

Data Broadcasting in Wireless Environment Using Spectral Approach

Chien-Yi Wu and Guang Ming Wu
Department of Information Management, NanHua University

Abstract. Data broadcasting is an efficient approach for disseminating data in the mobile computing environment. To shorten the mobile clients' average access time, many strategies have been proposed to schedule the order of data items on the broadcast channel. However, the issues of a client accessing multiple data items on the broadcast channel are less discussed. In this paper, we propose an efficient approach using eigenvector to improve the quality of data placement for accessing multiple data items in wireless environment. Experiments are performed to compare to the previous scheduling approach [8] and then show the promise of our data scheduling algorithm.

Keywords: data placement, data broadcasting, eigenvector.

1 INTRODUCTION

With the rapid advances of wireless technologies, a user can now access data from wireless networks any time any where through a mobile device. Differing from conventional wire-line networks, wireless networks have some properties. First, the bandwidth of the wireless network and the energy needed for portable devices are limited. It is noted that the mobile clients use small batteries for their operation without directly connecting to any power source, and the bandwidth of wireless broadcast channel is lower than that of wire-line networks. Second, the wireless environment is asymmetric, that is, considering the power consumption, sending data is more costly than receiving data for a portable device. Due to above properties, the traditional request-response system is no longer suitable for data dissemination in the wireless environment. Hence, *data broadcasting* is becoming an interestingly attractive data dissemination method for a large amount of mobile clients in a mobile computing environment.

In the broadcast-based information system [1], [7], [8], [15], a server periodically broadcasts data items to mobile clients according to a predetermined broadcast program on a single broadcast channel. To retrieve data items of interest,

the mobile clients need to wait for the appearance of the data items on the broadcast channel instead of sending requests to the information system. We summarize advantages of data broadcasting in the following:

Power conservation: this is due to the fact that mobile clients need not explicitly send data items to the server.

High scalability: The high scalability is achieved since the system performance is independent of the number of mobile clients.

High bandwidth utilization: data items of high interest can be received by multiple mobile clients by on transmission on the broadcast channel.

Fig. 1 illustrates how the server broadcasts data items on a single channel with the pure-push-based technique. Firstly, the server gathers some histories in a period and generates a broadcast program according to the access frequencies of the mobile clients. And then the server periodically broadcasts data items and the mobile clients just passively listen to the broadcast channel to retrieve their desired data items without making any requests.

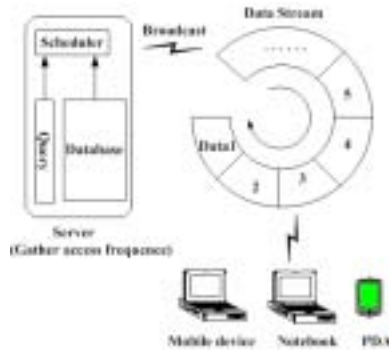


Fig. 1. A push-based system.

The repetition allows the server to save wireless bandwidth. The major advantage of this technique is that all of the mobile clients can access data items on the broadcast channel at the same time without increasing the server workload. However, the limitation of the pure-push-based technique is that mobile clients can only sequentially access data items of interest appearing on the broadcast channel.

In the mobile computing environment, *tuning time* and *access time* are two important performance metrics for data broadcasting approach. *Tuning time* is defined to be the amount of time spent by the mobile client listening to the broadcast channel to download a required data item. According to [15], the power consumption of a mobile device to download the required data items depends on the tuning time. To reducing tuning time, most studies used index techniques such as distributed index [15], hashing [16] and signature [12]. It is advantageous to use indexed data organization to broadcast data over wireless channels so that the mobile clients can be guided to the data item of interest efficiently and only need to be actively listening to the broadcasting channel when the relevant index or data item is present. The study in [6] also developed an algorithm to determine the optimal order for sequential data broadcasting with a given index tree.

Access time is defined to be the time elapsed since the client submitting its request for a data item to the time when all required data items are received by the client. *Access time* is frequently used to evaluate if the mobile clients can get their desired data item immediately. In general, given

access frequency of data items, the studies [1], [7], [8], [17], [20], [22] made efforts in the order of broadcasted data item to minimize the clients' access time as much as possible. Acharya et al. [1] proposed the concept of broadcast disks for disseminating data items in a broadcast environment in which all data items on a single broadcast channel were partitioned into several groups such that the groups containing data items with higher access frequencies had shorter broadcast periods. As a result, the average access time decreases. The initial studies of broadcast disks focused on the performance of the mechanism when the data being broadcast did not change. The performance of broadcast disks was further improved in [2], [3], [4]. They extended their studies to incorporate the impact of updates and the way of disseminating. The scheduling method [20] constructed the broadcast schedule by using the stochastic model. It considered the access frequencies of data items and controls their delivery intervals. In [22], they designed a broadcast schedule algorithm to minimize the wait time and also considered different clients may listen to different number of broadcast channels. However, all these methods considered the case that a query can access only one data item, and did not consider the case for a query accessing more than one data item.

The above techniques address some client issues such as cache management [1], [4], [13], [14], [21] and prefetching [4] and also some server issues such as allocation of data [1], [8], [17] for broadcasting over a single channel. In [34], Su and Tassiulas proposed an efficient broadcast schedule by taking both the access probability and the user caching policy into consideration. Hung et al. [14] cached the indices instead of caching data items to reduce the response time and power consumption of the mobile clients. They designed two index-caching policies to reduce both the tuning time and the access time for accessing data items in a mobile computing environment. *The lower level index first (LLIF)* policy tended to cache the low level index nodes so as to minimize the tuning time. In contrast, *the cut plane first (CPF)* policy tried to cache a cut-plane of the index tree so as to optimize the utilization of the cache. Hung et al. [13] further

considered the problem about the out-of-date of a cached data item, they proposed a data reaccess scheme to allow a mobile client to correctly reaccess its cached data items while the server inserts data items into or deletes data items from the broadcast structure in the course of data broadcasting.

In this paper, we make effort in generating a broadcast program to decide the order of the broadcasted data items such that the mobile clients can access their desired data items with a lower access time. Moreover, in general, a client may request multiple data items simultaneously. For example, a mobile client wants to know weather forecast of Taipei, Chiayi, and Kaohsiung at the same time. In previous studies, this mobile client needs to issue three queries to acquire three weather statuses individually. Besides, the server schedules data items without considering the relationship between them. Therefore, the access of multiple data items in a query is an important issue in mobile computing environments.

The query optimized issue is further considered in [5], [7], [8], [17]. In [5], the method for finding the optimal broadcast program for two dependent files was proposed. In Chung and Kim [7], a broadcast schedule method called *Query Expansion Method (QEM)* to minimize the average access time was proposed. In [17], an efficient algorithm was proposed to determine the placement of the data on the broadcast channel such that frequently co-accessed data items are not only allocated closed to each other, but also in a particular order which optimizes the performance of query processing. In [8], after sorting all queries according to their corresponding access frequencies, it constructed the broadcast schedule by appending each query's data set in a greedy manner. During the process of expanding, they still maintain the order of data items which are previously expanded and shorten the distance of data items in the same query which has higher frequency as much as possible. We assume the system environment as follows:

The server broadcasts data items via pure-push-based technique.
The data items are broadcasted with a single broadcast channel.

A query that the mobile clients request must contain at least two data items.

Each data item appears exactly once on the broadcast channel (we can call this broadcast approach as uniform broadcasting).

All mobile clients only listen the broadcast channel continuously to retrieve their desired data items.

In the following, we will introduce the broadcast schedule problem and its measure method, and use an eigenvector approach which is to solve many ordering problem widely [9], [10] to generate an efficient broadcast schedule. We first transform the total queries of the mobile clients into a graph form, and initially use the order after sorting the coordinates of eigenvector by computing the nonzero smallest eigenvalue as our broadcasted data order. Then we revise the graph with new weighted value iteratively and select the shortest average access time among 100 times in every experiment. Experiments are performed to compare the previous approach [8] with our approach. The results show that our broadcast schedule generated by eigenvector approach outperforms the broadcast schedule in a greedy manner [8].

The remaining of this paper is organized as follows. In Section 2, we describe the problem about scheduling the data items on the broadcast channel. We propose a scheduling method with some illustrative examples in Section 3. In Section 4, we compare the performance between our method and the previous method. Finally, in Section 5, conclusion and future work are presented.

2 PRELIMINARIES

In this section, we first define the problem of data placement in the wireless environment and then explain some notations that will be used throughout this paper. Finally, we quote a *QD* [8] method to measure average access time of each query that the mobile clients request.

2.1 Broadcast Scheduling Problem

In this section, we explain the need for

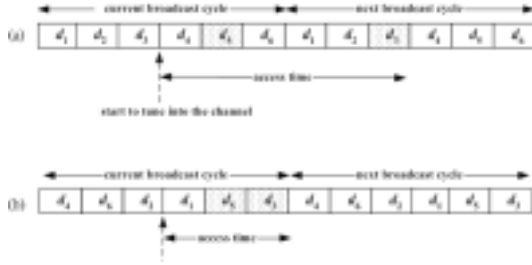


Fig. 2. An example for access time in different broadcast schedule.

scheduling wireless data items and illustrate the broadcast scheduling problem in the mobile computing environment.

The scheduling of wireless data items is to determine the sequence of data items to be broadcasted. We call the determined sequence of data items as a broadcast schedule, denoted by σ . In this paper, we investigate the wireless data placement on a single broadcast channel to reduce the clients' average access time and the broadcasted data items are not replicated. In other words, each data item appears exactly once on the broadcast channel. Moreover, we also consider that a query must access at least two data items.

In Fig.2, we assume that the server periodically broadcasts a set of data items $\langle d_1, d_2, d_3, d_4, d_5, d_6 \rangle$ and a client device wants to retrieve data items d_3 and d_5 . In Fig. 2a the entry of the client tuning into the channel is the position where a dotted line points. After the sever broadcasts data item d_4 in the current broadcast cycle, the client can access data item d_5 . However, data item d_3 in the current broadcast cycle passed, the client has to wait for the next broadcast cycle to access data item d_3 . Assume that all data items are of the same size. Hence the client spends 6 unit lengths of data items to complete its request. If the broadcast schedule $\sigma = \langle d_4, d_6, d_2, d_1, d_5, d_3 \rangle$ as shown in Fig. 2b, then the client can only spends 3 unit lengths of data items in the same broadcast cycle to access data items of interest. The access time in Fig. 2b is shorter than that in Fig. 2a. Therefore, a good broadcast schedule should place the data items accessed in a query closely to reduce the access time.

Table 1 Description of notations

Description	Notation
a data item to be broadcasted	d_i
the length of data item d_i	$ d_i $
the set of data items d_i	D
a query that is issued on the broadcast data stream	q_i
the frequency of q_i	$freq(q_i)$
the set of data items which q_i access	$QDS(q_i)$
the number of data items in $QDS(q_i)$	$ QDS(q_i) $
the set of queries	Q
the length of a broadcast stream	B
the broadcast schedule	σ
the position of data item d_i on σ	$\sigma^{-1}(d_i)$

To clearly explain the data placement for wireless broadcast, we first provide relevant notations that will be used throughout the paper. Table 1 shows the description of notations. Given a set of queries $Q = \{q_1, q_2, \dots, q_m\}$ and a set of data items $D = \{d_1, d_2, \dots, d_n\}$. Each query q_i accesses a set of data items called *Query Data Set*, represented by $QDS(q_i)$, where $QDS(q_i) \subset D$, and $D = \bigcup_{1 \leq i \leq m} q_i$.

$|QDS(q_i)|$ denotes the number of data items in the data set $QDS(q_i)$. The frequency of query q_i is denoted by $freq(q_i)$ and is named as *query frequency*. Let B denotes the length of a broadcast stream.

Thus, B is equal to $\sum_{1 \leq i \leq n} |d_i|$, where $|d_i|$ is the length of data item d_i . We denote a broadcast schedule on a broadcast channel by $\sigma = \langle d_i, d_j, \dots, d_k \rangle$. $\sigma^{-1}(d_i)$ denotes the position of data item d_i on σ . For example, in Fig. 2a, σ for the broadcast cycle is $\langle d_1, d_2, d_3, d_4, d_5, d_6 \rangle$ and $\sigma^{-1}(d_3) = 3$.

2.2 Measure Definition

In this section, we explain the measure method for average access time.

Let $AT^{avg}(q_i, \sigma)$ denotes the average access time of a query q_i in a broadcast schedule σ . The data placement problem for wireless broadcast is to find a broadcast schedule σ that minimizes the total access time (*TAT*) of a set of queries Q , denoted by:

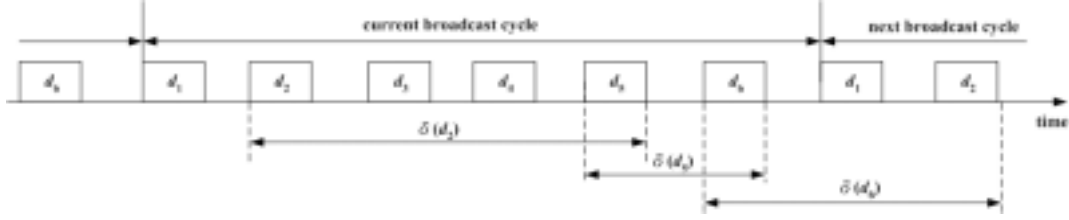


Fig. 3. Graphical illustration of the QD of a query.

$$TAT(\sigma) = \sum AT^{avg}(q_i, \sigma) \times freq(q_i), q_i \in Q$$

However, $AT^{avg}(q_i, \sigma)$ is hard to measure because it depends on the time of a client starting to tune into the broadcast channel. Fortunately, in [8], it proposed a measure method *Query Distance (QD)* to evaluate $AT^{avg}(q_i, \sigma)$. QD method is the minimal access time of a query considering all possible start positions. The measure was found to be effective for representing the average access time of a query that contains at least two data items. The definition of QD of query q_i on broadcast schedule σ is described as follows.

Definition 1 Given a $QDS(q_i) = \{d_{i_1}, d_{i_2}, \dots, d_{i_k}\}$ and a broadcast schedule σ . Let $\delta(d_{i_j})$ is the interval between d_{i_j} and $d_{i_{j-next}}$ (the broadcasted data item in $QDS(q_i)$ next to d_{i_j}) in the broadcast schedule σ . $\delta(d_{i_j}) = \sigma^{-1}(d_{i_{j-next}}) - \sigma^{-1}(d_{i_j}) - 1$, if d_{i_j} and $d_{i_{j-next}}$ are in the same broadcast cycle. Otherwise, if d_{i_j} and $d_{i_{j-next}}$ are in the different broadcast cycle, $\delta(d_{i_j}) = (\sigma^{-1}(d_{i_{j-next}}) + B) - \sigma^{-1}(d_{i_j}) - 1$, where B is the length of the broadcast stream σ . Then the QD of q_i on σ is defined as:

$$QD(q_i, \sigma) = B - \max(\delta(d_{i_j})), \text{ where } d_{i_j} \in QDS(q_i) \quad (1)$$

For example, given a broadcast schedule $\sigma = \langle d_1, d_2, d_3, d_4, d_5, d_6 \rangle$ and a query q_i with $QDS = \{d_2, d_5, d_6\}$. From Fig. 3, we get $\delta(d_2) = 5 - 2 - 1 = 2$, $\delta(d_5) = 6 - 5 - 1 = 0$, and $\delta(d_6) = (2 + 6) - 6 - 1 = 1$ (d_{6-next} is data item d_2 in the next broadcast cycle). The maximum $\delta(d_{i_j}) = \delta(d_2) = 2$. According to Equation (1), we can get $QD(q_i, \sigma)$ is $6 - 2 = 4$.

Lemma 1 [8] Given a query q_i and two schedules σ_1 and σ_2 , if $QD(q_i, \sigma_1) < QD(q_i, \sigma_2)$ then $AT^{avg}(q_i, \sigma_1) < AT^{avg}(q_i, \sigma_2)$.

By Lemma 1, we can use the new metric QD to measure access time of a query. So we redefine the problem of wireless data placement as follows. Let $TQD(\sigma)$ define the total query distance in broadcast schedule σ , denoted by:

$$TQD(\sigma) = \sum_{q_i \in Q} QD(q_i, \sigma) \times freq(q_i)$$

Problem 1 Given a set of queries Q and a set of data items D , the wireless data placement problem is to find a broadcast schedule σ such that $TQD(\sigma)$ is minimum.

Theorem 1 [8] The wireless data placement problem in Problem 1 is NP complete.

3 EIGENVECTOR APPROACH TO WIRELESS DATA PLACEMENT

Eigenvector approach had been widely used to solve many order problems [9], [10]. In this section, we use eigenvector approach to solve the problem of the wireless data placement such that the mobile clients can access the data on air in a short latency.

3.1 Basic Idea

Given a set of queries $\mathcal{Q} = \{q_1, q_2, \dots, q_m\}$, we can construct a graph $G = (V, E)$ corresponding to \mathcal{Q} . V represents the set of the vertex and E represents the set of the edge. We choose to construct G using *star* net modeling, where for each edge we introduce a dummy node. Let $V = V_D + V_Q$ and the definition of V_D and V_Q are described as follows.

- (1) V_D : the set of broadcasted data items $\mathcal{D} = \{d_1, d_2, \dots, d_n\}$, where $V_D = \{v_{d_1}, v_{d_2}, \dots, v_{d_n}\}$.
- (2) V_Q : for each query q_i , we introduce a corresponding dummy node v_{q_i} . V_Q is the set of dummy node v_{q_i} , where $V_Q = \{v_{q_1}, v_{q_2}, \dots, v_{q_m}\}$.
- (3) Hence, in the graph G , the total number of V is the sum of the number of V_D and V_Q . In other words, $V = V_D + V_Q = |n + m|$.

The broadcast environment we consider is that a query contains at least two data items. Each query q_i accesses a set of data items represented $QDS(q_i) = \{d_{i_1}, d_{i_2}, \dots, d_{i_j}\}$, and we introduce a dummy node v_{q_i} for query q_i . Then we build j edges between $(v_{q_i}, d_{i_1}), (v_{q_i}, d_{i_2}), \dots,$ and (v_{q_i}, d_{i_j}) with weight $freq(q_i) / (|QDS(q_i)| - 1)$, where $|QDS(q_i)|$ denotes the number of data items in the data set $QDS(q_i)$. Finally, this graph G can be

described by the $(n + m) \times (n + m)$ adjacency matrix $A = [a_{ij}]$, where the matrix element a_{ij} is the weight of the connection between dummy node v_{q_i} and data item d_{i_j} of $QDS(q_i)$.

After transforming a set of \mathcal{Q} into a graph, the broadcast scheduling problem can be seen as the linear placement problem as below:

$$\min \sum_{i>j} \sum_j a_{ij} |\sigma^{-1}(d_i) - \sigma^{-1}(d_j)|$$

where $\sigma^{-1}(d_k)$ is the position of data item d_k on broadcast schedule σ where $1 \leq k \leq n$. The linear placement problem is known to be NP-complete [8]. With a spectral approach, a *continuous* linear placement, where the restriction on placing data items at specific position is released, is usually used as the heuristic to solve the above linear placement problem.

Example 1 Table 2 shows a query profile. Let us assume that there are eight data items to be broadcasted and four queries that mobile clients request on the broadcast channel. The size of all data items is assumed to be equal. The $QDS(q_1)$ contain data items $d_2, d_4,$ and d_6 . $freq(q_1) = 35$ shows that the total number of the query q_1 requested by mobile clients is 35. Due to the number of the queries are four, we introduce four dummy nodes $v_{q_1}, v_{q_2}, v_{q_3}, v_{q_4}$. According to the above definition, we can obtain the weight of $(d_2, v_{q_1}), (d_4, v_{q_1}), (d_6, v_{q_1}) = 35 / (|QDS(q_1)| - 1) = 17.5$; $(d_3, v_{q_2}), (d_5, v_{q_2}), (d_6, v_{q_2}) = 35 / (|QDS(q_2)| - 1) = 17.5$; $(d_1, v_{q_3}), (d_5, v_{q_3}), (d_7, v_{q_3})$

Table 2 A query profile.

$QDS(q_i)$	$freq(q_i)$
$QDS(q_1) = \{d_2, d_4, d_6\}$	35
$QDS(q_2) = \{d_3, d_5, d_6\}$	35
$QDS(q_3) = \{d_1, d_5, d_7\}$	33
$QDS(q_4) = \{d_2, d_4, d_8\}$	33

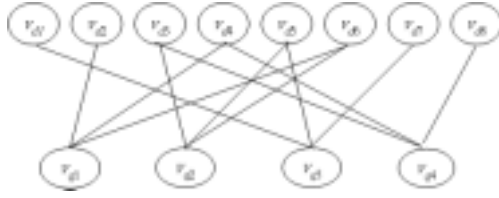


Fig. 4. A graph corresponding to a set of queries.

	v_{q1}	v_{q2}	v_{q3}	v_{q4}	v_{d1}	v_{d2}	v_{d3}	v_{d4}	v_{d5}	v_{d6}	v_{d7}	v_{d8}
v_{q1}												16.5
v_{q2}	17.5											
v_{q3}		34										
v_{q4}			34									
v_{d1}												
v_{d2}												
v_{d3}												
v_{d4}												
v_{d5}												
v_{d6}												
v_{d7}												
v_{d8}												
v_{d1}												
v_{d2}												
v_{d3}												
v_{d4}												
v_{d5}												
v_{d6}												
v_{d7}												
v_{d8}												
v_{d1}												
v_{d2}												
v_{d3}												
v_{d4}												
v_{d5}												
v_{d6}												
v_{d7}												
v_{d8}												

Fig. 5. An adjacency matrix.

$$= 33 / (|3| - 1) = 16.5; (d_3, v_{q_4}), (d_4, v_{q_4}), (d_8, v_{q_4}) = 33 / (|3| - 1) = 16.5.$$

Fig. 4 shows a graph corresponding to Example 1 and its corresponding adjacency matrix A as shown in Fig. 5.

3.2 Spectral Placement

In this section, we explain a theoretical basis for our method and the concepts of spectral placement.

Given a weighted graph $G = (V, E)$, represented by the $(n + m) \times (n + m)$ adjacency matrix $A = [a_{ij}]$. Let $D = [d_{ij}]$ be a diagonal matrix with $d_{ii} = \sum_{j=1}^{n+m} a_{ij}$. Fig. 6 shows a diagonal matrix corresponding to Example 1. Finally, the $(n + m) \times (n + m)$ Laplacian matrix of A

	v_{q1}	v_{q2}	v_{q3}	v_{q4}	v_{d1}	v_{d2}	v_{d3}	v_{d4}	v_{d5}	v_{d6}	v_{d7}	v_{d8}
v_{q1}	16.5											
v_{q2}		17.5										
v_{q3}			34									
v_{q4}				34								
v_{d1}					34							
v_{d2}						35						
v_{d3}							16.5					
v_{d4}								16.5				
v_{d5}									16.5			
v_{d6}										16.5		
v_{d7}											16.5	
v_{d8}												16.5

Fig. 6. A diagonal matrix.

is given by $Q = D - A$.

Hall's result [11] showed that the eigenvectors of the matrix $Q = D - A$ solve the one-dimensional *quadratic placement problem* of finding the vector $X = (x_1, x_2, \dots, x_n)$ which minimizes

$$z = \frac{1}{2} \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} (x_i - x_j)^2 a_{ij} \quad (2)$$

Subject to the constraint $z = \frac{1}{2} \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} (x_i - x_j)^2 a_{ij}$

It can be shown that $z = X^T Q X$, so to minimize z we form the Lagrangian as follows

$$L = X^T Q X - \lambda (X^T X - 1)$$

Taking the first partial derivative of L with respect to x and setting it equal to zero yields

$$2QX - 2\lambda X = 0$$

which can be rewritten as

$$(Q - \lambda I)X = 0$$

Heuristic Discrete Placement: The solution to the continuous placement problem provides a heuristic solution to a discrete placement problem. A discrete linear placement can be obtained by ordering the vertices according to the continuous placement X .

However, in [18], Li et. al found that another modified revising strategy often yields better results than the above Equation (4) with the experiment. They also order the vertices according to the value of previous placement X_k . It is different that they replaced the coordinate of x_i with $\pi(i)$, where $\pi(i)$ is the position of vertex i in the permutation X_k .

According to the result of [18], we have rewritten a'_{ij} of Equation (4) as below:

$$a'_{ij} = \frac{freq(q_i)}{|\pi(v_{d_j}) - \pi(v_{q_i})|} \quad (5)$$

wher $\pi(v_{d_j})$ denotes the position of v_{d_j} and $\pi(v_{q_i})$ denotes the position of v_{q_i} on the placement X_k .

Example 2 Referring to Example 1, Fig. 8 shows the eigenvector corresponding to the nonzero smallest eigenvalue $\lambda = 2.522381$ is $X = \{x_1 = 0.4745, x_2 = -0.2670, x_3 = -0.1359, x_4 = -0.2477, x_5 = 0.1954, x_6 = -0.1379, x_7 = 0.4745, x_8 = -0.2717, v_{q1} = -0.2285, v_{q2} = -0.0275, v_{q3} = 0.4020, v_{q4} = -0.2301\}$. Initially, a discrete linear placement can be obtained by ordering vertex v_{d_j} excluding dummy node v_{q_i} according to the coordinate of eigenvector X . Finally, the wireless data order is $\sigma = \langle d_8, d_2, d_4, d_6, d_3, d_5, d_7, d_1 \rangle$.

Recall the query profile in Table 2 and Equation (5). We sort the coordinates of eigenvector, the order is $\langle v_{d_8}, v_{d_2}, v_{d_4}, v_{q_4}, v_{q_1}, v_{d_6}, v_{d_3}, v_{q_2}, v_{d_5}, v_{q_3}, v_{d_7}, v_{d_1} \rangle$. Hence We get $\pi(v_{d_2})$

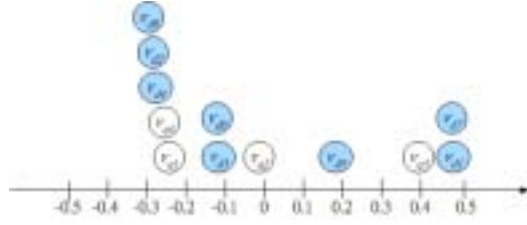


Fig. 8. A coordinate position graph of eigenvector.

$= 2, \pi(v_{d_4}) = 3, \pi(v_{d_6}) = 6, \pi(v_{q_1}) = 5$. The new weighted value of $(v_{d_2}, v_{q_1}), (v_{d_4}, v_{q_1})$ and (v_{d_6}, v_{q_1}) are $35 / |2-5| = 11.67, 35 / |3-5| = 17.5$ and $35 / |6-5| = 35$ and so on.

3.4 α -Order Objective Function

In the linear placement problem, the linear objective function obtains much better quality than the quadratic objective function; that is mainly because the linear function is more accurate measurement for the linear placement problem than the quadratic function. On the other hand, the quadratic function still has its advantage over the linear function. The quadratic function tends to make very long edges shorter than the linear function does, or the standard deviation of the edge lengths is smaller for the quadratic function than for the linear function [19]. This means the quadratic function tends to place vertices more sparsely, resulting in less vertices overlain each other.

Because the linear placement problem is heuristically solved by interpreting the eigenvector as the order of vertices, the more sparsely the vertices are placed, the less numerical errors are introduced on the linear placement; therefore, the continuous linear placement solution of the linear placement should be sparse enough to be interpreted while the objective function is as accurate as possible. Based on the result in [19], we propose the α -order objective function for the continuous linear placement problem as shown below:

$$\min \sum_{i>j} \sum_j freq(q_i) |\pi(d_{i_j}) - \pi(v_{q_i})|^\alpha =$$

$$\sum_{i>j} \sum_j \frac{freq(q_i)}{|\pi(d_{i_j}) - \pi(v_{q_i})|^{2-\alpha}} (\pi(d_{i_j}) - \pi(v_{q_i}))^2 \quad (6)$$

where $1.0 \leq \alpha \leq 2.0$. When $\alpha = 1.0$, the α -order function becomes the linear function; and the α -order function becomes the quadratic function when $\alpha = 2.0$. With the α -order objective function, we hope to increase the sparsity of the solution to the continuous linear placement, while we still maintain accurate enough measurement for the linear placement. Moreover this continuous linear placement with the α -order objective function can also be solved iteratively.

Because it is very hard and seem to be impossible to derive theoretical statements about the effects of various α on the solution to the linear placement problem; therefore, we here show the effects by experiments on different $QDS(q_i)$ in comparison with TQD . We do experiments on different $QDS(q_i)$ with $\alpha = 2.0$ (quadratic), $\alpha = 1.5$, $\alpha = 1.2$, and $\alpha = 1.0$ (linear) in the α -order objective function. The results are shown in Table 3.

As shown in Table 3, $nd100q100s002$ represents the amount of broadcasted data items is 100(it can be seen as the broadcast length is 100) and 100 query patterns that mobile clients request and each $QDS(q_i)$ contains 2% of all data items.

From the experimental results, it is easy to see that α -order objective function with $\alpha = 1.5$ has obtained the best quality on the linear placement in terms of total query distance(TQD) among these three cases. That is because the α -order($\alpha = 1.5$) objective function has higher sparsity on the continuous linear placement than the linear function does, and it also has more accurate measurement on the continuous linear placement than the quadratic function does, resulting in better performance.

Table 3 Comparison of different objective function.

	Linear	$\alpha = 1.2$	$\alpha = 1.5$	Quadratic
$nd100q100s002$	1656	1601	1551	1533
$nd200q100s002$	9662	9513	9427	10443
$nd300q100s002$	20543	22053	20180	20371

Algorithm LEA (Linearized Eigenvector Approach)

Input: Given a graph G of transforming $K QDS(q_i)$
Output: The broadcast scheduling

1. Take $K QDS(q_i)$ to construct a graph G described by an adjacency matrix A
2. Transform A into the matrix of the Laplacian Q
3. Calculate G 's Laplacian Q
4. $k = 1$;
5. Find the nonzero smallest eigenvalue of Q and its corresponding eigenvector X_k
6. Using X_k and Equation (6), giving new weights of A
7. $k = k + 1$;
8. Go to step 2 until the placement X_k converges

4 PERFORMANCE EVALUATIONS

In order to evaluate the performance of algorithm *LEA* and *QEM* [8], we implement a simulation model of the broadcast environment. Specifically, the simulation model is described in section 4.1. Experimental parameters are compared in section 4.2.

4.1 Simulation Models

We evaluate the performance of the proposed method through experiments in comparison with the *QEM* [8]. To the best of our knowledge, there is less other work that deals with a query accessing more than one data item. The performance metric to be considered in experiments is the total query distance (TQD) of all queries described in section 2.4. Using Java programming language, we generate the data of the experimental parameters randomly and the simulation is run on a desktop PC with P4 2.0 GHz, 256MB RAM and 40GB hard disk.

Table 4 summarizes the definitions for some primary simulation parameters. The number of data items to be broadcasted in a single broadcast channel is denoted by N and the amount of queries

Table 4 The parameters are used in the simulation

Parameter	Definition
N	Total number of data items to be broadcast
Q	Total number of queries requested by mobile clients
S	The percentage of all broadcasted data items that a query accesses

requested by mobile clients is Q . S represents the percentage of all data items which each query contains. The access frequencies of the query patterns have two distributions. One is normal distribution, and another is uniform distribution.

The following is the description about the parameters in experiments in detail.

- (1) *The number of data items (N):* This is the number of data items broadcasted in a single broadcast channel. A data item is broadcasted once in a broadcast cycle and accessed by at least one query. The size of each data item is assumed to be equal. Note that the size of a data item does not affect the performance of the experiments.
- (2) *The number of query patterns (Q):* This is the number of distinct queries that mobile clients submit on the broadcast channel. Each queries accesses at least two data items on the broadcast channel, and the set of data items which a query accesses ($QDS(q_i)$) is mutually independent and the amount of data items in each query is equal.
- (3) *The number of data items in a query's data set (S):* The selectivity S is the percentage of data items that a single query accesses among all the data items. For example, 2% selectivity means that every query accesses 2% of the data items being broadcast. In the simulation, we have experimented with various values of the number of data items that are accessed by a query.

4.2 Simulation Analysis

In our algorithm *LEA*, we use eigenvector by computing the nonzero smallest eigenvalue as our broadcasted data order. Then we revise the graph

iteratively by Equation (6) mentioned in section 3.4. In the process of iteration, we select the shortest TQD among 100 times in every experiment and use $\alpha = 1.5$ in the α -order objective function.

4.2.1 Effect of the Number of Data Items

Normal Distribution

First, we change the total number of data items (N) while 100 query patterns and 2% selectivity. The result of the experiment is shown in Fig. 10. As shown in the result, the TQD reduction of broadcast scheduling increases as the number of the data items increase. The reason is that the performance improvement decrease as the number of the data items increase in *QEM* approach. Therefore, *QEM* approach can get better result for less number of data items. However, the performance by *LEA* is better than that of *QEM* considering a broadcasted environment with less number of data items.

Uniform Distribution

In Fig. 11, we observe the variation of the number of data items. We use 100 query patterns and each of which accesses 2% data items of N . The result is shown in Fig. 11. As shown in the result, the TQD reduction in uniform distribution much more than that in normal distribution. The result is that all queries have similar access frequencies when the occurrence frequency of the query patterns is uniform distribution. Hence it is a disadvantage for *QEM* which expands the QDS of each query in a greedy manner after sorting all queries according to their corresponding access frequencies. It is the same as above experiment, *LEA* still outperforms *QEM* approach.

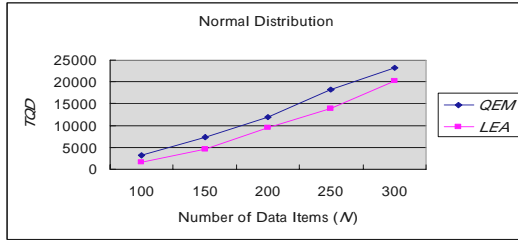


Fig. 10. TQD with various numbers of data items with normal distribution.

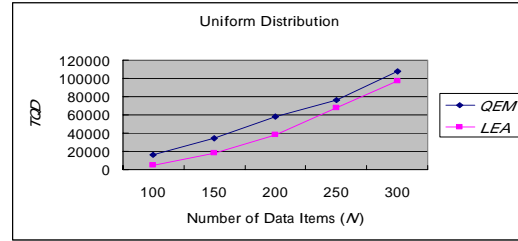


Fig. 11. TQD with various numbers of data items with uniform distribution.

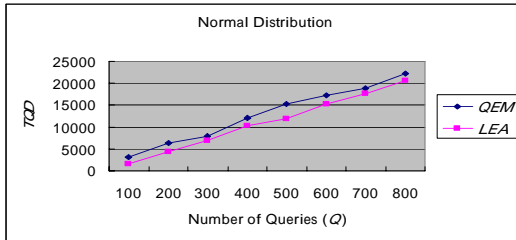


Fig. 12. TQD with various numbers of queries with normal distribution

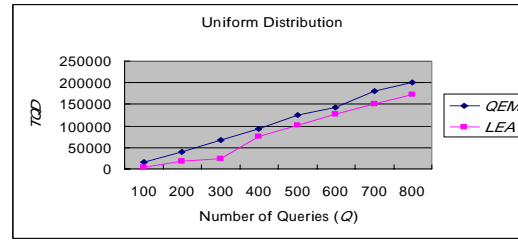


Fig. 13. TQD with various numbers of queries with uniform distribution

4.2.2 Effect of the Number of Query Patterns

Normal Distribution

Fig. 12 shows the result of an experiment with various numbers of query patterns. Here, the number of data items for broadcasting is 100, and each query accesses 2% of all data items. The performance is basically increased when the number of queries increases. However, our proposed approach gives better performance than the previous one (QEM).

Uniform Distribution

In Fig. 13, we change various numbers of query patterns while 100 broadcasted data items and 2% selectivity. As shown in the result, the TQD by LEA is shorter than that by QEM . We think it is from the fact that the overlapping data items are fewer and all queries have similar access frequencies. Therefore, QEM cannot allocate the high frequent

co-accessed data items adjacently, which on the other hand, the average access time becomes longer.

4.2.3 Effect of Selectivity Parameter S

In this section, we show the performance result according to various numbers of data items that a query accesses i.e., the size of a query's data set. In Fig. 14, we use 100 data items and 500 query patterns in this experiment. As shown in the Fig. 14, the performance of LEA is decreased with the increase of the size of a query's data set. This is because, as a query accesses more data items, data items are shared by more queries and thus, it is more difficult to find a broadcast schedule that is good for many queries. Particularly, the selectivity is smaller the performance is better. This benefit is meaningful. The reason is that a large value of selectivity means that a large number of the data items are in each query. As the value of selectivity becomes large, the order of broadcasted data items will not be so important for clients to access the broadcasted data items.

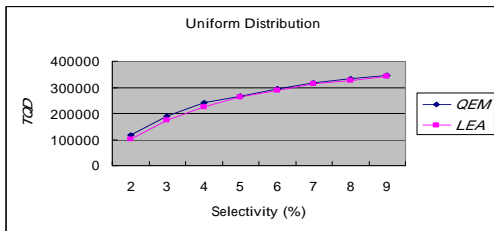


Fig. 14. Effect of selectivity

5 CONCLUSION

In this paper, we introduce the data placement problem for wireless broadcast when the mobile clients' queries access at least two data items on the wireless broadcast channel. As discussed, how to schedule the broadcast data can affect access time significantly. Using eigenvector approach, we can improve the quality of data order by iteratively revising the edge weights based on the current placement. We also set a linearization function named α -order objective function to adjust the coordinates of the eigenvector such that the order of data items obtain better quality. Simulation is performed to compare the performance between our approaches with a greedy approach. The results of experiments show that our data scheduling method can reduce the average access time such that the mobile clients can access the data on air in a short latency.

As a future work, we will extend this work on the environment with more large number of data items and more highly skewed frequency distribution of queries.

REFERENCES

- [1] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communication Environment," *Proceedings of ACM SIGMOD International Conference*, pp. 199-210, 1995.
- [2] S. Acharya, M. Franklin, and S. Zdonik, "Dissemination-based Data Delivery Using Broadcast Disks," *IEEE Personal Communications*, Vol. 2, No. 6, pp. 50-60, 1995.
- [3] S. Acharya, M. Franklin, and S. Zdonik, "Disseminating Updates on Broadcast Disks," *Proceedings of Very Large Data Bases Conference*, pp. 354-365, 1996.
- [4] S. Acharya, M. Franklin, and S. Zdonik, "Prefetching from a Broadcast Disk," *Proceedings of IEEE International Conference on Data Engineering*, pp. 276-285, 1996.
- [5] A. Bar-Noy and Y. Shilo, "Optimal Broadcasting of Two Files over an Asymmetric Channel," *IEEE INFOCOM Conference*, pp. 267-274, 1999.
- [6] M. S. Chen, P. S. Yu, and K. L. Wu, "Optimizing Index Allocation for Sequential Data Broadcasting in Wireless Mobile Computing," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 1, pp. 161-173, 2003.
- [7] Y. D. Chung and M. H. Kim, "QEM: A Scheduling Method for Wireless Broadcast," *Proceedings of International Conference on Database Systems for Advanced Applications*, pp. 135-142, 1999.
- [8] Y. D. Chung and M. H. Kim, "Effective Data Placement for Wireless Broadcast," *Distributed and Parallel Database*, pp. 133-150, 2001.
- [9] J. Frankle and R. M. Karp, "Circuit Placements and Cost Bounds by Eigenvector Decomposition," *Proceedings of IEEE International Conference on Computer-Aided Design*, pp. 414-417, 1986.
- [10] S. W. Hadley, B. L. Mark, and A. Vannelli, "An Efficient Eigenvector Approach for Finding Netlist Partitions," *IEEE Transactions on Computer-Aided Design*, Vol. 11, pp. 885-892, 1992.
- [11] K. M. Hall, "An r-Dimensional Quadratic Placement Algorithm," *Management Science*, Vol. 17, pp. 219-229, 1970.
- [12] Q. Hu, W. C. Lee, and D. L. Lee, "Indexing Techniques for Wireless Data Broadcast under Data Clustering and Scheduling," *Proceedings of ACM International Conference on Information and Knowledge*

- Management*, pp. 351-358, 1999.
- [13] J. J. Hung and Y. Leu, "An Energy Efficient Data Reaccess Scheme for Data Broadcast in Mobile Computing Environments," *Proceedings of IEEE International Conference on Parallel Processing Workshops*, pp. 5-12, 2003.
- [14] J. J. Hung and Y. Leu, "Efficient Index Caching Schemes for Data Broadcasting in Mobile Computing Environments," *Proceedings of IEEE International Workshop on Database and Expert Systems Applications*, pp. 139-143, 2003.
- [15] T. Imielinski, S. Viswanathan, and B. R. Badrinath, "Data on Air: Organization and Access," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 9, No. 3, pp. 353-372, 1997.
- [16] T. Imielinski, S. Viswanathan, and B. R. Badrinath, "Power Efficient Filtering of Data on Air," *Proceedings of International Conference on Extending Database Technology*, pp. 245-258, 1994.
- [17] G. Lee, S. C. Lo, and A. L. P. Chen, "Data Allocation on the Wireless Broadcast Channel for Efficient Query Processing," *IEEE Transactions on Computers*, Vol. 51, pp. 1237-1252, 2002.
- [18] J. Li, J. Lillis, and C. K. Cheng, "Linear Decomposition Algorithm for VLSI Design Applications," *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 223-228, 1995.
- [19] G. Sigl, K. Doll, and F. M. Johanness, "Analytical Placement: A Linear or a Quadratic Objective Function?" *Proceedings of the IEEE/ACM International Conference on Design Automation*, pp. 427-432, 1991.
- [20] C. J. Su, L. Tassiulas, and V. J. Tsotras, "Broadcast Scheduling for Information Distribution," *ACM-Baltzer Journal of Wireless Networks*, Vol. 5, No. 2, pp. 137-147, 1999.
- [21] C. J. Su and L. Tassiulas, "Joint Broadcast Scheduling and User's Cache Management for Efficient Information Delivery," *ACM Journal of Wireless Networks*, Vol. 6, No. 4, pp. 279-288, 2000.
- [22] N. H. Vaidya and S. Hameed, "Scheduling Data Broadcast in Asymmetric Communication Environment," *ACM-Baltzer Journal of Wireless Networks*, Vol. 5, No. 3, pp. 171-182, 1999.