

南 華 大 學

資訊管理學系碩士論文

利用頻譜方法於無線廣播頻道之資料

Data Broadcasting in Wireless Environment Using Spectral Approach



研 究 生：吳建億

指 導 教 授：吳光閔 博士

中 華 民 國 九 十 三 年 六 月

南 華 大 學

資訊管理學系

碩 士 學 位 論 文

Data Broadcasting in Wireless Environment Using Spectral  
Approach

研究生：吳建偉

經考試合格特此證明

口試委員：賴泳伶

王男斌

指導教授：吳光陵

所 長：\_\_\_\_\_



口試日期：中華民國 93 年 6 月 23 日

# Acknowledge

論文的完成，首先要感謝指導教授 吳光閔老師，每每的會議討論，使得背景知識、相關研究範疇、解決方法都能快速且完整的明瞭。研究方法的建立至論文內容的修正均有賴於老師細心的解惑與指導，才得以完成。此外，感謝口試委員王昌斌教授與賴泳伶教授於口試中對本論文提出不同的看法與建議，使得本論文更完善的呈現，在此亦表示感謝。

此段求學期間，除了感謝父母的支持，也特別感謝我的夥伴奎佑在課業上相互扶持與勉勵。也感謝慶華、美倫、如卿、建磐、元安、乾訓、閔皓、俊杰、明哲在生活及課業上的幫忙。

吳建億 謹識

于南華大學

民國九十三年六月

# 利用頻譜方法於無線廣播頻道之資料

研究生：吳建億

指導教授：吳光閔 博士

南華大學資訊管理學系碩士班

## 摘 要

在行動計算環境中，資料廣播對於資料的散播是一種有效率的方法。為了縮短行動客戶端的平均存取時間，許多策略已經提出如何在廣播頻道上安排資料項目的順序。然而，客戶端在廣播頻道上一次存取多個資料項目的議題是很少被討論到的。在本篇論文中，考量在無線環境中存取多個資料項目，我們提出一個有效率的方法，藉著使用特徵向量來改善資料配置的品質。比較先前的安排方法[12]，實驗結果顯示出我們資料安排演算法的效率。

關鍵字：資料配置、資料廣播、特徵向量

# Data Broadcasting in Wireless Environment

## Using Spectral Approach

Student : Chien-Yi Wu

Advisors : Dr. Guang-Ming Wu

Department of Information Management  
The M.B.A. Program  
Nan-Hua 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 research, 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 [12] and then show the promise of our data scheduling algorithm.

**Keywords:** data placement, data broadcasting, eigenvector

# Table of Contents

Qualified Certification.....	i
Acknowledge.....	ii
Chinese Abstract.....	iii
English Abstract.....	iv
Table of Contents.....	v
Table of Tables.....	vii
Table of Figures.....	viii
Chapter 1 Introduction.....	1
1.1 Background.....	1
1.2 Motivation.....	3
1.3 Related Work.....	7
1.4 Research Objectives.....	10
1.5 Overview of the Research.....	11
Chapter 2 Preliminary.....	12
2.1 Broadcast Scheduling Problem.....	12
2.2 Notations Description.....	13
2.3 Measure Definition.....	14
Chapter 3 Eigenvector Approach to Wireless Data Placement.....	17
3.1 Basic Idea.....	17
3.2 Spectral Placement.....	19
3.3 Placement Linearization.....	22
3.4 $\alpha$ -Order Objective Function.....	24

Chapter 4 Performance Evaluation .....	27
4.1 Simulation Models .....	27
4.2 Simulation Analysis.....	28
4.2.1 Effect of the Number of Data Items .....	28
4.2.2 Effect of the Number of Query Patterns.....	30
4.2.3 Effect of Selectivity Parameter S .....	32
 Chapter 5 Conclusion.....	 34
 Reference.....	 35

## Table of Tables

Table 2-1: Description of notations .....	14
Table 3-1: A query profile.....	18
Table 3-2: Comparison of different objective function .....	25
Table 4-1: The parameters are used in the simulation .....	27
Table 4-2: Improvement ratio with various data items with normal distribution .....	29
Table 4-3: Improvement ratio with various data items with uniform distribution .....	30
Table 4-4: Improvement ratio with various queries with normal distribution.....	31
Table 4-5: Improvement ratio with various queries with uniform distribution .....	32
Table 4-6: Improvement ratio with various numbers of selectivity.....	33



## Table of Figures

Figure 1-1: The architecture of the wireless communication environment.....	2
Figure 1-2: A push-based system.....	4
Figure 1-3: A pull-based system.....	5
Figure 1-4: A hybrid data broadcasting in an asymmetric environment.....	6
Figure 2-1: A example for access time in different broadcast schedule .....	13
Figure 2-2: Graphical illustraion of the $QD$ of a query .....	15
Figure 3-1: A graph corresponding to a set of queries.....	19
Figure 3-2: An adjacency matrix .....	19
Figure 3-3: A diagonal matrix.....	20
Figure 3-4: A Laplacian matrix.....	21
Figure 3-5: A coordinate position graph of eigenvector .....	23
Figure 4-1: $TQD$ with various numbers of data items with noraml distribution .....	29
Figure 4-2: $TQD$ with various numbers of data items with uniform distribution.....	30
Figure 4-3: $TQD$ with various numbers of queries with noraml distribution.....	31
Figure 4-4: $TQD$ with various numbers of queries with uniform distribution .....	32
Figure 4-5: Effect of selectivity.....	33

# Chapter 1 Introduction

## 1.1 Background

With the rapid advances of wireless technologies, a user can now access data from wireless networks any time any where through a mobile device. Figure 1-1 [21] illustrates the global architecture of the wireless communication environment. As shown in Figure 1-1, a wireless mobile network can be seen as a static wired part and a mobile wireless part. The details about two parts are described as follows.

A static wired part consists of fixed hosts and base stations that are interconnected through a high speed wired network. The fixed host provides query request from the base station about database. The base station can communicate with the nearest fixed host via the fixed network. Each base station has a wireless interface to communicate with the mobile units. Additional, each base station will provide commonly used application software, so that a mobile unit can download the software from the closet base station and run it on the palmtop or execute it remotely on the base station.

In a mobile wireless part, mobile units communicate with the base stations through wireless connections. A base station can only communicate with the mobile units moving within its coverage area called cell. Mobile units can communicate with each other only through at least one base station. Mobile units run on batteries while base stations are supplied by stable system power from static networks.

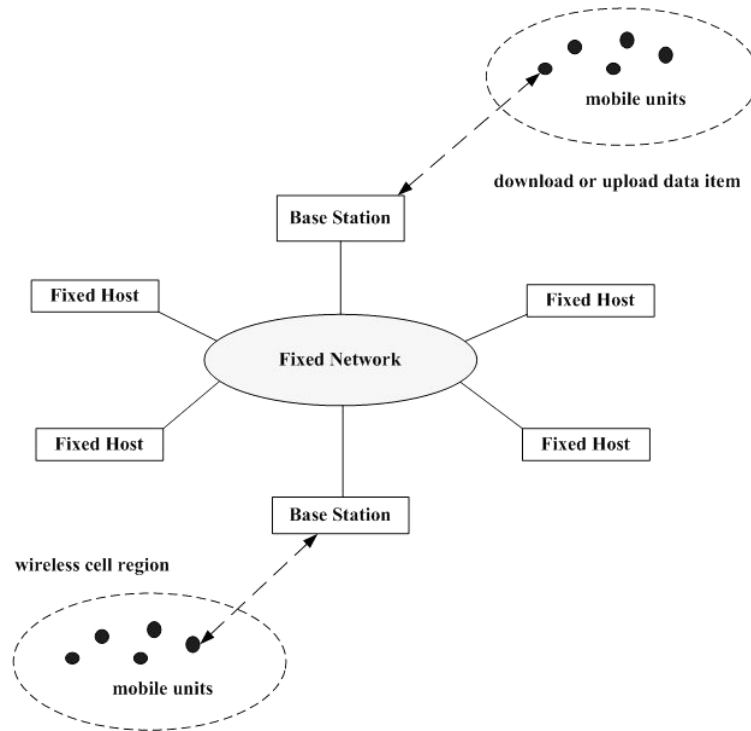


Figure 1-1: The architecture of the wireless communication environment

In wireless mobile computing, there is another wireless communication mode named Ad-Hoc networks where both servers and clients are nomadic. In ad-Hoc networks, every mobile unit can move freely and communicate directly with another mobile unit as long as that mobile unit is in its communication coverage area. However, in this environment, both servers and clients are mobile, energy conservation issues must be considered in developing broadcast strategies for both servers and clients. Cheng et al. [9] proposed *Minimum Longest Edge (MLE)* algorithm with polynomial time to constructs a broadcast tree using minimum spanning tree. This algorithm produced a broadcast tree with the shortest longest edge among all spanning trees for the objective of energy saving. To overcome communication overhands, Cho et al. [10] presented a *neighbor caching* strategy that makes neighboring nodes possible to share each other's caching storage and also presented *the ranking based prediction* that selects the most appropriate neighbor which data can be stored in. *The ranking based prediction* is an adaptive algorithm that adjusts frequency of neighbor caching and makes neighbor caching flexible according to the idleness of nodes.

## 1.2 Motivation

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], [11], [12], [22], 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. Finally, we summarize advantages of data broadcasting in the following:

- (1) *Power conservation*: this is due to the fact that mobile clients need not explicitly send data items to the server.
- (2) *High scalability*: The high scalability is achieved since the system performance is independent of the number of mobile clients.
- (3) *High bandwidth utilization*: data items of high interest can be received by multiple mobile clients by one transmission on the broadcast channel.

In the mobile computing environment, there are three kinds of data delivery techniques: pure-push-based, pure-pull-based, and a combination of these two techniques.

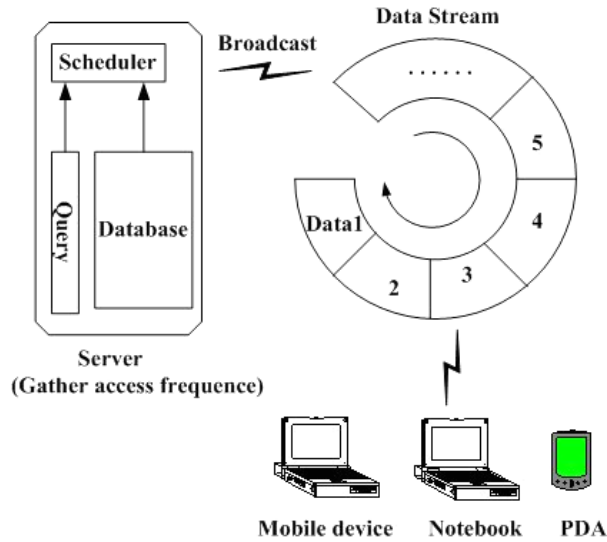


Figure 1-2: A push-based system

Figure 1-2 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. 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.

The pure-pull-based technique is that mobile clients have to explicitly send requests to the server as shown in Figure 1-3. The server then responds to each request individually via an on-demand channel. Such technique cannot satisfy any number of mobile clients due to restriction on the wireless bandwidth and the access latency depends on the server workload. Several previous researches used the algorithm of CPU scheduling like *First Come First Served (FCFS)* to schedule data items. In the later, some researches proposed the algorithm named *Most Requests First (MRF)* to broadcast the data items with the

maximum number of pending requests and the algorithm named *Longest Wait First (LWF)* to select the data items which the total wait time of pending requests is the largest. Aksoy et al. [5] also developed the  $R \times W$  algorithm, which combine the benefits of *MRF* and *LWF* in order to provide good performance for both hot and cold data items.

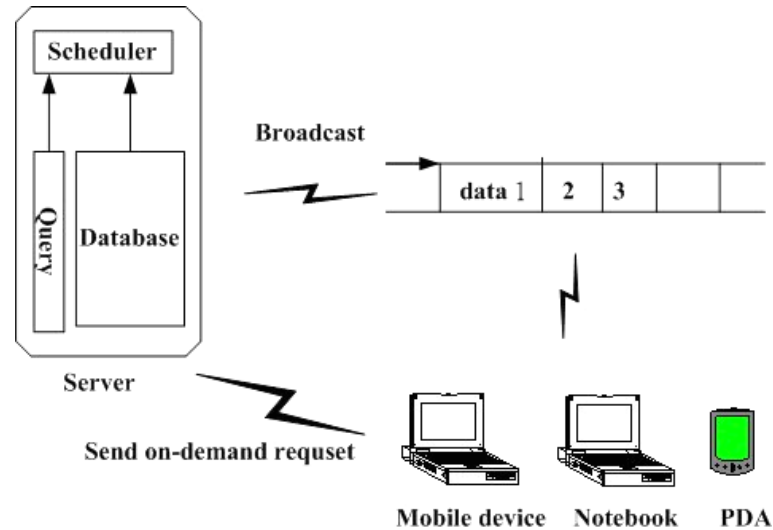


Figure 1-3: A pull-based system

The third technique is a combination of the previous two, which benefits both the server and clients as shown in Figure 1-4. In [32], a hybrid approach is proposed by using pure-push-based technique for hot data and pure-pull-based technique for cold data. In [16], Hu et al. proposed an adaptive balanced scheme (ABS) for a hybrid data delivery in a multi-channel data dissemination environment. The server broadcasts hot data items over push channel periodically and delivers cold data items by pull channel in respond to clients' explicit access requests.

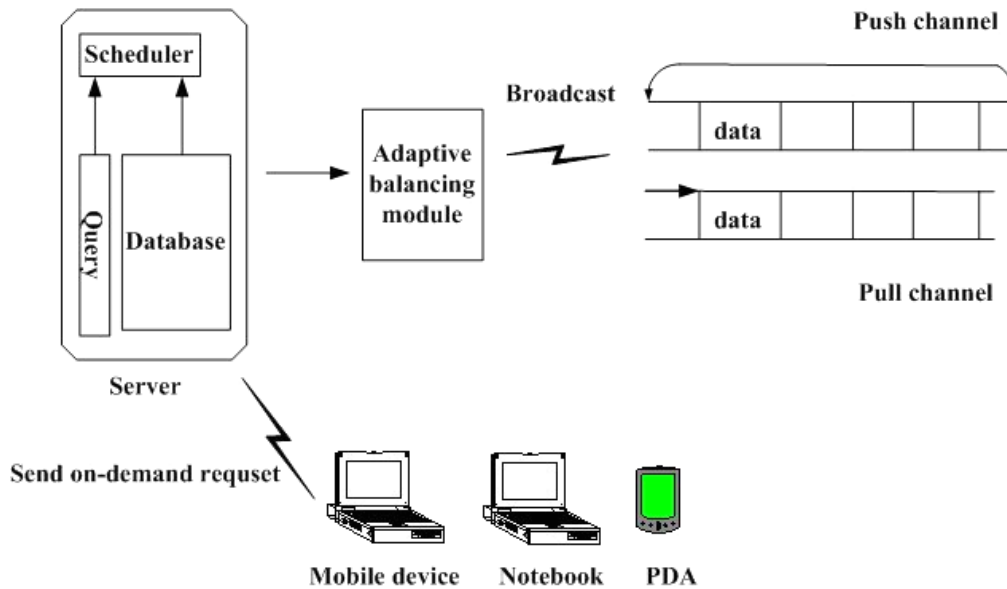


Figure 1-4: A hybrid data broadcasting in an asymmetric environment

However, most prior researches in above three data delivery techniques mainly considered a query can access only one data item. Moreover, 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. Hence, in this research, we not only consider that a query can access more than one data item but also determine the optimal order of broadcasted data items on a single broadcast channel to speed up the data access for mobile clients.

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 [22], the power consumption of a mobile device to download the required data items depends on the tuning time. For energy saving, it is important for a mobile client to be able to operate in two different modes: *doze mode*, where it is still connected to the network but it is not active and *active mode*, where it performs usual activities. Therefore, to achieve energy saving, it is highly desirable to let the mobile devices stay in the doze mode most of the time and only come to the active mode when it is necessary.

To reducing tuning time, most researches used index techniques such as distributed index [22], hashing [23] and signature [17]. 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 research in [8] 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 researches [1], [11], [12], [24], [33], [35] made efforts in the order of broadcasted data item to minimize the clients' access time as much as possible.

To shorten the average access time of mobile clients, several broadcast scheduling strategies have been proposed [1], [33], [35]. Acharya et al. [1] analyzed the access preference of each data item and differentiates the delivery frequencies of data items. That is, the more popular data items are more frequently broadcasted. This way gives an advantage for frequently accessed data items, and a disadvantage for unpopular ones due to the increase of the broadcast cycle. The scheduling method [33] constructed the broadcast schedule by using the stochastic model. It considered the access frequencies of data items and controls their delivery intervals. In [35], 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.

### **1.3 Related Work**

To reduce the access time of the mobile clients, some scholars [1], [12], [24], [35] proposed several efficient broadcast schedules. 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 above techniques address some client issues such as cache management [1], [4], [19], [20], [34] and prefetching [4] and also some server issues such as allocation of data [1], [12], [24] 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. [20] 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. [19] 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 a real world, some information has to be delivered within timing constraints while other information does not have a timing constraint. For example, current stock list information must be broadcasted in a few minutes or seconds but weather information does not have to be broadcasted in a few minutes. Hence time sensitive information should be broadcasted within their time constraint, while time insensitive information of common interest to many users need to be broadcasted with minimized average access time.

Baruah et al. [6] introduced the concept of real-time Bdisks (broadcast disks) which had additional advantage of being fault-tolerant. The idea of real-time fault-tolerant Bdisks was augmented with a facility to take into consideration real-time updates to the data being broadcast. Lim et al. [26] further proposed real-time broadcast algorithms to meet the

deadline for real-time information as well as to minimize the average access time for non-real-time information. They assigned the required bandwidth periodically for transferring the real-time information. The rest of the capacity of the communication channel can be used for transferring the non-real time information. Clients will satisfy when the non-real-time information is broadcasted with minimized average access time and real-time information is broadcasted within their deadline.

There are some other researches focusing on generating broadcast program on multiple channels. Organizing data in multiple broadcast channels raises a number of new research problems. One is to construct a skewed index tree with the popular data having a shorter path from the root of the index tree, which minimizes the average tuning time [8], [27], [28], [29]. The construction of this index tree is analogous to that of Huffman code, which is pointed out in [30]. Another class of Huffman trees termed *Alphabetic Huffman tree* is proposed in [18], which functions as a binary search tree. Shivakumar et al. [30] extend it to a k-nary search tree and allocate this index tree to multiple channels. However, it is inflexible because the number of channels must equal the height of the tree.

The research in [28] explored the problem of generating hierarchical broadcast programs with the data access frequencies and the number of broadcast disks in a broadcast disk array given. Specially, the problem of generating hierarchical broadcast program is first transformed into the one of constructing a *channel allocation tree* with variant-fanout. By exploring the feature of tree generation with variant-fanout, a heuristic algorithm to minimize the average access time of data items in the broadcast program is developed. Peng and Chen [28] constructed a channel allocation tree with variant fan-out and propose a heuristic algorithm  $VF^k$  to generate a broadcast program. This approach only works well as the number of channels is a power of 2.

Peng and Chen [29] further considered that the data access frequencies may vary with time. They developed an efficient algorithm  $DL$  to dynamically adjust broadcast programs to effectively respond to the changes of data access frequencies. Lo and Chen [27] proposed a solution for optimal index and data allocation in multiple broadcast channels, which minimizes both the average access time and the average tuning time for any number of broadcast channels. Chen et al. [8] devised algorithms  $CF$  (fixed index fanouts) for

constructing a balanced index tree and *VF* (variant index fanouts) for constructing an imbalanced index tree. The two algorithms were based on access frequencies of data items. In addition, they also developed an algorithm *ORD* to determine the optimal order for sequential data broadcasting with a given index tree.

## 1.4 Research Objectives

In the broadcast-based information systems with pure-push-based technique, a server continuously and repeatedly broadcasts data items on a specific broadcast channel. The mobile clients can only sequentially access data items appearing on the broadcast channel. Hence, it is of importance to arrange the order of broadcasted data items to satisfy each mobile clients listening on the broadcast channel.

In this research, 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 research, 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 [7], [11], [12], [24]. In [7], the method for finding the optimal broadcast program for two dependent files was proposed. In Chung and Kim [11], a broadcast schedule method called *Query Expansion Method (QEM)* to minimize the average access time was proposed. In [24], 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 [12], 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.

In this research, the following system environment is assumed:

- (1) The server broadcasts data items via pure-push-based technique.
- (2) The data items are broadcasted with a single broadcast channel.
- (3) A query that the mobile clients request must contain at least two data items.
- (4) Each data item appears exactly once on the broadcast channel (we can call this broadcast approach as uniform broadcasting).
- (5) 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 [13], [14] 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 [12] with our approach. The results show that our broadcast schedule generated by eigenvector approach outperforms the broadcast schedule in a greedy manner [12].

## **1.5 Overview of the Research**

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

## Chapter 2 Preliminary

In this chapter, we first define the problem of data placement in the wireless environment and then explain some notations that will be used throughout this research. Finally, we quote a *QD* [12] 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 scheduling wireless data items and illustrate the broadcast scheduling problem in the mobile computing environment.

In the data broadcasting mechanism, we have two goals to solve. First, by shorten tuning time to save the mobile device's power. Second, minimize access time to reduce the waiting time of the mobile client. Hence, how to placement data items on air to speed up data access for the mobile client is very of importance.

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  $\sigma$ . In this research, 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.

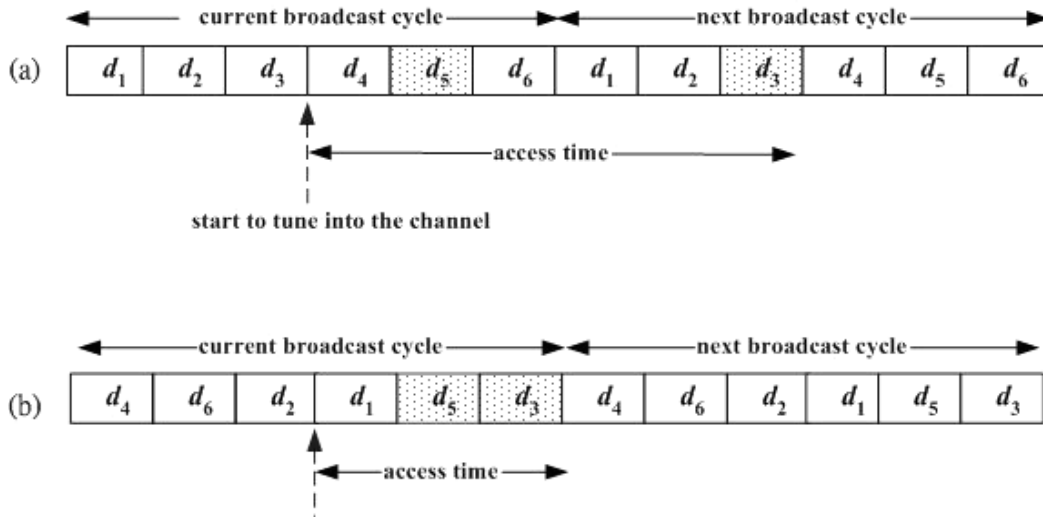


Figure 2-1: An example for access time in different broadcast schedule

In Figure 2-1, 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 Figure 2-1(a) 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 accesses 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 length 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 Figure 2-1(b), then the client can only spends 3 unit length of data items in the same broadcast cycle to access data items of interest. The access time in Figure 2-1(b) is shorter than that in Figure 2-1(a). Therefore, a good broadcast schedule should place the data items accessed in a query closely to reduce the access time.

## 2.2 Notations Description

To clearly explain the data placement for wireless broadcast, we first provide relevant notations that will be used throughout the research. Table 2-1 shows the description of notations. Given a set of queries  $\mathcal{Q} = \{q_1, q_2, \dots, q_m\}$  and a set of data items  $\mathbf{D} = \{d_1, d_2, \dots, d_n\}$ . Each query  $q_i$  accesses a set of data items called *Query Data Set*, represented by  $QDS(q_i)$ , where  $QDS(q_i) \subset \mathbf{D}$ , and  $\mathbf{D} = \bigcup_{1 \leq i \leq m} q_i$ .  $|QDS(q_i)|$  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  $\sigma$ . For example, in Figure 2-1(a),  $\sigma$  for the broadcast cycle is  $\langle d_1, d_2, d_3, d_4, d_5, d_6 \rangle$  and  $\sigma^{-1}(d_3) = 3$ .

Table 2-1: 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	$\sigma$
the position of data item $d_i$ on $\sigma$	$\sigma^{-1}(d_i)$

## 2.3 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  $\sigma$ . The data placement problem for wireless broadcast is to find a broadcast schedule  $\sigma$  that minimizes the total access time ( $TAT$ ) of a set of queries  $Q$ , denoted by:

$$TAT(\sigma) = \sum AT^{avg}(q_i, \sigma) \times freq(q_i), \quad q_i \in Q \quad (2-1)$$

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 [12], 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  $\sigma$  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  $\sigma$ . 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  $\sigma$ .  $\delta(d_{i_j}) = \sigma^{-1}(d_{i_{j+next}}) - \sigma^{-1}(d_{i_j}) - 1$ , 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  $\sigma$ . Then the  $QD$  of  $q_i$  on  $\sigma$  is defined as:

$$QD(q_i, \sigma) = B - \max(\delta(d_{i_j})), \text{ where } d_{i_j} \in QDS(q_i) \quad (2-2)$$

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 Figure 2-2, 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 (2-2), we can get  $QD(q_i, \sigma)$  is  $6 - 2 = 4$ .

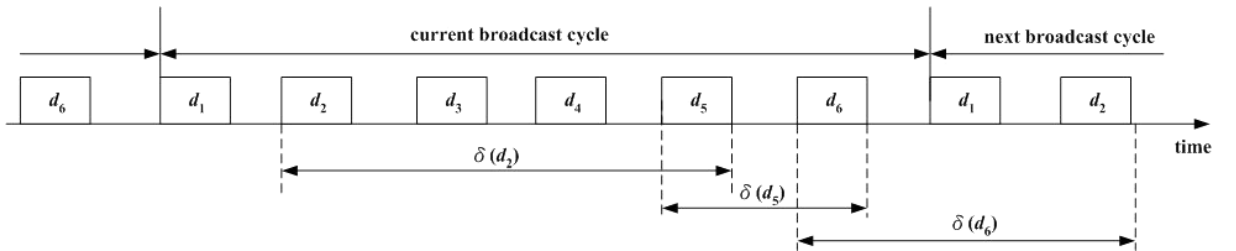


Figure 2-2: Graphical illustration of the  $QD$  of a query

**Lemma 1** [12] Given a query  $q_i$  and two schedules  $\sigma_1$  and  $\sigma_2$ , 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  $\sigma$ , denoted by:



$$TQD(\sigma) = \sum_{q_i \in Q} QD(q_i, \sigma) \times freq(q_i) \quad (2-3)$$

**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  $\sigma$  such that  $TQD(\sigma)$  is minimum.*

**Theorem 1** [12] *The wireless data placement problem in Problem 1 is NP complete.*

# Chapter 3 Eigenvector Approach to Wireless Data Placement

Eigenvector approach had been widely used to solve many order problems [13], [14]. In this chapter, 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  $\mathbf{D} = \{d_1, d_2, \dots, d_n\}$ , 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  $Q$  into a graph, the broadcast scheduling problem can be seen as the linear placement problem as below:

$$\min \sum_{i>j} \sum_j a_{ij} |\sigma^{-1}(d_i) - \sigma^{-1}(d_j)| \quad (3-1)$$

where  $\sigma^{-1}(d_k)$  is the position of data item  $d_k$  on broadcast schedule  $\sigma$  where  $1 \leq k \leq n$ . The linear placement problem is known to be NP-complete [12]. 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.

Table 3-1: 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
$QDS(q_4) = \{d_3, d_4, d_8\}$	33

**Example 1** Table 3-1 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

$$(d_2, v_{q_1}), (d_4, v_{q_1}), (d_6, v_{q_1}) = 35 / (|Q| - 1) = 17.5,$$

$$(d_3, v_{q_2}), (d_5, v_{q_2}), (d_6, v_{q_2}) = 35 / (|Q| - 1) = 17.5,$$

$$(d_1, v_{q_3}), (d_5, v_{q_3}), (d_7, v_{q_3}) = 33 / (|Q| - 1) = 16.5,$$

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

Figure 3-1 shows a graph corresponding to Example 1 and its corresponding adjacency matrix  $A$  as shown in Figure 3-2.

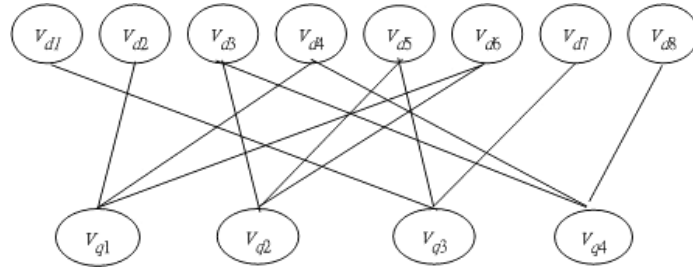


Figure 3-1: A graph corresponding to a set of queries

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

Figure 3-2: An adjacency matrix

### 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  $d_{ii} = \sum_{j=1}^{j=n+m} a_{ij}$ . Figure 3-3 shows a diagonal matrix corresponding to Example 1. Finally, the  $(n + m) \times (n + m)$  Laplacian matrix of  $A$  is given by  $Q = D - A$ .

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

Figure 3-3: A diagonal matrix

Hall's result [15] 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

$$z = \frac{1}{2} \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} (x_i - x_j)^2 a_{ij} \quad (3-2)$$

Subject to the constraint  $|X| = (X^T X)^{1/2} = 1$

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

$$L = X^T Q X - \lambda (X^T X - 1) \quad (3-3)$$

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

$$2QX - 2\lambda X = 0 \quad (3-4)$$

which can be rewritten as

$$(Q - \lambda I)X = 0 \quad (3-5)$$

where  $I$  is the identity matrix. This is readily recognizable as an eigenvalue formulation for  $\lambda$ , and the eigenvectors of  $Q$  are the only nontrivial solutions for  $X$ . The minimum eigenvalue 0 yields the uninteresting solution  $X = (1/\sqrt{n}, 1/\sqrt{n}, \dots, 1/\sqrt{n})$ , and hence the

eigenvector corresponding to the nonzero smallest eigenvalue which is a lower bound on a nontrivial solution to Equation (3-2) is used.

Figure 3-4 is the Laplacian matrix corresponding to Figure 3-2. In this research, the Laplacian matrix we construct has the following properties.

- $Q$  is the symmetric matrix.
- $Q$  has  $n + m$  eigenvalues  $\lambda_1 \quad \lambda_2 \quad \dots \quad \lambda_{n+m}$ , associated with the corresponding eigenvectors  $X_1, X_2, \dots,$  and  $X_{n+m}$ .
- $Q$  is singular (the Laplacian matrix's rows or columns sum to zero).
- It has rank at most  $n - 1$  and 0 as an eigenvalue.

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

Figure 3-4: A Laplacian matrix

By computing the nonzero smallest eigenvalue and its corresponding eigenvector  $X$  of the Laplacian  $Q$ , we obtain a non-trivial solution to above the *quadratic programming problem*, and the heuristic solution to the linear placement problem is obtained by interpreting the eigenvector as a linear order on the vertices  $V$ . With such an approach, as we can see, the objective function to be minimized is the squared wire length.

### 3.3 Placement Linearization

In this section, we propose a linearization method to improve the quality of linear order.

In our research, the linear objective function formulated as below is used to solve the continuous linear placement.

$$\min \sum_{i>j} \sum_j a_{ij} |x_i - x_j| \quad (3-6)$$

where  $x_i$  is the coordinate of vertex  $v_{d_j}$  of  $QDS(q_i) = \{d_{i_1}, d_{i_2}, \dots, d_{i_j}\}$  or  $v_{q_i}$  in the continuous placement.

Our goal to the spectral approach is that it minimizes the squared edge length of Equation (3-2) rather than the linear edge length of Equation (3-6) of the placement. Hence, such a linear objective function can also be rewritten as a quadratic function by modifying  $a_{ij}$  with the distance  $|x_i - x_j|$  as follows:

$$\sum_{i>j} \sum_j a_{ij} |x_i - x_j| = \sum_{i>j} \sum_j a'_{ij} (x_i - x_j)^2 \quad (3-7)$$

where

$$a'_{ij} = \begin{cases} \frac{a_{ij}}{|x_i - x_j|} & \text{if } |x_i - x_j| > \varepsilon \\ \frac{a_{ij}}{\varepsilon} & \text{otherwise} \end{cases} \quad (3-8)$$

Equation (3-8) specifies the weight revising method and  $\varepsilon$  is numerical stability. Therefore, the quadratic programming method can still be used to solve the continuous linear placement with linear objective function. After that we use new  $a'_{ij}$  to revise the edge weights based on the previous placement iteratively.

**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 [25], Li et. al found that another modified revising strategy often yields better results than the above Equation (3-8) 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 [25], we have rewritten Equation (3-8) as below:

$$a'_{ij} = \frac{freq(q_i)}{|\pi(v_{d_{ij}}) - \pi(v_{q_i})|} \quad (3-9)$$

where  $\pi(v_{d_j})$  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, Figure 3-5 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 3-1 and Equation (3-9). 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  $\pi(v_{d_2}) = 2, \pi(v_{d_4}) = 3, \pi(v_{d_6}) = 6, \pi(v_{q_1}) = 5$ . 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.

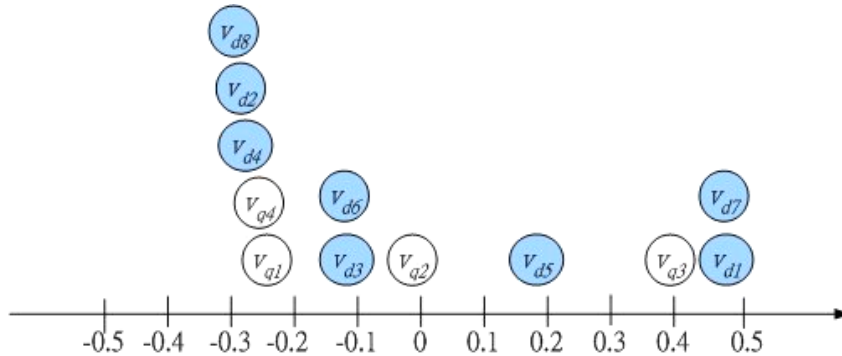


Figure 3-5: A coordinate position graph of eigenvector



### 3.4 $\alpha$ -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 [31]. 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 [31], we propose the  $\alpha$ -order objective function for the continuous linear placement problem as shown below:

$$\min \sum_{i>j} \sum_j freq(q_i) |\pi(d_{i_j}) - \pi(v_{q_i})|^\alpha = \sum_{i>j} \sum_j \frac{freq(q_i)}{|\pi(d_{i_j}) - \pi(v_{q_i})|^{2-\alpha}} (\pi(d_{i_j}) - \pi(v_{q_i}))^2 \quad (3-10)$$

where  $1.0 \leq \alpha \leq 2.0$ . When  $\alpha = 1.0$ , the  $\alpha$ -order function becomes the linear function; and the  $\alpha$ -order function becomes the quadratic function when  $\alpha = 2.0$ . With the  $\alpha$ -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  $\alpha$ -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  $\alpha$  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  $\alpha$ -order objective function. The results are shown in Table 3-2.

As shown in Table 3-2,  $nd100q100s002$  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.

Table 3-2: 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

From the experimental results, it is easy to see that  $\alpha$ -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  $\alpha$ -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.

**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 (3-10), giving new weights of  $A$
7.  $k = k + 1$ ;
8. Go to step 2 until the placement  $X_k$  converges

# Chapter 4 Performance Evaluation

In order to evaluate the performance of algorithm *LEA* and *QEM* [12], 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* [12]. 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 4-1: 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

Table 4-1 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 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 (3-10) 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  $\alpha$ -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 Figure 4-1. 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.

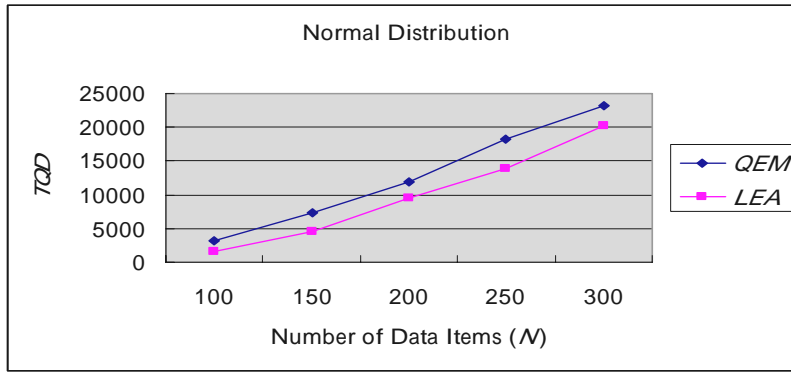


Figure 4-1:  $TQD$  with various numbers of data items with normal distribution

The improvement ratio about the change of data items are summarized in Table 4-2. The results of  $LEA$  are superior to  $QEM$  in terms of the total query distance. On the average,  $LEA$  yields improvement of 29.25% over  $QEM$  considering the occurrence frequency of the query patterns is normal distribution.

Table 4-2: Improvement ratio with various data items with normal distribution

Normal Distribution			
Number of data items	$QEM$	$LEA$	Improve (%)
100	3158	1551	+50.89
150	7252	4528	+37.56
200	11940	9427	+21.05
250	18188	13847	+23.87
300	23160	20180	+12.87
Average			+29.25

- Uniform Distribution

In Figure 4-2, 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 Figure 4-2. 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.

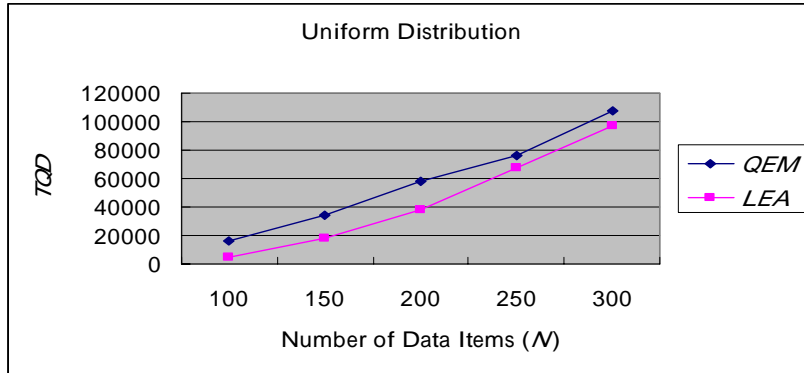


Figure 4-2:  $TQD$  with various numbers of data items with uniform distribution

The improvement ratio about the change of data items are summarized in Table 4-3. The results of  $LEA$  are superior to  $QEM$  in terms of the total query distance. On the average,  $LEA$  yields improvement of 34.98% over  $QEM$  in Table 4-3 considering the occurrence frequency of the query patterns is uniform distribution

Table 4-3: Improvement ratio with various data items with uniform distribution

Uniform Distribution			
Number of data items	$QEM$	$LEA$	Improve (%)
100	16310	4590	+71.86
150	34000	17680	+48.00
200	58490	38390	+34.36
250	75740	67790	+10.50
300	108080	97060	+10.20
Average			+34.98

## 4.2.2 Effect of the Number of Query Patterns

- Normal Distribution

Figure 4-3 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 ( $QEM$ ).

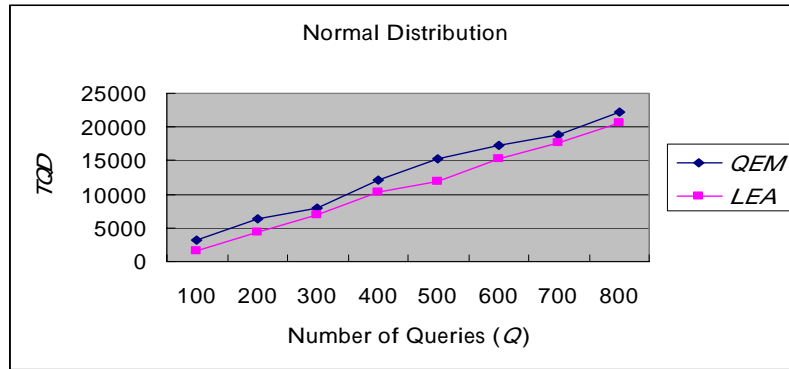


Figure 4-3:  $TQD$  with various numbers of queries with normal distribution

In Table 4-4, the results of  $LEA$  are superior to  $QEM$  in terms of the total query distance. On the average,  $LEA$  yields improvement of 19.86% over  $QEM$  considering the occurrence frequency of the query patterns is normal distribution.

Table 4-4: Improvement ratio with various queries with normal distribution

Normal Distribution			
Number of queries	$QEM$	$LEA$	Improve (%)
100	3185	1551	+50.89
200	6341	4289	+32.36
300	8034	6861	+14.60
400	12026	10329	+14.11
500	15330	11968	+21.93
600	17296	15217	+12.02
700	18771	17643	+6.00
800	22160	20600	+7.00
Average			+19.86

- Uniform Distribution

In Figure 4-4, 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.



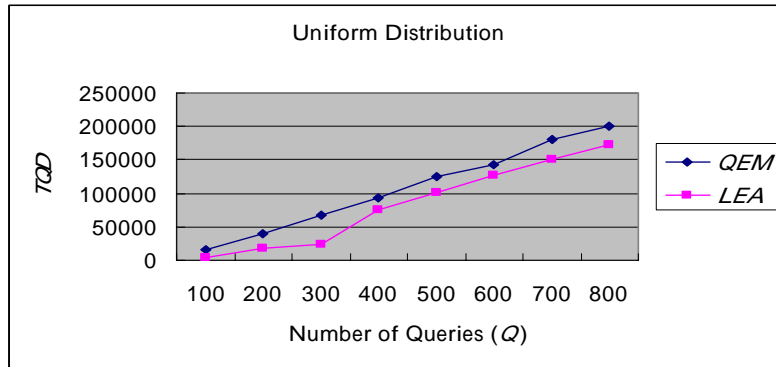


Figure 4-4:  $TQD$  with various numbers of queries with uniform distribution

In Table 4-5, the results of  $LEA$  are superior to  $QEM$  in terms of the total query distance. On the average,  $LEA$  yields improvement of 34.35% over  $QEM$  in Table 4-5 considering the occurrence frequency of the query patterns is uniform distribution.

Table 4-5: Improvement ratio with various queries with uniform distribution

Uniform Distribution			
Number of queries	$QEM$	$LEA$	Improve (%)
100	16310	4590	+71.86
200	39300	16980	+56.79
300	67950	22930	+66.25
400	92950	75250	+19.04
500	124380	101120	+18.70
600	142660	126190	+11.54
700	180790	150990	+16.48
800	199950	171670	+14.14
Average			+34.35

### 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 Figure 4-5, we use 100 data items and 500 query patterns in this experiment. As shown in the Figure 4-5, 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. Anyway, the performance of the proposed approach is better than *QEM*.

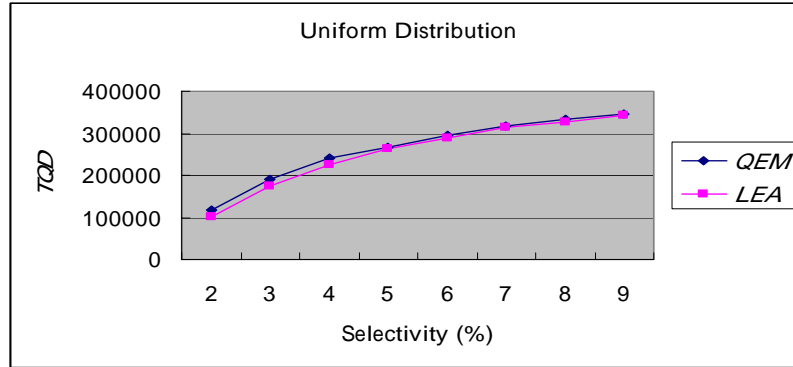


Figure 4-5: Effect of selectivity

The experiment results about the change of selectivity are summarized in Table 4-6. In this experiment, the occurrence frequency of each query shows uniform distribution. The results of *LEA* are superior to *QEM* in terms of the total query distance. On the average, *LEA* yields improvement of 4.56% over *QEM*.

Table 4-6: Improvement ratio with various numbers of selectivity

Selectivity (%)	<i>QEM</i>	<i>LEA</i>	Improve (%)
2	111630	100220	+13.85
3	191180	174350	+8.80
4	240850	225610	+6.33
5	266600	263870	+1.02
6	295410	289940	+1.85
7	318860	313510	+1.68
8	333360	326890	+1.94
9	345290	341800	+1.01
Average			+4.56

## Chapter 5 Conclusion

In this research, 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  $\alpha$ -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] D. Aksoy and M. Franklin, "RxW: A Scheduling Approach for Large-Scale On-Demand Data Broadcast," *ACM/IEEE Transactions on Networking*, Vol. 7, No. 6, pp. 846-860, 1999.
- [6] S. Baruah and A. Bestavros, "Real-Time Mutable Broadcast Disks," *Proceedings of RTDB'97: The Second International Workshop on Real-Time Databases*, 1997.
- [7] A. Bar-Noy and Y. Shilo, "Optimal Broadcasting of Two Files over an Asymmetric Channel," *IEEE INFOCOM Conference*, pp. 267-274, 1999.
- [8] 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.
- [9] M. X. Cheng, J. Sun, M. Min, and D. -Z. Du, "Energy-efficient Broadcast and Multicast Routing in Ad Hoc Wireless Networks," *Proceedings of IEEE International Conference on Performance, Computing, and Communications*, pp. 87-94, 2003.
- [10] J. Cho, S. Oh, J. Kim, H. H. Lee, and J. Lee, "Neighbor Caching in Multi-Hop Wireless Ad Hoc Networks," *IEEE Transactions on Communications Letters*, Vol. 7, pp. 525-527, 2003.

- [11] 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.
- [12] Y. D. Chung and M. H. Kim, "Effective Data Placement for Wireless Broadcast," *Distributed and Parallel Database*, pp. 133-150, 2001.
- [13] 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.
- [14] 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.
- [15] K. M. Hall, "An  $r$ -Dimensional Quadratic Placement Algorithm," *Management Science*, Vol. 17, pp.219-229, 1970.
- [16] C. L. Hu and M. S. Chen, "Adaptive Balanced Hybrid Data Delivery for Multi-Channel Data Broadcast," *Proceedings of IEEE International Conference on Communications*, pp. 960-964, 2002.
- [17] 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.
- [18] T. C. Hu and A. C. Tucker, "Optimal Computer Search Trees and Variable-length Alphabetic Codes," *SIAM Journal of Application*, Vol. 21, No. 4, pp.514-532, 1971.
- [19] 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.
- [20] 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.
- [21] T. Imielinski and B. R. Badrinath, "Mobile Wireless Computing: Challenges in Data

- Management,” *Communication of ACM*, Vol. 37, No. 10, pp. 18-28, 1994.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] S. H. Lim and J. H. Kim, “Real-time Broadcast Algorithm for Mobile Computing,” *ACM Journal on Systems and Software*, Vol. 69, pp. 173-181, 2004.
- [27] S. C. Lo and A. L. P. Chen, "Optimal Index and Data Allocation in Multiple Broadcast Channels," *Proceedings of IEEE International Conference on Data Engineering*, pp. 293-302, 2000.
- [28] W. C. Peng and M. S. Chen, “Dynamic Generation of Data Broadcasting Programs for Broadcast Disk Arrays in a Mobile Computing Environment,” *Proceedings of ACM International Conference on Information and Knowledge Management*, pp. 35-45, 2000.
- [29] W. C. Peng, J. L. Huang, and M. S. Chen, “Dynamic Leveling: Adaptive Data Broadcasting in a Mobile Computing Environment,” *Mobiles Networks and Applications*, Vol. 8, pp. 355-364, 2003.
- [30] N. Shivakumar and S. Venkatasubramanian, “Energy-efficient Indexing for Information Dissemination in Wireless Systems,” *ACM Journal of Wireless and Nomadic Application*, 1996.

- [31] 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.
- [32] K. Stathatos, N. Roussopoulos, and J. S. Baras, "Adaptive Data Broadcast in Hybrid Networks," *Proceedings of International Conference on Very Large Data Bases*, pp. 326-335, 1997.
- [33] 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.
- [34] 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.
- [35] 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.