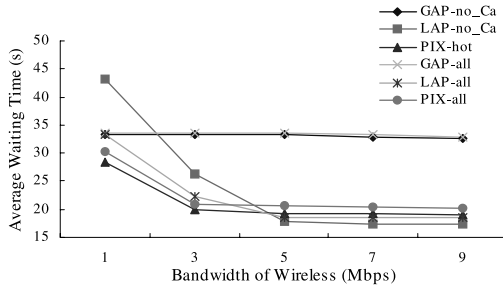


Fig. 7 Impact of bandwidth of wireless channel.

while, the bandwidth utilization of the main channel becomes too low, due to too many items removed from the broadcast. PIX-all, in contrast, performs the best with all large caches, even better than PIX-hot. Although they both push the hottest items to the fastest disk, the processing for PIX-hot to remove that many items from the broadcast lowers the utilization of the main channel, whereas the increasing cache size for PIX-all just expands the number of cache hits but does not debase the efficiency of the main channel.

### 3.4 Impact of Pull Bandwidth

We investigated what impact pull bandwidth had on both the wireless channel and sub channel. As we can see from Fig. 7, when the bandwidth of the wireless channel is very limited (e.g., 1 Mbps), LAP-no\_Ca performs the worst, while PIX-hot performs the best. The reason is that the narrow wireless bandwidth limits the strategies with LAP caching to utilize their higher cache hits ratio, but this has a less influence on those with PIX caching. Once the bandwidth increases (above 5 Mbps), these strategies with LAP caching perform better than those with a lower cache hits ratio. LAP-no\_Ca, in particular, performs the best with a sufficient amount of bandwidth. We can also see from Fig. 8 that the performance of all strategies increases as the bandwidth of

the sub channel increases. However, when the bandwidth becomes extremely large, the differences between almost all strategies become very small, since the bandwidth of the sub channel is so huge that all the strategies mainly use the sub channel to respond to queries.

### 3.5 Special Conditions

Finally, we undertook some supplementary experiments to evaluate the performance of strategies under special conditions, e.g., mobile clients using high-speed wireless communications, where base stations had a sufficient amount of cache. The bandwidth of the wireless channel in these experiments was the same as that of the main channel, i.e., 100 Mbps; the cache at the base station contained 800 items; while the other parameters were the same as in the previous experiments. Figure 9 shows that LAP-all and PIX-all outperform the others under heavy access loads. This is because both of them broadcast all items and have a high cache hits ratio so that they could make more effective use of the broadband main channel as well as the broadband wireless channel. Comparing these two strategies, the differences between LAP caching and PIX caching are small, because their cached items are identical due to the sufficient amount of cache capacity. LAP-no\_Ca, on the other hand, only approaches these two strategies when access load



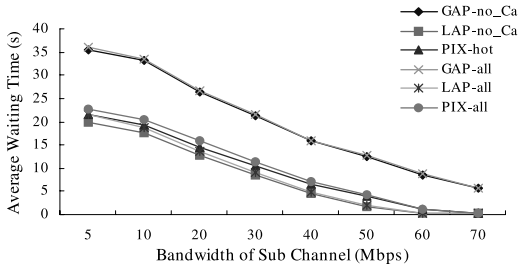


Fig. 8 Impact of bandwidth of sub channel.

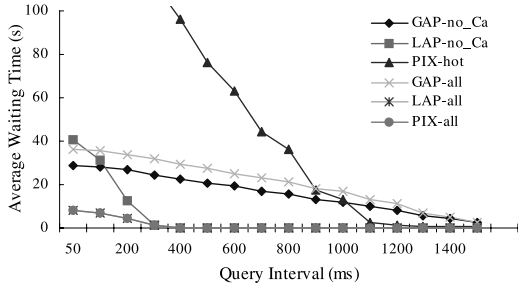
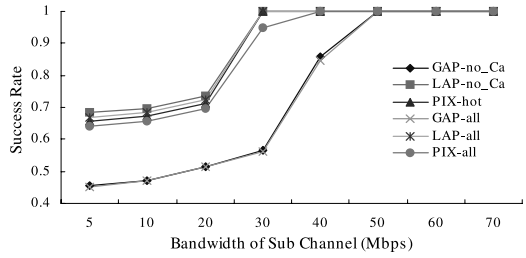
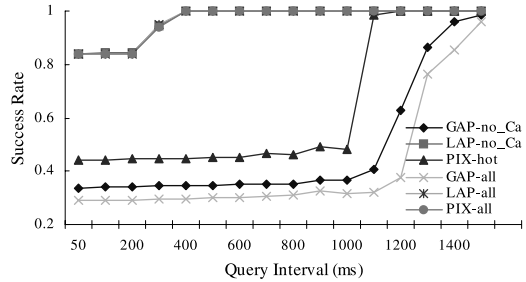


Fig. 9 Performance under special conditions.



is not that heavy. The reason is that the requested items could quickly be responded to by either the broadband main channel or the broadband wireless channel.

3.6 Discussion

We could provide answers to the questions in Section 1.4 and arrive at conclusions for integrated control in the HWB system by comparing how well the six strategies performed, which employed different processes for base station caching and broadcast scheduling.

One consideration is that maintaining items with the highest local access probability in the base station cache, to acquire the highest cache hits ratio, is more effective than broadcasting them with higher broadcast frequency, and much better than caching the hottest global items under normal conditions. This is suitable for heterogeneous access of clients in different areas, and can provide low-latency service to the clients in the same region with homogeneous access interests. Our experiments confirmed that strategies maintaining the hottest local items in the base station cache normally performed the best.

Another consideration is that removing some unnecessary items from the broadcast program is better than broadcasting all the items in the database. This can shorten the broadcast cycle to increase the available bandwidth for the main channel, and can also make more effective use

of the on-demand channel to pull items that have disappeared from the main channel. Our experiments confirmed that strategies adopting this process performed better than strategies that broadcast all items.

However, if there is no limit to using these two processes, when the increased number of cache hits is beyond the bandwidth of the wireless channel, and when too many items are removed from the main channel, system performance will worsen, due to the increasingly heavy load on the wireless channel and the lower bandwidth utilization by the main channel.

The most suitable strategy for cooperative control of the HWB system depends on access situations.

LAP-no\_Ca is the most suitable strategy under ordinary conditions, i.e., when load on the system is ordinarily heavy, clients' access is ordinarily skewed, and the bandwidth for the wireless channel is not too small. The reason for this is that LAP-no\_Ca makes the most efficient use of the three modes of data delivery for the HWB model under normal conditions, by maintaining the hottest local items in the base station cache, removing the cached items from broadcast, pushing the hottest global items to the fastest disk, and pulling items that have disappeared from the broadcast. This strategy is suited to providing location-based information services, such as those for traffic reports, which

are closest to the available service.

PIX-hot is the best choice when load on the system is extremely heavy, client access is extremely skewed, while the bandwidth of the wireless channel is rather small. The reason for this is that PIX-hot makes the most advantageous use of bandwidths under extreme conditions, by broadcasting the hottest global items more often, maintaining the most cost-efficient data in the base station cache, and removing unnecessary cold items from the broadcast. This strategy is suited to providing popular information services in hot time, such as those required for reporting on presidential elections, or sports scores like at the Olympic Games.

LAP-all is the most robust of all the strategies. This is because it performs somewhat better under various conditions than the others, by maintaining the hottest local items in the base station cache, pushing the cached items to a slower disk, and leaving the hottest global items in the fastest disk. This strategy is suited to providing information services to mobile clients under uncertain access conditions.

#### 4. Related Works

Recent advances in computer and wireless communication technologies have increased interest in combinations of pull-based data dissemination, data broadcasting, and data caching. There has been some research on integrating push-based broadcast with pull-based broadcast or with pull-based point-to-point data dissemination<sup>2),6),11),13)</sup>. Also some researchers have adopted data caching in broadcasting systems<sup>1)~4),6),8)~10),12),14),15)</sup>. Meanwhile, research on data management in broadcasting environments has received more attention. Many researchers have independently investigated broadcast scheduling and cache management<sup>5),7),9),11)~13)</sup>. Some other studies, in contrast, have paid more attention to the joint control of broadcast scheduling and caching<sup>1),4),8),10),14),15)</sup>.

Broadcast Disks are well known for integrating multi-disks scheduling with cost-based PIX caching<sup>1)</sup>. The relationship between broadcast scheduling and caching in broadcast disk systems was further explored by Liberatore<sup>10)</sup>. Additionally, Yajima, et al. investigated broadcast scheduling and suitable caching policies by taking the correlation between data items into account<sup>15)</sup>; while Kim, et al. discussed a joint approach by considering the structural

relation between information items through a linked data model<sup>8)</sup>. Their common failing was that they did not take heterogeneous or homogeneous access by clients into account.

Su, et al., on the other hand, proposed an integrated method to simultaneously produce a broadcast schedule and a scheme for prefetch by considering heterogeneous access by clients<sup>14)</sup>. However, they did not consider utilizing data caching to provide low-latency services to clients with similar access patterns. Additionally, Ercetion, et al. addressed joint issues in a two-stage satellite-terrestrial wireless broadcast system by considering the lower average latency of local clients<sup>4)</sup>, whose motivation was similar to ours. However, they did not support uplink channels or on-demand responses. They suggested further work needed to be done to consider hybrid systems.

These studies revealed that joint design of broadcast scheduling and cache management can provide more efficient data delivery. However, they were mostly based on push-based broadcasting, and managed caches on the client side. To the best of our knowledge, no studies have discussed cooperative broadcast scheduling and cache management based on hybrid broadcasting environments, or have taken heterogeneous and homogeneous access of clients into account.

#### 5. Conclusion

In this paper, we investigated cooperative control in the HWB system, taking heterogeneous and homogeneous access of clients into consideration, by integrating broadcast scheduling and cache management of the base station. We also designed several joint strategies to evaluate efficiently integrated control. Our simulations revealed that the performance of the system could be improved even further with suitable cooperative control. We also confirmed that our cooperative strategy usually performed better, by effectively using the HWB data dissemination and by taking advantage of the base station cache cooperatively with broadcast scheduling.

Unfortunately, we had to leave the dynamic control of the HWB system for future study, because the access patterns and query loads of the system dynamically change in real environments. Although the current work was conducted under static conditions, it would be helpful to design appropriate control for a dy-

dynamic systems. We also have an interest in expanding our study in the future to further explore the system's coordination with clients' caches.

**Acknowledgments** This research was supported in part by Grant-in-Aid for Scientific Research (B) (15300033) and on Priority Areas (18049050), and by the 21st Century Center of Excellence Program of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

## References

- 1) Acharya, S., Alonso, R., Franklin, M. and Zdonik, S.: Broadcast disks: Data management for asymmetric communication environments, *Proc. ACM SIGMOD'95*, pp.199–210 (1995).
- 2) Acharya, S., Franklin, M. and Zdonik, S.: Balancing push and pull for data broadcast, *Proc. ACM SIGMOD'97*, pp.183–194 (1997).
- 3) Cai, J., Terada, T., Hara, T. and Nishio, S.: A query processing method for hybrid wireless-broadcast networks, *Proc. MDDS'05*, pp.1078–1082 (2005).
- 4) Ercetin, O. and Tassiulas, L.: Push-based information delivery in two stage satellite-terrestrial wireless systems, *IEEE Trans. Comput.*, Vol.50, No.5, pp.506–518 (2001).
- 5) Guo, Y., Pinotti, M.C. and Das, S.K.: A new hybrid broadcast scheduling algorithm for asymmetric communication systems, *ACM SIGMOBILE MC2R*, Vol.5, No.3, pp.39–54 (2001).
- 6) Hu, Q.L., Lee, D.L. and Lee, W.-C.: Performance evaluation of a wireless hierarchical data dissemination system, *Proc. Mobicom'99*, pp.163–173 (1999).
- 7) Hu, C.L. and Chen, M.S.: Adaptive multi-channel data dissemination: support of dynamic traffic awareness and push pull time, *IEEE Transactions on Vehicular Technology*, Vol.54, No.2, pp.673–686 (2005).
- 8) Kim, J.D. and Kim, C.K.: Response time analysis in a data broadcast system with user cache, *Telecommunication Systems*, Vol.22, No.1–4, pp.119–1390 (2003).
- 9) Kim, S.S., Yang, S.O. and Lee, S.K.: Maintaining mobile transactional consistency in hybrid broadcast environments, *Acta Informatica*, Vol.41, pp.65–81 (2004).
- 10) Liberatore, V.: Caching and scheduling for broadcast disk systems, *Proc. 2nd Workshop on Algorithm Engineering and Experiments (ALENEX '00)*, pp.15–28 (2000).
- 11) Lin, C.W., Hu, H. and Lee, D.L.: Adaptive realtime bandwidth allocation for wireless data delivery, *Wireless Networks*, Vol.10, pp.103–120 (2004).
- 12) Seifert, A. and Scholl, M.H.: A multi-version cache replacement and prefetching policy for hybrid data delivery environments, *Proc. VLDB'02*, pp.850–861 (2002).
- 13) Stathatos, K., Roussopoulos, N. and Baras, J.S.: Adaptive data broadcast in hybrid networks, *Proc. VLDB'97*, pp.326–335 (1997).
- 14) Su, C.J. and Tassiulas, L.: Joint broadcast scheduling and user's cache management for efficient information delivery, *Wireless Networks*, Vol.6, No.4, pp.279–288 (2000).
- 15) Yajima, E., Hara, T., Tsukamoto, M. and Nishio, S.: Scheduling and caching strategies for correlated data in push-based information systems, *Proc. ACM SAC'01*, pp.504–510 (2001).

(Received May 8, 2006)

(Accepted November 2, 2006)

(Online version of this article can be found in the IPSJ Digital Courier, Vol.3, pp.30–41.)



**Jing Cai** is currently pursuing her Ph.D. at the Graduate School of Information Science and Technology of Osaka University in Japan. She received her B.E. from Dalian Maritime University in 1990 and her M.E. from Wuhan University of Technology in 1993, both in China. Her research interests include broadcasting computing, information system access control, and e-Learning. She is a student member of IPSJ and DBSJ.



**Tsutomu Terada** is a lecturer at the Cybermedia Center of Osaka University. He received his B.E., M.E., and Ph.D. from Osaka University in 1997, 1999, and 2003. He has been an Assistant Professor at the same center since 2005. Dr. Terada has previously worked on active database technologies, and he is currently investigating the application of active database techniques within the context of smart object systems. He is a member of IEEE, IEICE, and IPSJ.



**Takahiro Hara** received his B.E., M.E., and D.E. in Information Systems Engineering from Osaka University in Japan, in 1995, 1997, and 2000. He is currently an Associate Professor at the Department of Multimedia Engineering of Osaka University. His research interests include distributed database systems in advanced computer networks, such as high-speed networks and mobile computing environments. Dr. Hara is a member of ACM, IEEE, IEICE, IPSJ, and DBSJ.



**Shojiro Nishio** received his B.E., M.E., and Dr.E. from Kyoto University, Japan, in 1975, 1977, and 1980. He was with the Department of Applied Mathematics and Physics of Kyoto University from 1980 to 1988. In October 1988, he joined the faculty of the Department of Information and Computer Sciences of Osaka University. He became a full professor at the Department of Information Systems Engineering of Osaka University in August 1992. He has been a full professor at the Department of Multimedia Engineering at the same university since April 2002. From April 2000 to July 2003, he served as the founding director of the Cybermedia Center of Osaka University. He has been serving as the dean of the Graduate School of Information Science and Technology of this university since August 2003. His current research interests include database systems and multimedia systems. Dr. Nishio has served on the Editorial Board of *IEEE Transactions on Knowledge and Data Engineering* and *ACM Transactions on Internet Technology*, and is currently involved with the editorial board of *Data and Knowledge Engineering*. He is a fellow of IEICE and IPSJ, and he is a member of six learned societies, including ACM and IEEE.

---