南　　華　　大　　學

資訊管理學系

碩士論文

以漸近基礎運用於無線廣播排程資料佈置之研究

Relaxation-Based Data Placement for Wireless Broadcast

研 究 生：盧奎佑

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

Relaxation-Based Data Placement for Wireless Broadcast

研究生：

經考試合格特此證明

口試委員：

阮金聲

指導教授：

所　　長：

口試日期：中華民國 93 年 6 月 23 日

誌　　　　　謝

提筆至此，回首以往，心中五味雜陳，要感謝的人實在太多；父母的養育之恩、師長們的教導、兄姐的支持及朋友的惕勵等等，本人由衷感激，僅於此以寥寥數語表達無限的謝意。

盧奎佑　謹識

于南華大學

民國九十三年六月

以鬆散基礎應用於無線廣播資料排程佈置之研究

學生：盧奎佑　　　　　　　　　　　　　　指導教授：吳光閔

南華大學資訊管理學系碩士班

摘　　要

在無線環境中，使用廣播的方式來傳送資料不僅可以節省頻寬，同時也可以降低伺服端資源的使用。由於無線廣播資料排程的順序將影響客戶端接收資料的時間，伺服端如何決定廣播頻道上資料的佈置在無線環境中是重要的議題之一。

本研究運用漸近基礎的方式安排資料在廣播排程上的順序。實驗證明本研究所提出的方法較之前別人所提的方法其效果要來的好。

關鍵字：無線廣播、存取時間、資料佈置

Relaxation-Based Data Placement for Wireless Broadcast

Student: Kuei-Yu Lu Advisor: Dr. Guang-Ming Wu

Department of Information Management

The M.B.A Program

Nan-Hua University

ABSTRACT

Broadcasting is an efficient method for resolving the bandwidth limitation in wireless environment. Due to the ordering of data items on a broadcast channel affects the clients' access time, how to decide the data placement in broadcast channel to reduce the clients' access time is an important issue in wireless environment.

In this paper, we apply a *Relaxation-Based* method to the data placement of broadcasting. A comparison to previously published algorithms shows that the results of our algorithm are superior.

Keywords: Wireless Broadcast, Access Time, Data placement

Table of Contents

Table of Tables

Table of Figures

Chapter 1 Introduction

In this chapter, we describe the environment of wireless broadcast. Some problems are discussed usually and are solved. Illustrate our approach and list the research framework of this paper.

In the life has the more and more installments utilization wireless communication technology; For example, cellular phone, PDA, laptop and so on. The advantage of utilizes the wireless communication is enhanced the convenience and extremely portable. The wireless communication sometimes is congested because of the bandwidth short. In wired transmission also has the bandwidth insufficient situation occurrence as in transmits the multimedia material. In [21], the research proposed broadcasting to the specific group and layered video transmission to solve the video-on-demand bring about the congested.

Some researches discuss the allocation of broadcast data items in the disk of server [13][20][14]. They put the popular data times in the disk which access speed quicker and unpopular in slow one to improve the effect of access in broadcast data items. The clients access the data items on broadcast channel to regard as access on disk. Then, they called "Broadcast Disk". In [5][13][14][15], they study the replace question of broadcast data items when the clients have cache. The client keeps the data that more commonly used or nearer use in cache. Then the clients need not to receive on broadcast channel when the demand data items in cache. Above this kind of research all is discusses the arrangement and depositing of broadcast data items in storage.

In Fig. 1-1, the broadcast scheduler server receives the queries of clients, through the schedule processing, and then gets the order of data broadcasting. Broadcast media transmits the data, and then the clients receive the data on the broadcast channel. Each client keeps the data what it requests and discards the data that it not demands. The data items are broadcasted according to the order of the schedule on broadcast channel. This broadcast schedule repeats broadcasting.

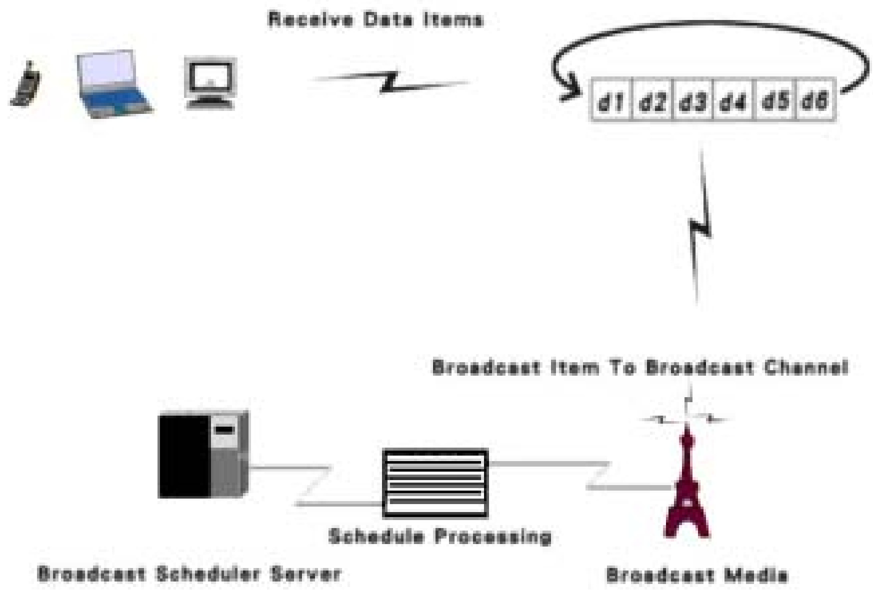


Figure 1-1 The environment of wireless broadcast.

In wireless environment, used broadcast way can effective to solve the bandwidth short is approved. Many researches discussed with the problem when using the broadcast method to solve the insufficient bandwidth [2][3][5][7][11][12][14][15][17][18].

In [2][3][7][11], they consider the data items exist dead-line constraint. The dead-line problem mostly occurs on the real-time system. In multi broadcast channel environment, the dead line data-items put on on-demand (pull) channel and others put on broadcast channel (push). Some researches consider in multi broadcast channel, the broadcast cycle in each broadcast channel can be equal or difference [7][12][14]. Mostly, the popular data items put in the shorter broadcast cycle to reduce the client receive data items on wait. In

[6], they consider the query in rang, that is, request by a range. And in [1], they consider only two data items be broadcasted. Decide the popular one and unpopular one how to placement on broadcast channel to reduce the client wait for data items. In [10], they consider only one request in a query. They want to decide a broadcast schedule to reduce the client receive data items on broadcast channel. There are so many differences considering on data placement for wireless broadcast.

The broadcasting is frequently applied to server to save bandwidth. When server adopts this way, a client will care about how long of time it can receive the data that it wants and its power consumption. Therefore, there are two important issues: *Access Time* and *Tuning Time* related with wireless broadcasting. The *Access Time* is the amount of time from a client submits a query to the server to its interests data received. The *Tuning Time* discusses about how long of time that a client listening to the broadcast. It can be divided into two modes during a client listening to the broadcast channel. When the clients are listening to the data items, we called they operating in the active mode during which it spent much more power. Another mode is called doze mode when the requested data items have not arrived. Express *Access Time* and *Tuning Time* in Fig. 1-2. The *Index* recording the information that the data items when will be broadcasted. And the *Data* means the client requests data items. Assume there is a query to request $D2$ and $D3$. With start to receive data items on broadcast channel when the server broadcasting I_2. Then the period from start to end is called *Access Time*, but *Tuning Time* is for receives I2 to add on $D2$ and $D3$ in this paragraph.

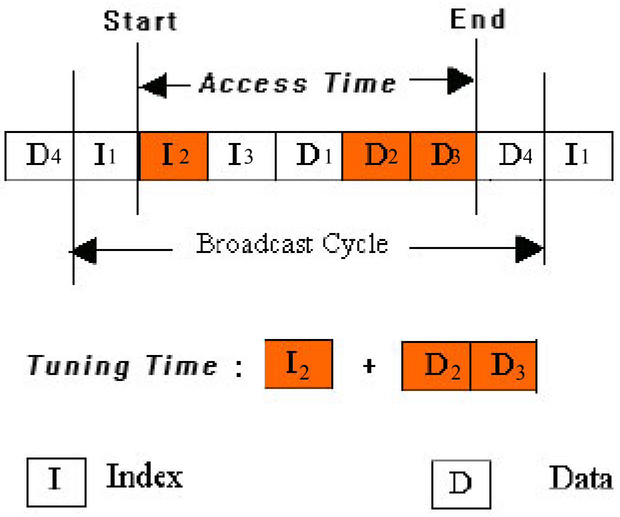


Figure 1-2 The example of *Access Time* and *Tuning Time*.

There have been some researches on *Tuning Time* by appended *Index* to broadcasting schedule [9, 12, 18, 23]. These researches create B+ tree and apply search trees' skills to construct model, shown as Fig. 1-3. The data items are the position in the leaf nodes and the internal node store *Index*.

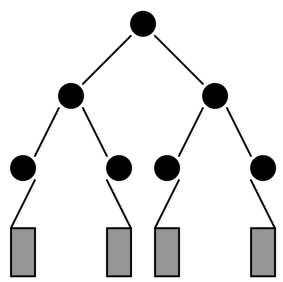


Figure 1-3 The approach for *Tuning Time* by B+ tree.

The *Index* informs clients that the *Data* on the broadcasting channel will arrive. So that the clients can choose on active mode to prepare for receiving *Data* or they can choose doze mode to save the battery consumption.

However, appended *Index* in a broadcast schedule will increase the clients' *Access Time*. By contrast, if the clients received that they wanted, they wouldn't have to listening the broadcast channel anymore. The clients gain the wanted data earlier. Therefore, we focus on deciding the data placement to reduce the clients' *Access Time*.

Several related researches on *Access Time* was proposed [4], [19], [24] and [26]. In [19], they show how to minimize the average response time given multiple broadcast channels by optimally partitioning data among them. And also offer an approximation algorithm that is less complex than the optimal for a wide range of parameters.

In [4], it proposed optimal linear ordering algorithm to determine the allocation of the data on single broadcast channel. It believes that frequently co-accessed data are not only allocated close to each other, but in a particular order which optimizes the performance of query processing.

In [26], it based on analyzing the history of queries (i.e., the chronological sequence of items that have been requested by clients) using data mining techniques to decide the broadcast schedule. Through process data compilation, analysis, and designing association rules. According to the association rules to decide which data items put on the same place to get the broadcast schedule. For example, analyze on the queries, the supposition requests the A data item, then simultaneously also institutional records of a reign B data item. If the supposition established, then A and B connected together to reduce the *Access Time*. However, ponders the causal relation then establishment association rules by the result is a completely difficult method, and also the association rules have to spend the long time to make the analysis to the queries.

Because the statistical analysis more difficult and time-consuming. Therefore, some researches use to analyze in batches, namely each time considers a part to decide the broadcast schedule. In [24], they used *QEM* method to construct the broadcast schedule by each query's data set appended to decrease the *Access Time*. It first considers the frequency highest query and frequency high. Finds the connection stronger data items and put on the same place. Afterwards, pursues frequency highest query from does not have the consideration to advance considered. That is, the higher-frequency query takes precedence over the lower one. The higher-frequency query appended to broadcast schedule first. When appending to the broadcast schedule, lower one can't affect previously processed and make sure that each query appended to the broadcast schedule always minimizes the *Access Time* as much as possible. In spite of their approach considered the relationship between data items when a query access more than one data object, they only think of the data of queries appended to the broadcast schedule one by one. That is, their shortcoming is not considering relationship of whole data at one time.

In this paper, we propose an algorithm, named *Relaxation-Based Placement*, which consider the relations of the entire data items. Our work is to decide a broadcast schedule to make the sum of *Access Time* smaller much as possible This research assumes the broadcast channel is a single channel, the data items not repeat in single broadcast cycle, the size of data items are equally, and a query request several data items. This approach transforms the entire query patterns into flow network. The data items of query patterns are regarded as the vertices of the flow network. Query patterns are regarded as the edges of the flow network and the frequency are regarded as the weight of edges. We use the maximum-flow algorithm with a min-cut [16], which divided data items into two parts. The relation is not closed between two parts. That is, the relations of data items are more closed in each part. Then the data items are going to together by the strong of correlation between each other. We believe that the clients' *Access Time* will be reducing through considering the relations of entire data items to placement. Experimental results show that our approach is better than *QEM* [24] for reducing clients' *Access Time*.

This paper is organized as follows. Chapter 2 describes problems formulation. We present solutions in Chapter 3. In Chapter 4, we introduce into *Preceding Process*. Chapter 5 shows the simulations. Finally, concludes this work in Chapter 6.

Chapter 2 Problem Formulation

This chapter defines the problem of this research. Introduce how the question does produce, appraised the method and show this problem belong to NP-complete.

We assume that wireless computing environment consists of many clients and one server. The server analyzes the data access patterns of the clients' requests and resolves an order of a schedule on the data items broadcast. We assume the broadcast channel is a single channel, the data items not repeat in single broadcast cycle, the size of data items are equally, and a query request several data items. The order of data items broadcast will affect clients' *Access Time*. For example, assume the broadcast schedule is formed as <d_1, d_2, d_3, d_4, d_5, d_6>, then a client wants to retrieving d_2 and d_4 (Shown as Fig. 2-1).

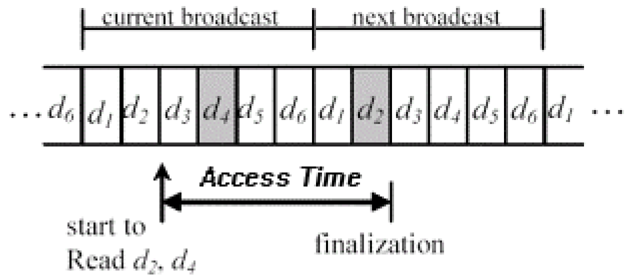


Figure 2-1. A query onto the wireless broadcast.

As in Fig. 2-1, that the order of broadcast is d_1, d_2, d_3, d_4, d_5, and d_6. A client submits its query when the server broadcasting d_3. If the client requests d_2 and d_4, then the *Access Time* of this query equals 6. (Shown as Fig. 2-3) If the order of broadcast is changed from d_1, d_2, d_3, d_4, d_5, and d_6 to d_1, d_5, d_3, d_4, d_2, and d_6 as Fig. 2-2. The *Access Time* equals 3 when the other conditions are not changed. Thereby, we know that the order of broadcasting will affect the *Access Time* of client.

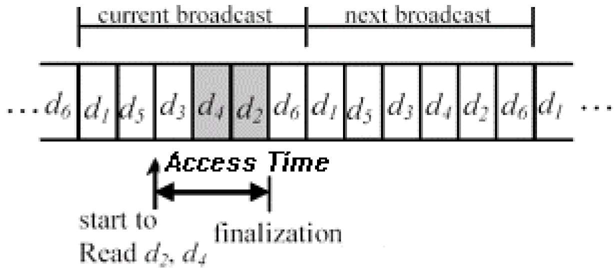


Figure 2-2. The of broadcast schedule is changed from d_1, d_2, d_3, d_4, d_5, and d_6 to d_1, d_5, d_3, d_4, d_2, and d_6.

We extend the *Access Time* to get the problem that we want to solve. Assume there are two queries, query 1 and query2. The query 1 requests d_2 and d_3; query 2 requests d_4 and d_5. The broadcast schedule is d_1, d_2, d_3, d_4, d_5, and d_6. Both queries are submitted when the server broadcasting d_1. The *Access Time* of query 1 equals 3 and the *Access Time* of query 2 equals 5. The sum of *Access Time* in two queries equal 8.

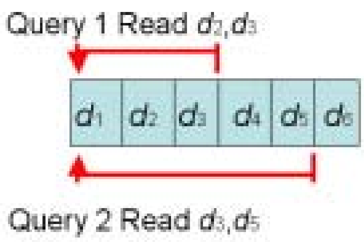


Figure 2-3. Both queries are submitted when the server broadcast d_1.

If we change the order of form d_1, d_2, d_3, d_4, d_5, d_6 to d_1, d_5, d_3, d_4, d_2, d_6. The *Access Time* of query 1 has increased from 3 to 5. Query 2 has decreased form 5 to 4. Shown as Fig. 2-4. The sum of *Access Time* in two queries equals 9. In this case, we know the benefits between queries maybe conflict. As decreases the *Access Time* of query 2 but increases the *Access Time* of query 1.

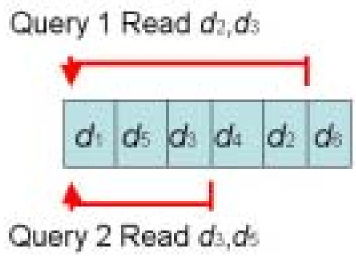


Figure 2-4. The *Access Time* of query 1 has increased from 3 to 5. Query 2 has decreased form 5 to 4.

Then, it is so complex and decides hard. Here, bases on the integral benefit to make the broadcast schedule. Our work is to decide a broadcast schedule to make the sum of *Access Time* smaller much as possible.

Table 2-1. Symbol definitions

Notation	Meaning
d_i	a data item to be broadcasted
D	the set of data items d_i; $\{d_1, d_2, \ldots, d_N\}$
q_i	a query that is issued on the broadcast data stream
$QDS\ (q_i)$	the set of data items that q_i accesses
$freq\ (q_i)$	the frequency of query q_i
Q	The set of queries q_i; $\{q_1, q_2, \ldots, q_M\}$
σ	the broadcast schedule of D

Table 2-1 defines some notations [24] for problem definition. The server needs to arrange the the data items on the broadcast channel to minimize the total *Access Time* (*TAT*), denoted by [24]:

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

where $AT^{avg}(q_i, \sigma)$ denote the average *Access Time* of query q_i in the broadcast schedule σ. In [24], a new measure, called Query Distance (*QD*), which is used of the data items a query accesses to show the degree of coherence.

Definition 1[24]

Suppose DQS(q_i) is {d_1, d_2, ..., d_n}, and δ_j is the interval between d_j and d_j+1 in broadcast schedule σ. Then the QD of q_i on σ is defined as:

$QD(q_i, \sigma) = B - \max(\delta_k)$ *for* $k = 1 \sim n$,

where B is the size of a broadcast cycle.

For example, we assume a broadcast schedule σ = <$d_1, d_2, d_3, d_4, d_5, d_6$>(as shown in Fig. 2-5). The *B* is equal to 6. There is a query q_i with *QDS* (q_i) = {d_2, d_4}. In Fig. 2-5, δ_1 is equal to 1 and δ_2 is equal to 3. So the *QD* (q_i, σ) is 6 - 3 = 3.

Definition 2[24]

Given a query q_i and two schedules σ_1 and σ_2,

$$\text{If } QD(q_i, \sigma_1) \geqq QD(q_i, \sigma_2) \text{ then } AT^{avg}(q_i, \sigma_1) \geqq AT^{avg}(q_i, \sigma_2)$$

Therefore, the summation of *QD* (q_i, σ)$\cdot$ *freq*(q_i) is denoted *TQD*(σ), where $q_i \in Q$. The broadcast scheduling problem redefine to minimizes the *TQD*(σ).

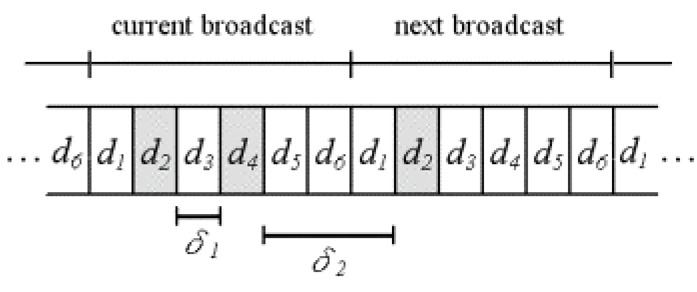


Figure 2-5. An example for δ_i in broadcast schedule.

Definition 3[24]

Given a set of data objects D and a set of queries Q, the wireless data placement problem is to find a broadcast schedule σ_i *such that TQD(*σ_i*) is minimum among all possible* σ_i*, i = 1, …*

Theorem 1[24]

The wireless data placement problem in Definition 3 is NP-complete.

Proof: [24]

The proof follows transformation from Optimal Linear Arrangement Problem (OLAP) [8] that is known as a NP-complete problem. OLAP is to find a one-to-one mapping function f such that $f : V \rightarrow \{1, 2, \ldots, |v|\}$, $\Sigma \{u, v\} |f(u) - f(v)| \leqq k$ in a graph $G = (V, E)$, where k is a positive constant. Let's assume that each query accesses only two data objects, the size of each data object is equal and the reference frequency of each query is equal. Then the problem is transformed into OLAP by considering the data object as the node in the graph and the query as the edge in the graph. For details, see the reference [25].

Chapter3 Relaxation-Based Placement

This chapter describes the solution in this research. Bases on consider at relation of the data items to make the broadcast schedule. Different with others, the approach of this research considers the relation of complete data items once but not consider partial data items in difference times.

3.1 Relaxed Placement

We use constraints release to make the complex problem simpler. We ignore the constraint of time slot to get rough solution that the data's relationship stronger at the same time slot (Shown as Fig. 3-1). It is an infeasible schedule on single broadcast channel that disallow broadcast data more than one at the same time. Therefore, the schedule need to legalization to get physically feasible solution.

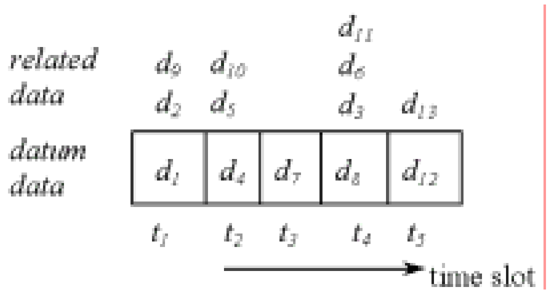


Figure 3-1. Relaxation is ignoring time slot constraint.

Our method is through compares the force from left hand and right hand on each data items to decide the position of data items. The total data item is divided into two parts: the first part is called *datum data* that are to be datum points and the second part is called *related data* that are belong to *datum data* by relationships. We transform the entire query pattern into flow network. We use the maximum-flow algorithm with a min-cut to partition to get *Relaxed Placement* as Fig. 3-2. We model by a graph $G = \{V, E\}$, where V is a set of vertices and E is a set of edges. $V = \{V', D', s, t\}$. $V' = \{v_1, v_2, \ldots, v_N\}$, which v_i corresponding with d_i. $D' = \{d_{11}, d_{12}, d_{21}, d_{22}, \ldots, d_{M1}, d_{M2}\}$, which $d_{i1}$and d_{i2} are created q_i. That is, each query pattern introduces two dummy nodes. Vertex s and vertex t present the source and sink individually in flow network. Directed edge (u, d_{i1}) and edge (d_{i2}, u) with capacity ∞, where $u \in QDS(q_i)$. D_{i1} and d_{i2} with an edge (d_{i1}, d_{i2}) which capacity is *freq* (q_i). The d_i, q_i, QDS (q_i) and the *freq* (q_i) had defined in Table 2-1. Below proposes Fig. 3-3 to make a description. Assume there are two queries the supposition has only two queries, query 1 and query 2. Query 1 requests d_1, d_3, and d_4; Query 2 requests d_1, d_3, and d_5. Model the flow network by the method in this research and show the flow network as Fig. 3-3. In Fig. 3-3, s represents source, t represents sink. The d_1 and d_5 are *datum data*. The net by Q_1-a and Q_1-b corresponds query 1, the net by Q_2-a and Q_2-b corresponds query 2.

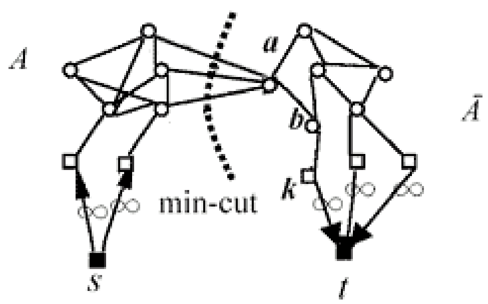


Figure 3-2. The min -cut size is three when the weight is one of all edges.

Because on the flow network too is complex above, therefore below we use Fig. 3-2 to explain our method. Given a set $F \subset V$, which F stands for *datum data*. We call the f_i *Fixed node*, $fi \in F$, form F we determine the set M, that is $M = (V - F)$. We call the m_i

Mobile Node, $m_i \in M$, which M stands for *related data*. We add directed edges to s and t with capacity ∞which connect *Fixed node* and source(sink) nodes(Shown as Fig. 2-1).

A min-cut is a two partition $(A, \bar{A})$ of G with $s \in A$, $t \in \bar{A}$ such that the size $c\ (A, \bar{A}) = \Sigma\ c\ (u, v)$ is minimized, where $c\ (u, v)$ is the capacity of edge (u, v). If the frequencies in all query patterns are one unit, the min-cut size equals one in Fig. 3-2. Every *Fixed Node* is connected from s in partition A and every *Fixed Node* connects to t in partition $\bar{A}$.

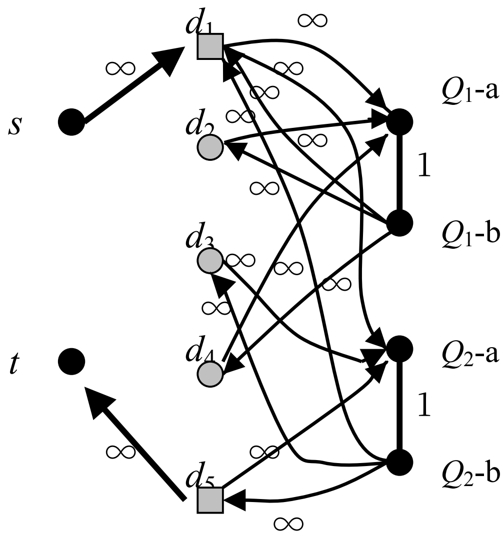


Figure 3-3. Example for constructs the flow network by the method in this research.

Assume that we move node k from $\bar{A}$ to A, the edge from node k to t will be removed and an edge form s to k will be created. As illustrated in Fig. 3-4, there is a new min-cut cause the flow is changed in the G. The new min-cut size equals two. The *Mobile Node a* and *Mobile Node b* follow node k from $\bar{A}$ to A in virtue of the relations with node k are stronger than other *Fixed Node* (k to a and b as d_4 to d_5 and d_{10} in Fig. 3-1).

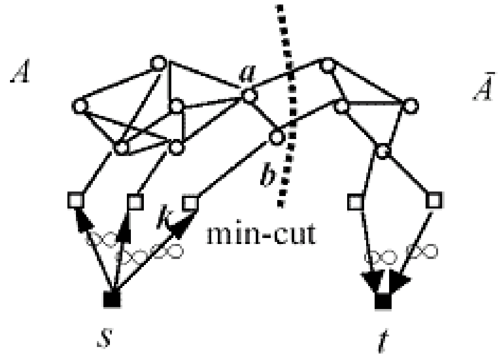


Figure 3-4. A new min-cut after extracts *Fixed Node* from $\bar{A}$ to A executes min-cut from the case shown in Fig. 3-2.

3.2 Placement Legalization

A *Relaxed Placement* results many data items at the same time slot or some time slot haven't data item. We should be resolved to get a feasible placement to broadcast.

For a data placement P, let $P[i]$ denote the data item placed in the position of time slot t_i and $P^{-1}[v]$ denote the position of time slot of data item v. Let S be a set of coincident nodes and n_i a set of nets adjacent to node i. Suppose the x_l is the position of the time slot for each node in S in the *Relaxed Placement*. Set l_e be the left limit time slot of a net e and L_i be a subset of n_i such that $L_i = \{e \mid e \in n_i \text{ and } l_e < x_l\}$. Similarly, set r_e be the right limit time slot of a net e and R_i be a subset of n_i such that $R_i = \{e \mid e \in n_i \text{ and } r_e < x_l\}$. We use *F-value* which subtract the amount of L_i from the amount of R_i to get a relative order among coincident nodes. The *F-value* of node higher the places more right and randomly rearrange them if the nodes having the same. We show the *Relaxation-Based Placement* as follows:

Algorithm *Relaxation-Based Placement (RBP)*

Input: Q : the set of queries q_i and control value i_0

Output: placement *P'*

```
Initial placement P at random;
Itimes ← i0;
while (Itimes > 0) {
  Left ←  Φ;
  Random select M from D; // D = ∪_{qi ∈ Q} QDS(qi)
  F ← D \ M;
  Model the G = {V, E} corresponding to Q;
  while(F ≠ Φ){
    Extract fi from F by P ordering; // draw out the fi from Ā to A in G
    Delete edge (fi, t);
    Add edge (s, fi) with capacity ∞;
    T ← Find leftmost min-cut using a max-flow algorithm;
    T ← T \ F;
    related data ← T \ Left;
    xv ← P^-1[fi], ∀v ∈ related data;
    Left ← T;
  }
  P' ← placement legalization by F-value;
  P ← P';
  Itimes--;
}
```

Due to the order of input data items can affect the *TQD* (*Q*) by our approach. We set the control value i_0 to determine how many times to execute. The Fig. 3-5 shows an example of our approach. The input data show as Fig. 3-5. There are four queries and an initial *P* as <d_1, d_2, d_3, d_4, d_5, d_6, d_7>. Assume the frequency of each query equal one, so the *TQD* (*Q*) = 3 + 3 + 5 + 4 = 15. The d_1, d_3, d_4, and d_5 are *Fixed Node* and others are *Mobile Node*. A *Relaxed Placement* found after extracts *Fixed Node* from $\bar{A}$ to *A* executes min-cut one by one in Fig.3-6. We resolve to get a physically feasible placement for broadcasting schedule by *F-value* shown as Fig. 3-7. The *TQD* (*Q*) = 3 + 3 + 5 + 3 = 14.

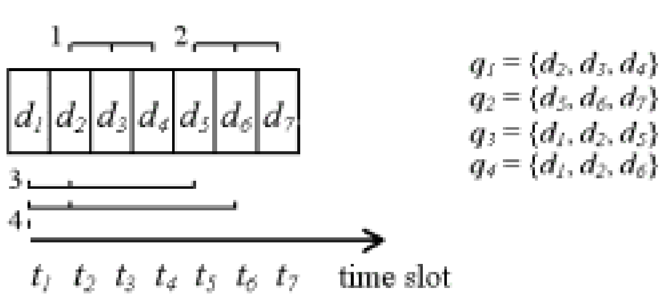


Figure 3-5. Initials placement *P* and the queries of clients.

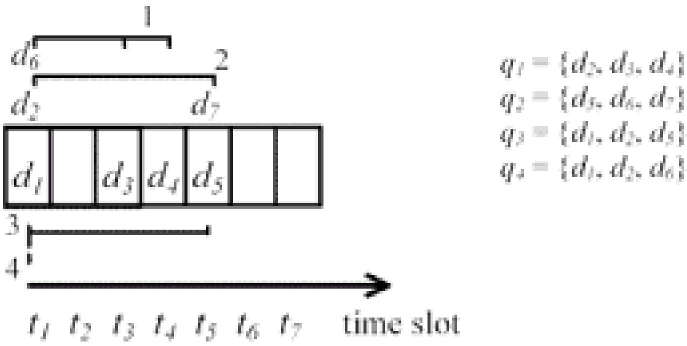


Figure 3-6 Acquire *Relaxed Placement* by extracts *Fixed Node* from $\bar{A}$ to A executes min-cut one by one.

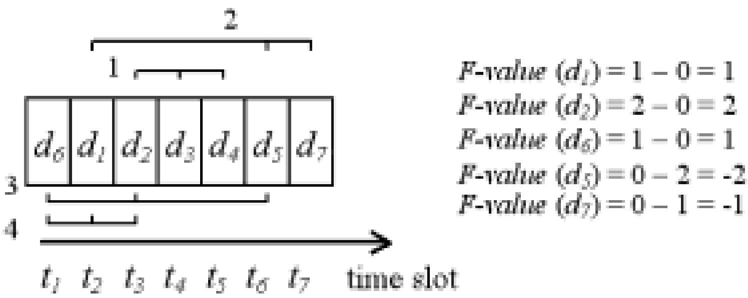


Figure 3-7. Make placement legalization by *F-value.*

When obtains a broadcast schedule difference with the initial and run again the algorithm in this research that will get the difference broadcast schedule. Even if in the same condition, the difference schedule will cause the difference result. Each time will carry out a broadcast schedule to retain for the next execution initial broadcast schedule. When the initial broadcast schedule better, obtains according to *F-value* the arrangement will be able to be better.

Chapter 4 Preceding Process

Some data items all simultaneously appear in queries, Bases on the relation placement that these data items should arrange together to reduce the total *Access Time*. But, the minimum-cut maybe separate them and then cause the *Access Time* growth. This chapter describes this kind of situation and proposes the solution.

4.1 Origin

This research used minimum-cut to make the connection stronger data items in the same place. The discovery of this research, some data items all together leaves in present query, but the minimum-cut has not actually turned over to these data items in the same place. For example, the supposition has only two queries, query 1 and query 2. Query 1 requests d_1, d_3, and d_4; Query 2 requests d_1, d_3, and d_5 (Show as Fig. 4-1). Model the flow network by the method in this research and show the flow network as Fig. 4-2. In Fig. 4-2, the s represents source, and the t represents sink. The d_1 and d_5 are *datum data*. The net by Q_1-a and Q_1-b corresponds query 1, the net by Q_2-a and Q_2-b corresponds query 2. Finds the Maximum Flow Path F by the maximum flow algorithm. The most left side minimum-cut cuts the net Q_1-a and Q_1-b. It means no data item follows d_1 when d_1 was extracted from $\bar{A}$ to A. But, the relation of d_4 and d_1 is stronger than the relation of d_4 and d_5. The d_4 should place with d_1.

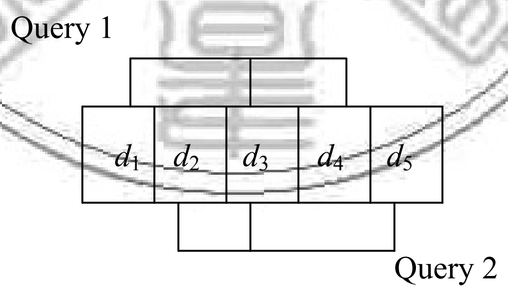
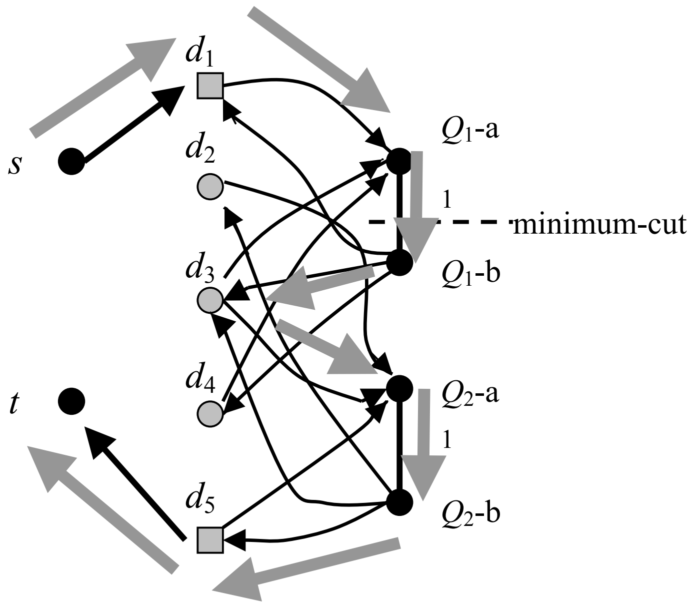


Figure 4-1. An example for *Preceding Process.*



Maximum-Flow path F : $s \rightarrow d_1 \rightarrow Q_1\text{-a} \rightarrow Q_1\text{-b} \rightarrow d_3 \rightarrow Q_2\text{-a} \rightarrow Q_2\text{-b} \rightarrow d_5 \rightarrow t$

Figure 4-2. The d_4 is not belongs to the datum data d_1 by minimum-cut.

4.2 Process

In order to solve the problem, which on raises front, this research joins the mechanism that we call that *Preceding Process*. Analysis the entire queries first, if discovered the data items all simultaneously appear in same queries, cluster these data items together. Chooses any data item to represent the cluster from whole data items in this cluster. After this processing, then use the *Relaxation-Based* algorithm in this research to get the *Relaxed Placement*. Uses *F-value* to get the feasible placement in single broadcast channel. During the process, which in using *F-value,* if discovered has the data item is the cluster representative, then spread the data items which in this cluster. Makes the data items in this cluster arranged one by one in the broadcast channel. Then, the problem which data items all simultaneously appear in same queries are not together is solved.

In Fig. 4-3, cluster the d_1 and d_4. The connection will be changed by cluster. Extend the concept to cluster when the data items occur in the same query completely. Example shows as in Fig. 4-4 and Fig. 4-5.

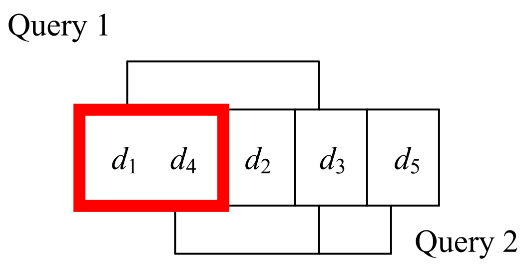


Figure 4-3. Cluster d_1 and d_4 to be a set.

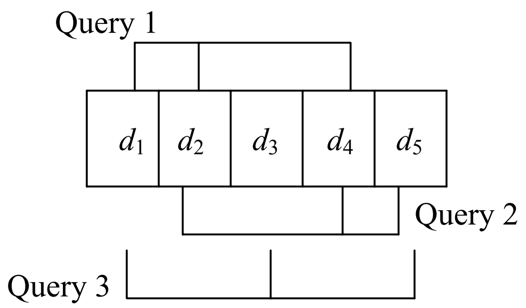


Figure 4-4. The d_2 and d_4 occur in the same query completely.

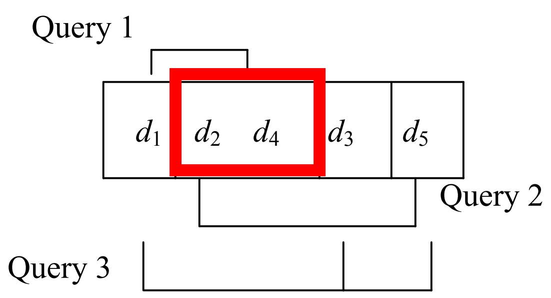


Figure 4-5. Cluster d_2 and d_4 to be a set.

Chapter 5 Simulations

In this chapter, simulates the broadcast environment in this research. Compares the performance of our approach with the other mean. Tests the different parameter value to experimental influences.

5.1 Experiment Environment

We implement the *RBP* algorithm and *QEM* approach by Java language. Compare the clients' *Access Time* of them through simulations. We tested these programs on P4 2.0 GHz ASUS L3 with 240 MB RAM. In server setting, the server broadcast data in one broadcasting channel and there is no replication in a single broadcasting schedule. In client setting, a query accesses over two data items and the amount of data items in each query to be equal. The *QDS* (q_i) are mutually independent and product the data items by random.

5.2 The Affect of Parameter in Experiment

This section tests the effect of parameters in our approach. First, we change the number of fixed point changed to influence the performance. Discovered by way of the experiment, the number of fixed-point change will influence the *Access Time* and the formula execution time. The results show as Fig. 5-1, Fig. 5-2, and Fig. 5-3.

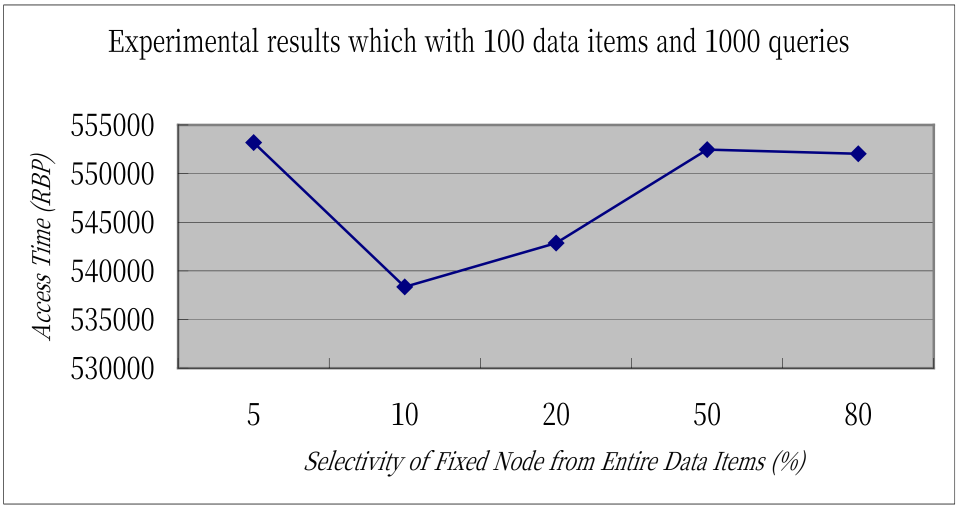


Figure 5-1 Experimental results which with 100 data items and 1000 queries.

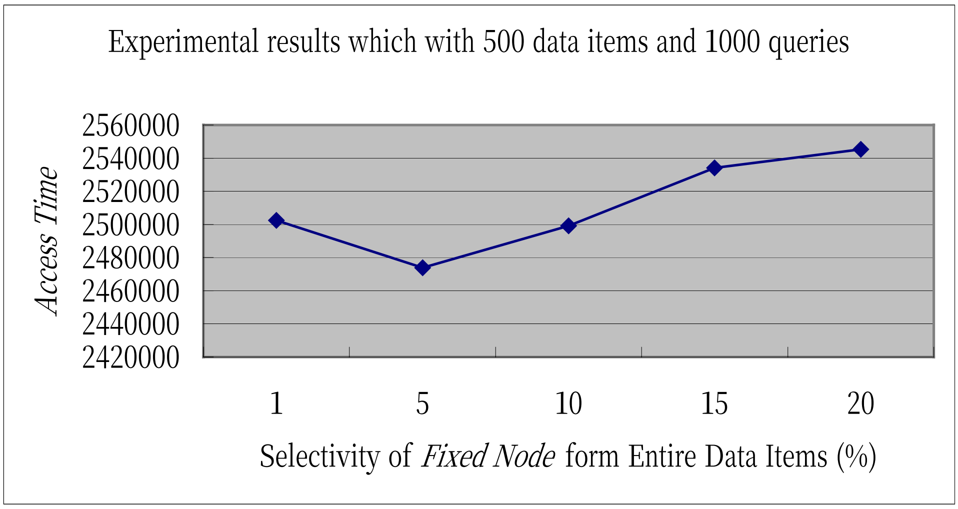


Figure 5-2 Experimental results which with 500 data items, 1000 queries and 1% selectivity.

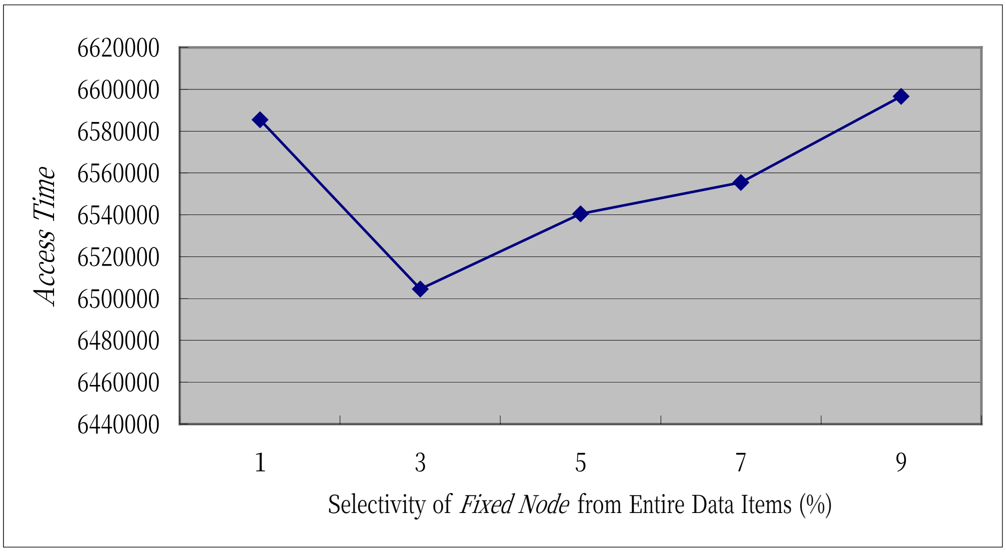
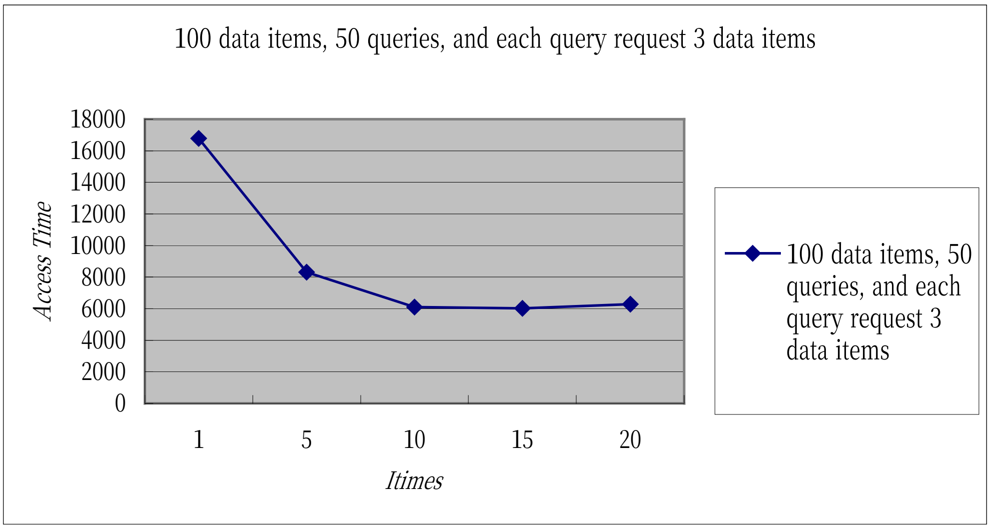
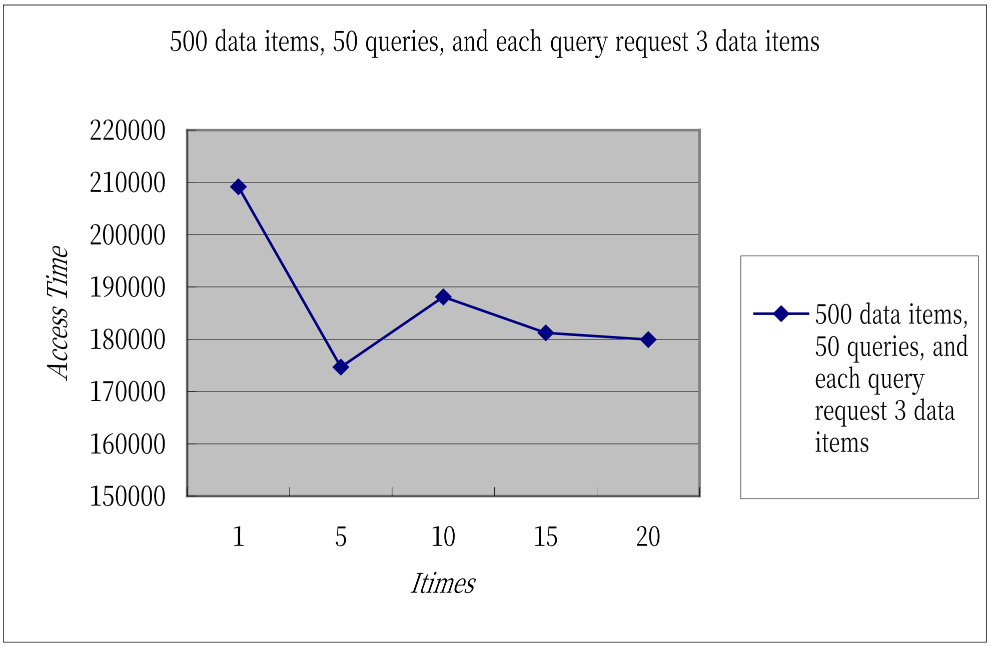


Figure 5-3 Experimental results which with 1000 data items, 1000 queries.

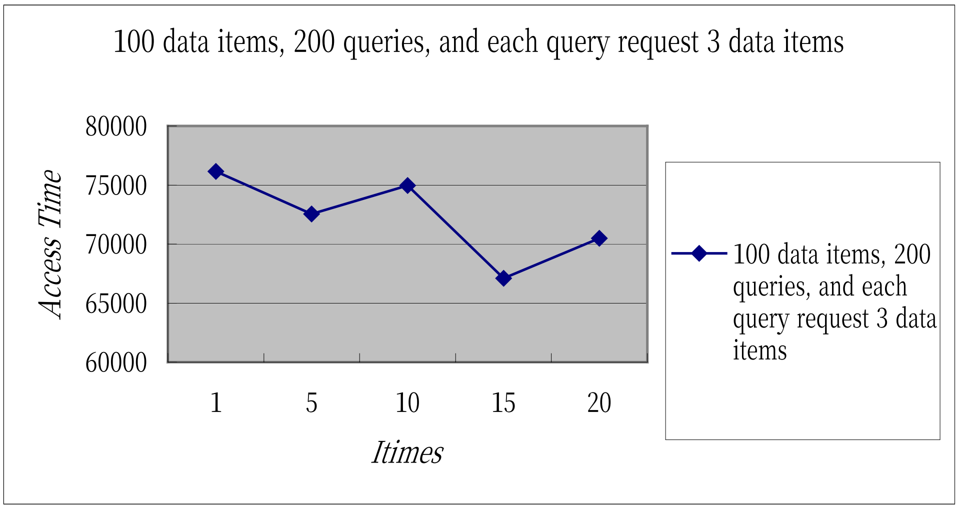
Fig. 5-1 shows when selective 10% from entire data items to be *Fixed Node*, the *Access Time* is shorter than others. We can image that the more *Fixed Node*, the more random placement on the broadcast schedule. Fig. 5-2 shows the *Access Time* of 1% selectivity longer than the *Access Time* of 5% selectivity, as less of *Fixed Node* to make most data items set by *F-value* and the *Relaxed Placement* doesn't work well. Fig. 5-3 shows the *Access Time* of selectivity around 1% to 9% not change more.



(a)



(b)



(c)

Figure 5-4 Carries out the number of Times to the Access Time influence.

In Fig 5-4, the *Access Time* of first execute are long than others. Rough says the entire tendency is convergence. The *Access Time* restrains to the certain degree then not converges anymore and it has little up or down.

In Table 5-1, the *Access Time* by *PPRBP* is shorter than by *RBP*. But it's not absolutely.

In 100 data items, 50 queries, and each query request 10 data items, the *Access Time* by *RBP* is shorter than *PPRBP*.

Table 5-1 The Access Time by *QEM*, *RBP*, and *PPRBP*

	QEM	*RBP*	*PPRBP*
100 data items, 100 queries, and each query request 3 data items	28620	26510	25640
1000 data items, 100 queries, and each query request 30 data items	651620	686800	673030
100 data items, 50 queries, and each query request 10 data items	30340	30680	30740
1000 data items, 50 queries, and each query request 30 data items	242690	260560	242040

5.3 Experimental Results

The range of move nodes is 0.9 through 0.99 times the number of data items. We set control value i0 to 10. We select the shortest *TQD* (*Q*) among 10 times in every experiment. There are three parameters in our experiments. Individually, the number of queries, selectivity (the degree of *QDS* (q_i) over the size of broadcast data set in terms of percentage), and the number of queries are changed.

First, we change the number of queries while 100 numbers of the data items and 5% selectivity. The results are shown in Fig. 5-5. It shows the clients' *Access Time* by our approach is less than by *QEM* approximately 2.26% in average. In Fig. 5-6, we change the selective values while 100 numbers of data items and 1000 queries. The clients' *Access Time* by our approach is less than by *QEM* about 0.7% in average. The 25% selectivity means a query access 1/4 of the set of data items. Finally, we change the number of data items while 6% selectivity and 1000 queries in Fig. 5-7.

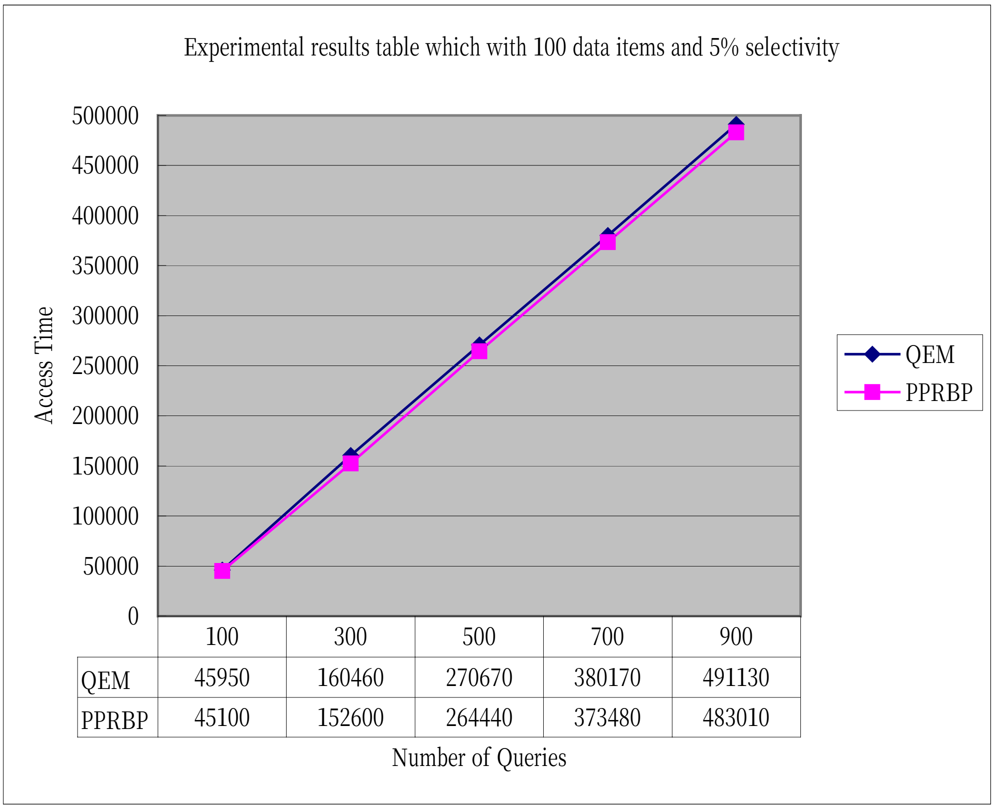


Figure 5-5 Experimental results table which with 100 data items and 5% selectivity.

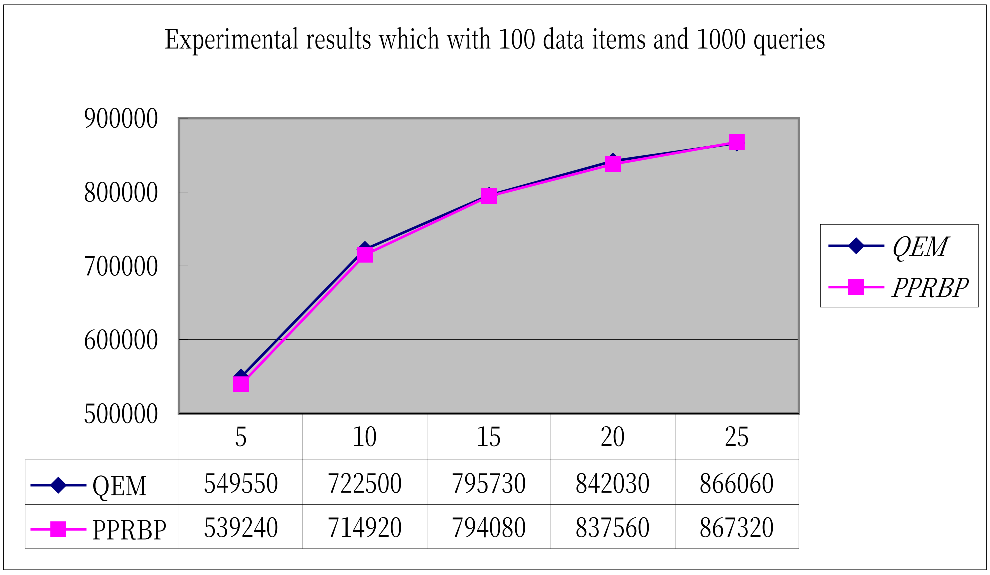


Figure 5-6 Experimental results table which with 100 data items and 1000 queries.

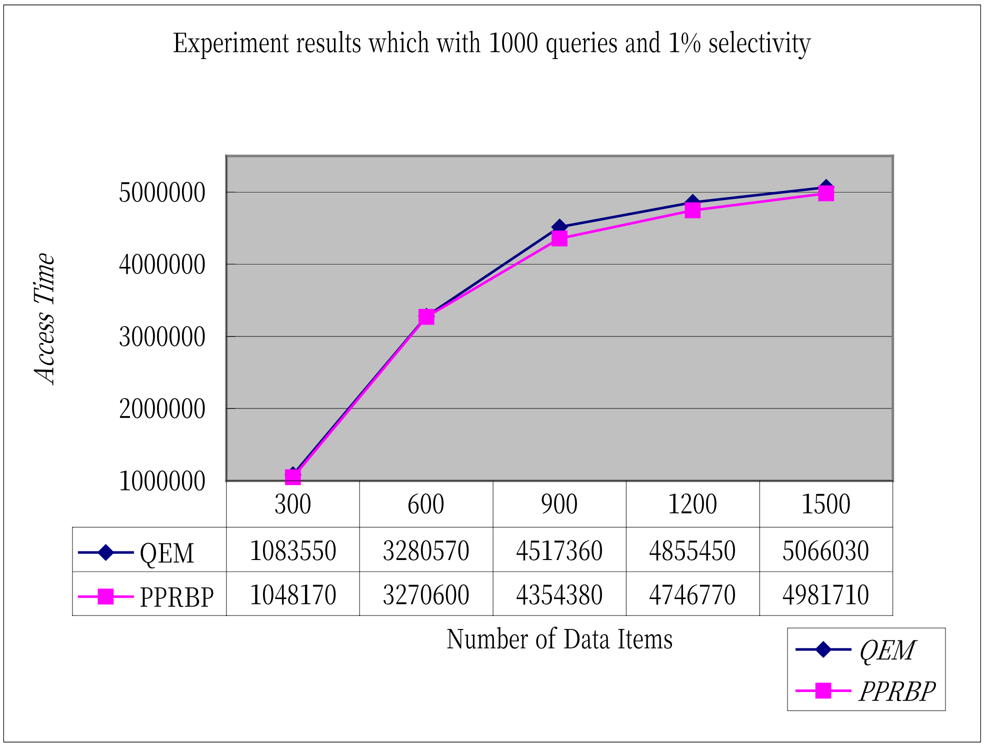


Figure 5-7　　Experimental results table which with 1000 queries and 1% selectivity.

Most of the experiments can be finished in less than 10 seconds. Generally, the clients' *Access Time* by using *RBP* is shorter than the clients' *Access Time* by using *QEM* in our simulations. Particularly the selectivity smaller is, the performance better is. This benefit is meaningful. We can imagine that when 90% selectivity or more, the order of broadcasting data will not be so important for clients to access the broadcast data items. On the other hand, no matter the number of queries or the number of data items was changed, it shows the benefit of our approach.

In order to save bandwidth and resource in wireless communication environment, the server tends to use broadcasting while the clients increased. The relations between data items are going stronger when the query patterns increase. Therefore, it is effectively by the relations between data items to decrease access latency in broadcast schedule in wireless broadcast environment.

Chapter 6 Conclusion

6.1 Conclusion

In this paper, we propose an algorithm; name *Relaxed Placement*, which consider the relationship of the entire data. Then, we use *F-value* to make the placement legalization. This approach improves the *QEM* only considers the relationships of the data items in each query. After the test simulations, Experimental results show that our approach is better than *QEM* for reducing clients *Access Time*. Through the simulations, it shown particularly the selectivity smaller is, the performance better is. Rough says the entire tendency is convergence. And *PPRBP* is better than *RBP* for reducing clients *Access Time*.

6.2 Future Research

In the future, we consider clustering the nodes based on each datum data items and its related data items to run *RBP* algorithm. This way can avoid the datum data items chosen by random that adjacent to each other no relationships. Application our *RBP* algorithm for multi-channel environment and non-uniform broadcasting environment is next work.

Reference

[1] Amotz Bar-Noy and Yaron Shilo, “Optimal Broadcast of Tow Files Over An Asymmetric Channel,” *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, 1999.

[2] Chi-Jiun Su, Leandros Tassiulas, and Vassilis J. Tsotras, “Broadcast Scheduling for Information Distribution,” *J.C. Baltzer AG, Science Publishers, Wireless Networks 5*, 1999.

[3] Giorgio C. Buttazzo, and Fabrizio Sensini, “Optimal Deadline Assignment for Scheduling Soft Aperiodic Tasks in Hard Real-Time Environments,” *IEEE Transactions on Computers*, 1999.

[4] Guanling Lee, Shou-Chih Lo, and Arbee L.P. Chen, “Data Allocation on Wireless Broadcast Channels for Efficient Query Processing,” *IEEE Transactions on Computers*, vol. 51, no. 10, 2002.

[5] Kam-Yiu Lam, Edward Chan, and Joe Chun-Hung Yuen, “Approaches for Broadcasting Temporal Data in Mobile Computing Systems,” *The Journal of Systems and Software 51*, 2000.

[6] Kian-Lee Tan and Jeffrey Xu Yu, “Generating Broadcast Programs That Support Range Queries,” *IEEE Transactions on Knowledge and Data Engineering*, 1998.

[7] Kiran Prabhakara, Kien A. Hua, and JungHwan Oh, “Multi-Level Multi-Channel Air Cache Designs for Broadcasting in a Mobile,” *Proc. IEEE International Conference on Data Engineering*, 2000.

[8] M.R. Garey and D.S. Johnson, “A Guide to the Theory of NP-Completeness,” *Freeman Publishing Company*, 1976.

[9] Ming-Syan Chen, Kun-Lung Wu, and Philip S. Yu,” Optimizing Index Allocation for

Sequential Data Broadcasting in Wireless Mobile Computing”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 1, 2003.

[10] Nitin H. Vaidya and Sohail Hameed, “Scheduling Data Broadcasting in Asymmetric Communication Environments,” *Kluwer Academic Publishers*, vol 5,1999.

[11] Sanjoy Baruah and Azer Bestavros, “Real-Time Mutable Broadcast Disks,” *Proceedings of RTDB'97: The Second International Workshop on Real-Time Databases*, 1997.

[12] Shou-Chih Lo and Arbee L.P. Chen, ”Optimal Index and Data Allocation in Multiple Broadcast Channels,” *Data Engineering*, 2000.

[13] Swarup Acharya, Michael Franklin, and Stanley Zdonik, “Dissemination-Based Data Delivery Using Broadcast Disks,” *Proceedings of the 22nd VLDB Conference Mumbai*, 1996.

[14] Swarup Acharya, Rafael Alonso, Michael Franklin, and Stanley Zdonik, “Broadcast Disk: Data management for Asymmetric Communication Environments,” *Proceedings of ACM SIGMOD International Conference*, 1995.

[15] Swarup Acharya, Michael Franklin, and Stanley Zdonik, “Disseminating Updates on Broadcast Disk,” *Proceedings of the 22nd VLDB Conference Mumbai*, 1996.

[16] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, *Introduction to Algorithms*, *Second Edition*, *MIT Press*, 2001.

[17] T. Imielinski, S. Viswanathan, and B.R. Badrinath, “Data on Air Organization and Access,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 9, 1997.

[18] T. Imielinski S. Viswanathan B. R, Badrinath, “Energy Efficient Indexing on Air,” *SiGMOD ACM*, 1994.

[19] Wai Gen Yee, Student Member, IEEE, Shamkant B. Navathe, Edward Omiecinski, and Christopher Jermaine, “Efficient Data Allocation over Multiple Channels at

Broadcast Servers," *IEEE Transactions on Computers*, vol 51, 2002.

[20] Wen-Chih Peng and Ming-Syan Chen, "Dynamic Generation of Data Boradcasting Programs for a Brodcast Disk Array in a Mobile Computing Enviroment," *Proceedings of ACM International Conference on Information and Knowledge Management*, 2000.

[21] X. Li, S. Paul, and M. Ammar, "Layered Video Multicast with Retransmissions (LVMR): Evaluation of Hierarchical Rate Control," *Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, 1998.

[22] Xue Li, S. Paul, P. Pancha, M. Ammar, "Layered Video Multicast with Retransmission (LVMR): Evaluation of Error Recovery Schemes," *Network and Operating System Support for Digital Audio and Video*, 1997.

[23] Yon Dohn Chung and Myoung Ho Kim, "An Index Replication Scheme for Wireless Data Broadcasting," *Journal of Systems and Software*, vol. 51, no. 3, 2000.

[24] Yon Dohn Chung and Myoung Ho Kim, "Effective Data Placement for Wireless Broadcast", *Distributed and Parallel Databases*, 2001.

[25] Y.D. Chung and M.H. Kim, "On Scheduling Wireless Broadcast Data," *Technical Report CS_TR-98-134, KAIST*, 1998.

[26] Yücel Saygin, and Özgür Ulusoy, "Exploiting Data Mining Techniques for Broadcasting Data in Mobile Computing Environments," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 6, 2002.