南　華　大　學

資 訊 管 理 學 系

碩 士 論 文

多頻無線行動環境下廣播資料配對方法之研究

An efficient method to data matching in multiple-channel mobile environment

研 究 生：鄭翔璞

指導教授：吳光閔　博士

中華民國　九十五　年　六　月

南　華　大　學

資 訊 管 理 學 研 究 所

碩 士 學 位 論 文

An efficient method to data matching in multiple-channel mobile environment

研究生：

經考試合格特此證明

口試委員：

吳光閔

指導教授：吳光閔

系主任(所長)：資訊管理學系 系主任吳光閔

口試日期：中華民國　95　年　6　月　29　日

南華大學資訊管理學系碩士論文著作財產權同意書

立書人：鄭翔璞之碩士畢業論文

中文題目：多頻無線行動環境下廣播資料配對方法之研究

英文題目：An effecient method to data matching in multiple-channel moblie environment

指導教授：吳光閔 博士

學生與指導老師就本篇論文內容及資料其著作財產權歸屬如下：

☑ 共同享有著作權

☐ 共同享有著作權，學生願「拋棄」著作財產權

☐ 學生獨自享有著作財產權

學　生：鄭翔璞（請親自簽名）

指導老師：吳光閔（請親自簽名）

中　華　民　國　95　年　6　月　29　月

Acknowledge

首先要感謝指導老師　吳光閔博士的教導，由於老師的用心與引導，讓這份論文得以完成。感謝口試委員王昌斌教授與陳信良教授，於口試中給予許多寶貴的意見。感謝鍾國貴教授、周志賢教授、邱英華教授、李翔詣教授與陳仁義教授等授業老師，於課堂上諄諄教誨。感謝父母親的支持與鼓勵。感謝老婆淑怡的照顧，讓我能無後顧之憂。感謝所有關心我的朋友，此刻心情願與大家分享。

鄭翔璞　僅識

于　南華大學

民國九十五年六月

多頻無線行動環境下廣播資料配對方法之研究

研究生：鄭翔璞　　　　　　　　　　　　　　　　指導教授：吳光閔　博士

南華大學資訊管理學系碩士班

摘　　要

在無線行動環境之下，廣播資料如果是透過多個頻道的方式散播，則有機會發生資料碰撞的情形。所謂資料碰撞，就一個廣播排程而言，是指當同一個用戶端所要求的資料，分別出現在不同頻道的同一個時間格而言。由於用戶端受限於單一的無線網路介面，因此在一個時間格裡，用戶端只能選擇一個頻道去聆聽資料，故而用戶端擇一聆聽之後，其他在同一個時間格的用戶端資料，用戶端就必須等待下一個廣播循環才能聆聽，如此將大幅增長用戶端對資料的存取時間，進而降低無線廣播的效率。因此，如何避免或減少在多頻無線廣播下的資料碰撞率，是一個值得研究的課題。本篇論文提出一個方法，將資料碰撞的問題轉換成資料配對(Bipartite matching)的問題，再利用 Ford-Fulkerson 的演算法，有效的降低資料碰撞的機率。

關鍵字：資料碰撞、資料配對、廣播排程、無線廣播、多頻道

An efficient method to data matching in multiple-channel mobile environment

Student：Charles Cheng Advisors：Dr. Guang-Ming Wu

Department of Information Management
The M.B.A. Program
Nan-Hua University

ABSTRACT

In this paper, we propose a data matching method to reduce the data collision in a multiple-channel wireless broadcasting system. Data collision occurs when two sets of data within a same query were transmitted to a mobile client on two different channels at the same time slot. Since the mobile client can only access one channel at the same time, data collision forces the mobile client to wait for another broadcast cycle. This leads to an increase of the system delay and slows down the system efficiency. Hence, reducing the data collision in data scheduling can improve the utilization of a wireless broadcasting system. The idea of reducing data collision is to find an efficient method to match data on different channels and re-schedule their position on the broadcast schedule. The proposed method is based on the technique of solving the maximum bipartite matching problem. The maximum bipartite matching problem was derived from the problem of finding the maximum flow in a flow network. The Ford-Fulkerson method, which is a solution to solve the maximum flow problem, was introduced to the proposed method. We have shown that the proposed method can reduce the data collision in two-channel broadcast scheduling as well as in multiple-channel broadcast scheduling.

Keyword: data collision, data matching, broadcast scheduling, wireless broadcasting, multiple channels

Contents

Figure

Table

Chapter 1 Introduction

1.1 Background

As the communication media move from wire to wireless, small powered mobile devices, such as cellular phones, personal digital assistants and pocket computers, are already changing our modern style of living. We are getting used to talk to friends via cellular phones rather then home phones and pay phones. Personal digital assistants that support global positioning system can navigate the directions to find the roads to the destination when we are traveling. Moreover, portable computers equipped with wireless network device have become a standard requirement in the computer industry, so that we can easily access Internet indoor and outdoor through laptop computers. All these newly developed phenomena are emerged from the mobile wireless technologies. Mobility has become one of the major design concerns of today's consumer electronics.

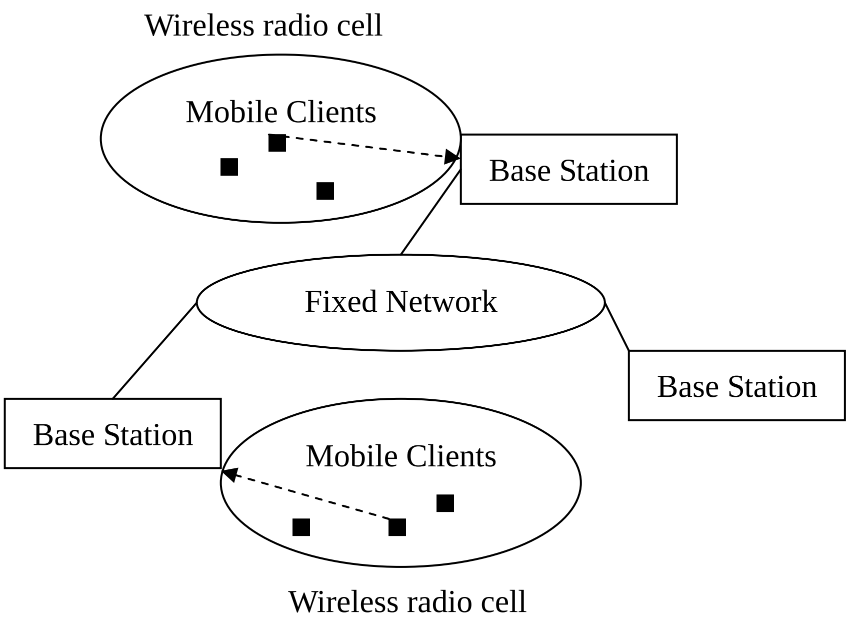


Figure 1-1, the relationship between mobile clients and fixed hosts

Mobility relies on a well-constructed wireless communication network, which consists of two sets of entities: mobile clients and base station. Figure 1-1 demonstrates the relationship between these two sets of entities. The base stations are equipped with wireless interfaces, in which they have the capability of connecting the mobile clients and the fixed network together. The base station is also referred to *access point*. Mobile clients communicate with the fixed network via the base stations.

One of the major characteristics of wireless communication network is the asymmetry in the communication [6]. The *downstream* communication is referred to the direction of data sending, which data is sent from the base station to the mobile client. In contrast, the *upstream* is referred to the data sending from the mobile client to the base station. Usually, the bandwidths of downstream and upstream are different. The downstream has relatively much larger bandwidth than the upstream. This is because most of the mobile client does not have enough power to support large data transmission, due to its mobility and portability concerns. On the other hand, base stations are fixed and capable of transmitting large amount of data. Having the characteristic of asymmetry in the communication, *data broadcasting* was introduced to the data dissemination of the wireless communication network.

Before a broadcast can be performed, base station needs to construct a broadcast program, where the order of the broadcast items was determined. The process of determining the order of the broadcast items is called *broadcast scheduling*. There are two factors need to be considered in the broadcast scheduling process: *access time* and *tuning time* [10]. The access time is the amount of time elapsed from the moment a mobile client requests for data to the receipt of the data requested on the broadcast channels. This factor is used to evaluate the performance of the broadcast scheduling process. The tuning time is the amount of time spent by a client listening to the channels. It is used to measure the power consumed by the mobile client while retrieving required data. Figure 1-2 shows the relationship of the broadcast factors.

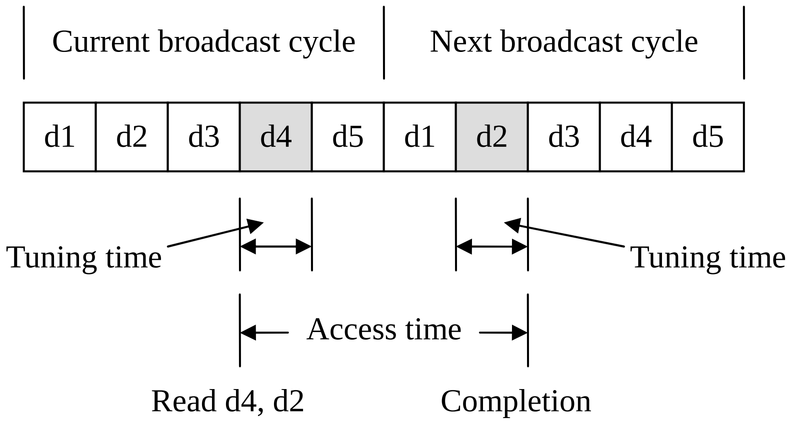


Figure 1-2, the relationship of the broadcast factors, access time and tuning time

The reduction of access time and tuning time can speed up the data receiving process and saves energy. The objectives of the broadcast scheduling process are to minimize both access time and tuning time. To reduce access time, there have been several techniques proposed such as caching and non-uniform broadcasting [1][2][20]. Other techniques including indexing and hashing have also been made to reduce tuning time [8][11][19].

1.2 Research Objectives

The IEEE 802.11 standard supports several independent and equal-capacity communication channels [15]. For example, IEEE 802.11a offers 12 independent channels and IEEE 802.11b/g offers 3 independent channels. These independent channels can be shared simultaneously and access by mobile clients. That means we can use more than one channel to transmit data in the wireless communication network. The base stations and the mobile clients can either equip with multiple 802.11 interfaces for each channel to gain more transmission efficiency, or equip only one 802.11 interface and switch from one channel to another to access multiple channels.

In the multiple-channel broadcast environment, it is possible that the requested data items for a specific mobile client are assigned to the same time slot on different channels. In this case, if the mobile client has equipped with multiple interfaces for each channels, then the mobile client can access different channels at the same time to collect data items. However, if the mobile client has equipped only one interface, it can only collect data items on one channel at a time and have to wait for the next broadcast cycle in order to collect other data items that located on a different channel. When this kind of situation occurred, we said that there was a *data collision* in the multiple-channel broadcast schedule. Data collision has a significant impact on broadcast scheduling. It increased the access time by forcing the mobile client to wait for the next broadcast cycle. In this paper we proposed an effective method that significantly reduces the data collision rate in the broadcast scheduling and increases the efficiency of the wireless network.

Chapter 2　Related Work

2.1　Network connection structures

There are two kinds of network connection structures in the mobile wireless environment: ad-hoc and broadcast. An ad-hoc network is a self-configuring network of mobile hosts connected by wireless links. The mobile hosts are free to move randomly and free to communicate to other mobile hosts nearby. The topology of the ad-hoc network may change rapidly and unpredictably. Figure 2-1 shows an ad-hoc network constructed with three mobile clients. Instead of a central data server in traditional client/server network, each mobile host in the ad-hoc network acts like a data repeater which receives data items from adjacent mobile host and transmits them to another adjacent mobile host, until the data items reach their destination host. The ad-hoc network problem was focused on routing [20] and power balancing [26].

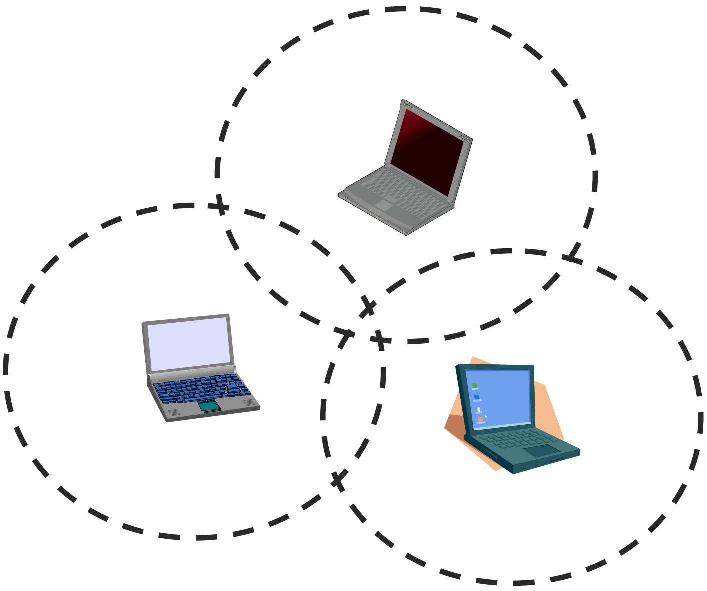


Figure 2-1, an ad-hoc network constructed with three mobile clients

A broadcasting network is a network that uses a central data server, called base station, to disseminate data items to mobile clients. Broadcasting network allows the base station to act like a data distribution server and sends data to the broadcast channels. Mobile clients listen to the broadcast channels and grab only the data items of their interest. Figure 2-2 shows the architecture of a wireless broadcasting system. In this paper we focus on the method of broadcasting network.

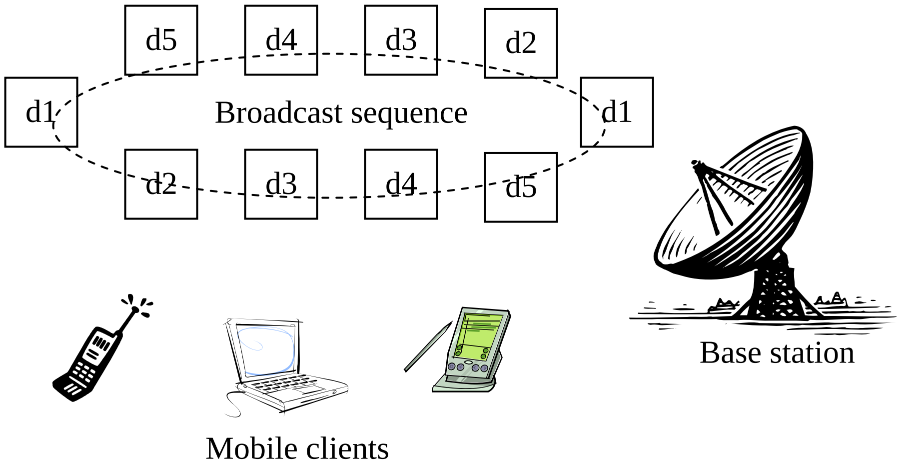


Figure 2-2, the architecture of a typical wireless broadcasting system

2.2 Broadcast methods

In general, there are three kinds of broadcast methods that we typically used in practice: push-based broadcasting [1][29][11][7], pull-based broadcasting [4][5][11] and hybrid broadcasting [28] [14][20].

1. *Push-based* (pure broadcast): In the push-based broadcasting network, the base station sends data to mobile clients without waiting for any request from mobile clients. The data items sent by the base station were collected according to a profile that provided by mobile client after subscribed to the network service. The broadcast scheduling program generates a sequential order list of data items that need to be sent on air. The base station broadcasts the data items according to the scheduled list periodically and continuously. Mobile clients listen to the broadcast channels and grab only the data items of their interest. As the quantity

of data items increased, the mean access time was increased respectively. Hence, the mobile clients spend more time listen to the broadcast channels, which significantly brings down the efficiency of the network. Moreover, it is possible for a mobile client missing a data item during the broadcast period. In this case, the mobile client has to wait for the next broadcast cycle to recover its missing item.

2. *Pull-based* (on-demand): In the pull-based broadcasting network, the mobile clients send their requests to the base station. The base station responds to each mobile client's request and broadcasts only the data items that mobile clients interested. One of the characteristic in wireless network is the asymmetry in communication, which means the upstream (client to server) communication is relatively smaller than the downstream (server to client) communication. Due to the asymmetric communication, there exists access latency while a large number of mobile clients try to send their request to the base station at the same time.
3. *Hybrid*: hybrid broadcasting network is an integration of the push-based broadcasting and the pull-based broadcasting. Hybrid broadcasting network uses push-based operation for data that requested most (hot data) and pull-based operation for data less popular (cold data). The mobile clients not only listen to the broadcast channels but also send request to the base station. The combination of push- and pull-based broadcasting makes the broadcast network more reliable and efficient.

2.3 Broadcast Channels

The wireless environment can support single broadcast channel or multiple broadcast channels.

1. *Single-channel*: when the members of a broadcast system used only one broadcast channel to communicate, the broadcast system is a single-channel broadcast system. Researches of single-channel broadcasting were focusing on finding the optimal method to construct a broadcast schedule using a minimum of access time [8][15][16].

2. *Multiple-channel*: multiple-channel broadcast systems use either multiple wireless interfaces for different communication channels or a single wireless interface for all channels. Equipping with multiple interfaces increases the cost of the application, in exchange for a higher performance. In the real world, most of our applications were still using single wireless interface instead of multiple interfaces. Cost is the primary concern. The trade-off is lowering the performance.

Chapter 3 Maximum bipartite matching for the data collision problem

3.1 Broadcast scheduling problem

The objective of the broadcast scheduling problem is to minimize the access time and the tuning time. In multiple-channel broadcast scheduling, we have described a data collision problem which caused by a misallocation of data items on the broadcast sequences. The effect of the data collision increases the access time by forcing the mobile client to wait for another broadcast cycle to acquire the data items of its interest. In this paper, we assume that mobile client has equipped only one wireless interface and the lengths of each data item on the broadcast sequence are equal.

In Figure 3-1a, we assume that the base station has two channel broadcast sequences <d_0, d_1, d_2, d_3, d_4> and <d_5, d_6, d_7, d_8, d_9>. The first channel has five data items denoted by d_0 to d_4. The second channel also has five data items denoted by d_5 to d_9. Suppose a query from a mobile client has requested for data item d_2, d_3, and d_7. We found that d_2 and d_7 are located on the same time slot of each channel. Here, we have a data collision on d_2 and d_7. The mobile client needs to access both channels at the same time in order to collect d_2 and d_7. Under the constraint of one wireless interface each mobile client, it is not possible for the mobile client to access both channels at the same time slot. The mobile client can only choose one data item to access. The mobile client has to access one channel to pick up a data item, and then wait for the next broadcast cycle to access another channel and pick up the other data item. In this example, the access time has almost been doubled. It is clearly that the situation here is what we do not want, because the increase in access time really brings down the efficiency of the broadcast system.

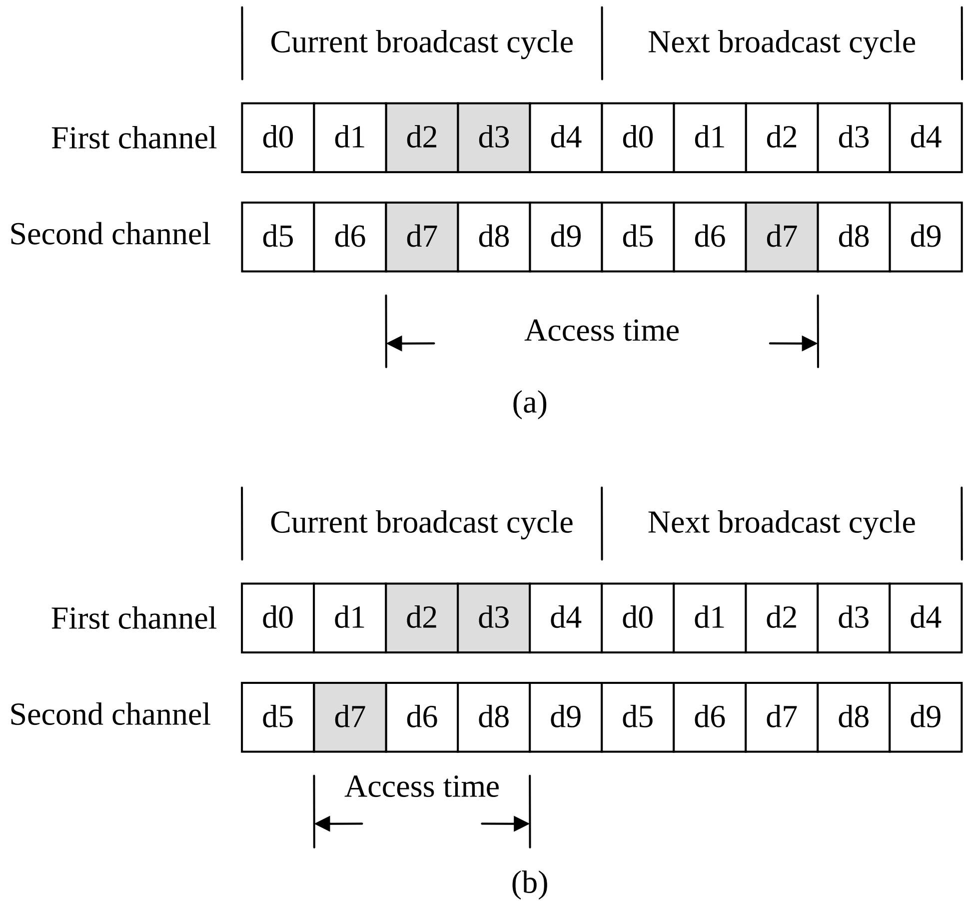


Figure 3-1, (a) the example of the data collision in multiple-channel broadcast schedule, (b) the result of reallocating d_7

To solve this problem, we have to reallocate d_2 or d_7. Figure 3-1b shows the result of the reallocation of d_7. After the reallocation of d_7, the mobile client can collect d_2, d_3, and d_7 during the same broadcast cycle. We will discuss the strategy for the reallocation of data items in the next section.

3.2 Basic idea

The example showed in Section 0 has given us a picture of how the misallocation of data items on the broadcast sequences could affect the efficiency of the broadcast system. It is possible to avoid the data collision problem by reallocating the placement of some data items.   In this paper, we applied the maximum bipartite matching technique to the data collision problem to reduce the data collision rate.

The maximum bipartite matching technique [11] is a well known method that derived from the Ford-Fulkerson method to find the maximum flow in a flow network.   Given a graph $G = (V, E)$, where *V* is the vertex set of graph *G* and *E* is the edge set of graph *G*, a bipartite graph is a special graph where the set of vertices can be divided into two disjoint sets such that no two vertices of the same set share an edge.   The vertex is matched if it is incident to an edge in the matching, otherwise the vertex is unmatched.   A maximum matching contains the largest possible number of edges.   The maximum bipartite matching technique uses the concept of the Ford-Fulkerson method.   The Ford-Fulkerson method is an algorithm that computes the maximum flow in a flow network.

If we can transform the multiple-channel broadcast scheduling problem to a maximum bipartite matching problem, we could use the Ford-Fulkerson method to find the maximum matching and reallocate the placement of the data items on the broadcast sequence according to the resulting matching.

3.3 Problem transformation

In order to transform the data collision problem to the maximum bipartite matching problem, we consider a two-channel broadcast sequence as a bipartite network.   Each data item in the broadcast sequence becomes a node in the network.   The columns represent the different channels of the broadcast sequence, and the rows represent the time slots where the data items were located.   Then we apply a source node *s* and a sink node *t* to the network in addition.   The source node *s* is connected to each node on the first column. Each node on the first column is connected to the node that located on the same row of the second column.   Each node on the second channel is connected to the sink node *t*. Figure 3-2a shows a graph of a transformed network.

In fact, there are constraints that guiding us from connecting the nodes on one column to the other. The constraints are provided from the request of the mobile clients, called the *queries*. The queries were series of information that sent by the mobile clients to the base station. These queries contained the information of the data items of the mobile clients' interest. Suppose that there exist a query {d_2, d_3, d_7} for the example in Figure 3-2a, because of the reason we mentioned in Section 0, we discovered a data collision occurred on d_2 and d_7. A connection between the first column and the second column represents of two nodes on the different column along the connection can be placed on a same row without any risk of data collision. Since there existed a data collision between d_2 and d_7, the connection between d_2 and d_7 is removed to indicate a data collision occurred. Figure 3-2b shows a connection been removed from a flow network.

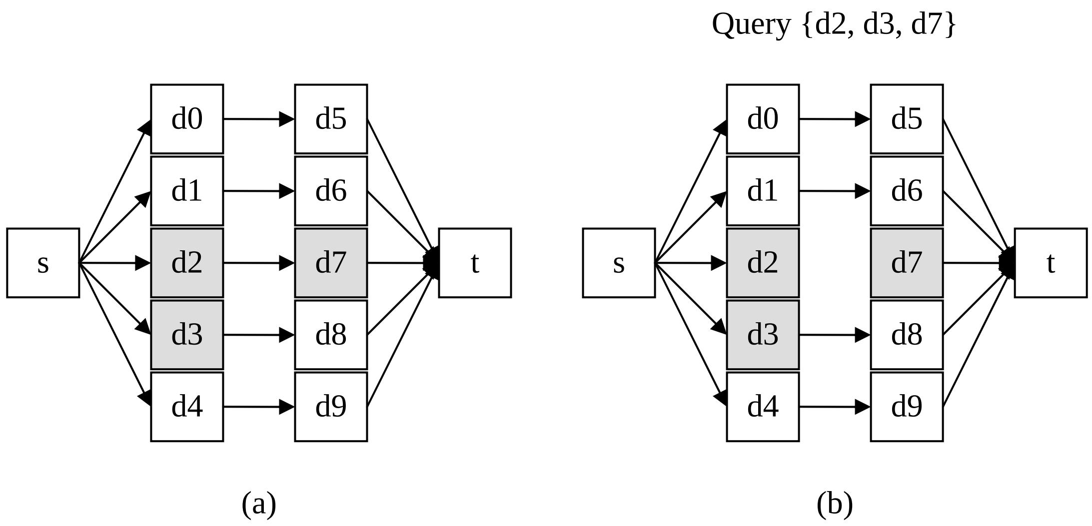
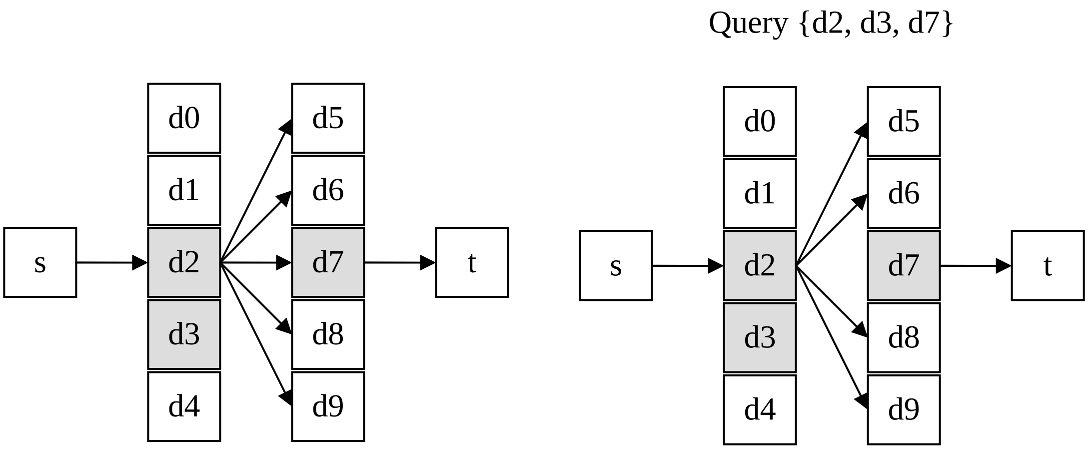


Figure 3-2, (a) a two-channel broadcast sequence transformed into a bipartite flow network, (b) a bipartite flow network with query constraint

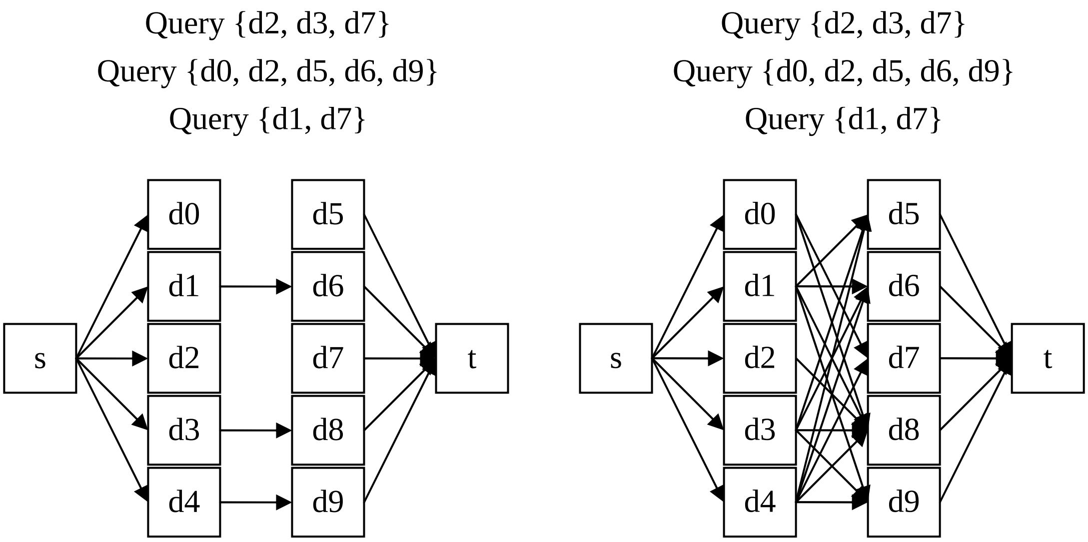
We then consider the connections between the nodes on the first column and the nodes on the second column. In the original graph, each node on the first column has only one connection that connected to the node that located on the same row of the second column, except for the nodes that appeared on same queries. Actually, it is possible for a node on one column to connect to every node of the other column. Figure 3-3a shows all possible connections of d_2. After applying the query constraint, the connection between d_2 and d_7 is removed, as in Figure 3-3b.



(a) (b)

Figure 3-3, (a) a graph of all possible connections of d_2, (b) the removal of the connection of d_2 and d_7

We expand the connection of all nodes in the network shown in Figure 3-3b, and apply two additional queries, {d_0, d_2, d_5, d_6, d_9} and {d_1, d_7}, to the network shown in Figure 3-2b and Figure 3-3b. Figure 3-4a and 3-4b show the result of expansion of all nodes and the application of the original query and two additional queries. We call the graph in Figure 3-4a the *original graph* and the graph in Figure 3-4b the *expanded graph*. In the original graph, the data collision occurs 2 times out of 5. Therefore, the data collision rate of the original graph is 0.4. To calculate the collision rate of the expanded graph, we have to apply the Ford-Fulkerson method to the expanded graph. We will be discussed in the following sections.



(a) (b)

Figure 3-4, (a) a graph of all possible connections of each node in the network, (b) the constraint queries applied to the graph

3.4 Ford-Fulkerson method

The Ford-Fulkerson method states as the following [11]:

Given a graph $G = (V, E)$, with residual capacity $c(u, v)$ and flow $f(u, v) = 0$ for the edge from u to v, we want to find the maximum flow from the source s to the sink t. In each iteration, the flow between u and v does not exceed the capacity (3-1) and the net flow is maintained (3-2).

$$f(u,v) \le c(u,v) \tag{3-1}$$

$$f(u,v) = -f(v,u) \tag{3-2}$$

Also, the amount of flow into a node equals the flow out of the node (3-3).

$$\sum_{v} f(u,v) = 0 \tag{3-3}$$

This means that the flow through the network is a legal flow after each round in the algorithm. We define the residual network $G_f = (V, E_f)$ to be the network with capacity $c_f(u,v) = c(u,v) - f(u,v)$ and no flow.

We illustrate the algorithm of the Ford-Fulkerson method as the following:

Inputs: Graph G with flow capacity c, a source node s, and a sink node t
Output: Flow f such that f is maximal from s to t

1. **for** each edges $(u,v) \in E[G]$
2. **do** $f[u,v] \leftarrow 0$
3. $f[v,u] \leftarrow 0$
4. **while** there exists a path p from s to t in the residual network G_f
5. **do** $c_f(p) \leftarrow \min\{c_f(u,v) : (u,v)$ is in $p\}$
6. **for** each edge in p
7. **do** $f[u,v] \leftarrow f[u,v] + c_f(p)$
8. $f[v,u] \leftarrow -f[u,v]$

3.5 Maximum bipartite matching

To find the maximum bipartite matching of the expanded graph, we applied the Ford-Fulkerson method to the graph. Each edge in the graph has a capacity of 1. The result was shown in Figure 3-5.

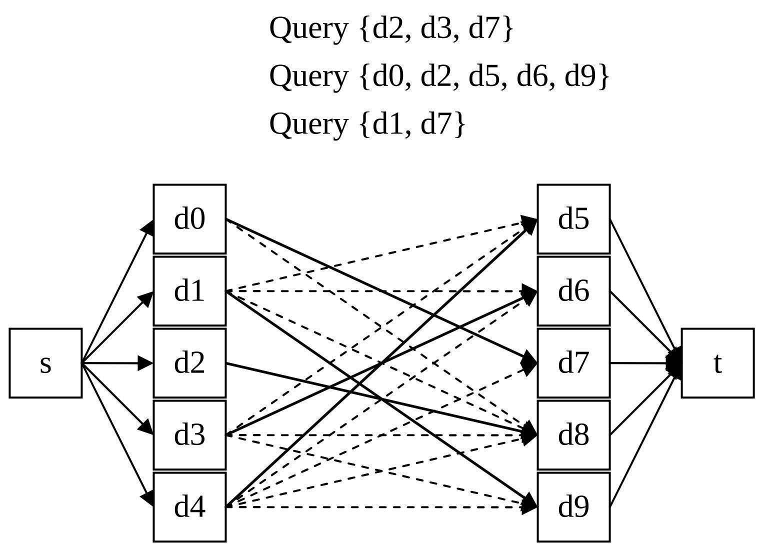


Figure 3-5, the result of applying Ford-Fulkerson method of the expanded graph

After applying the Ford-Fulkerson method, the solid edges have a flow of 1 and all other edges have no flow. The data collision rate of the original graph is 0 because there is no data collision in the network.

By the removal the source *s* and the sink *t*, our interest lies on the part of those solid edges. The solid edges correspond to the maximum matching of the bipartite graph and show a pair of vertices or nodes that should be aligned together on the same row of the broadcast sequence, e.g. the same time slot. We reallocate the broadcast sequence by fixing on one channel and switching the data items on the other channel according to the solid edges of the Ford-Fulkerson method. Figure 3-6 shows result of switching the data items on the broadcast sequence.

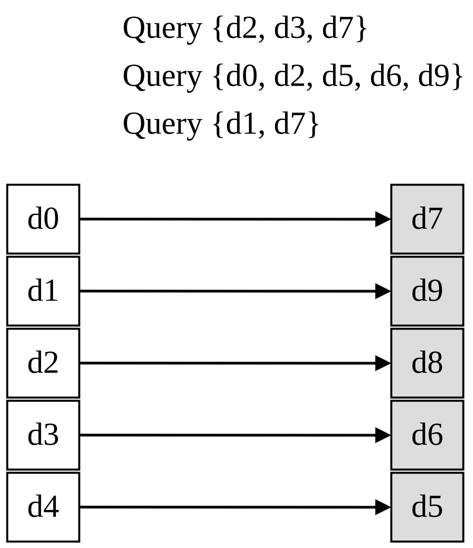


Figure 3-6, the result of switching the data items on the broadcast sequence

Chapter 4 Performance Evaluation

4.1 Simulation model.

We have composed a computer program to evaluate the performance of the proposed solution. In this simulation program, we focused on the condition of two broadcasting channels. The first input of the simulation program is a set of data items divided into two channels in uniform distribution. This dataset represents the original broadcast sequence in the broadcast system. The second input is a set of queries that provides the placement constraint, which represents the queries issued out from the mobile clients. We assumed that a mobile client can issue only one query.

We have introduced a set of parameters to help the evaluation of the simulation performance:

Parameter	Definition	Range
N	Total number of data items in the broadcast sequence	50 ~ 1000
Q	Total number of queries issued from the mobile clients	50 ~ 1000
R	Ratio of the maximum number of items in a query to N	0.05 ~ 0.4

4.2 Simulation analysis

In the simulation program, we compared the data collision rate of the original broadcast sequence and the sequence that derived from the maximum bipartite matching method. We have come to several results by using a set of various values of parameters. We will discuss the effect of the parameters below. The *Original collision rate* is the number of collision pairs divided by the number of total pairs from the original broadcast sequence.

The *Bipartite collision rate* is the number of collision pairs divided by the number of total pairs from the sequence of our result. The *Improve rate* is calculated as follow: (Original collision rate - Bipartite collision rate) / Original collision rate. When the improve rate reaches 100%, it means that our result sequence has eliminated all collision pairs from the original broadcast sequence.

4.2.1 Number of data items (N)

To find the effect of the number of data items, we let the parameter Q = 300 and R = 0.3. We obtained a set of results by applying various numbers of data items. In Figure 4-1, we come to a result that for each number of data items, the data collision rate of the bipartite matching sequence is lesser than the original sequence. As shown in Table 1, the data collision rates on both original and bipartite sequence are reduced as the number of data items increased. There is a significant increase of the improvement rate when the number of data items increased. The value 0 on data collision rate shows that there is no data collision in the broadcast sequence. This result indicates that when we increase the number of data items, the probability of finding a corresponding data match in the broadcast sequence is increased.

Table 1, the effect of the number of data items

N	Q	R	Original collision rate	Bipartite collision rate	Improve (%)
50	300	0.3	0.96	0.76	20.83%
100	300	0.3	0.92	0.7	23.91%
200	300	0.3	0.92	0.45	51.09%
300	300	0.3	0.933	0.407	56.38%
500	300	0.3	0.832	0.232	72.12%
1000	300	0.3	0.548	0	100.00%

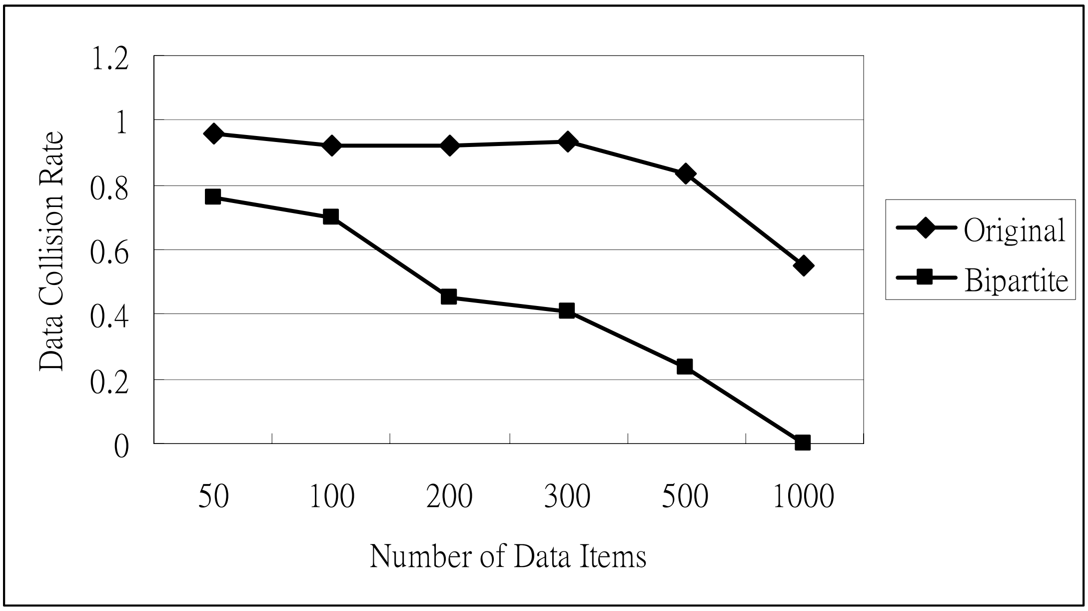


Figure 4-1, the comparison of the data collision rate of the original and the bipartite with various numbers of data items

4.2.2 Number of queries (Q)

To find the effect of the number of queries, we let the parameter N = 300 and R = 0.3. We obtained a set of results by applying various numbers of queries. In Figure 4-2, we come to a result that for each number of queries, the data collision rate of the bipartite matching sequence is lesser than the original sequence. As shown in Table 2, the data collision rates on both original and bipartite sequence are increased as the number of queries increased. There is a significant decrease of the improvement rate when the number of queries increased. This result indicates that when we increase the number of queries, the probability of finding a corresponding data match in the broadcast sequence is decreased.

Table 2, the effect of the number of queries

N	Q	R	Original collision rate	Bipartite collision rate	Improve (%)
300	50	0.3	0.167	0	100.00%
300	100	0.3	0.467	0	100.00%
300	200	0.3	0.747	0.147	80.32%
300	300	0.3	0.933	0.407	56.38%
300	500	0.3	0.927	0.653	29.56%
300	1000	0.3	0.99	0.893	9.80%

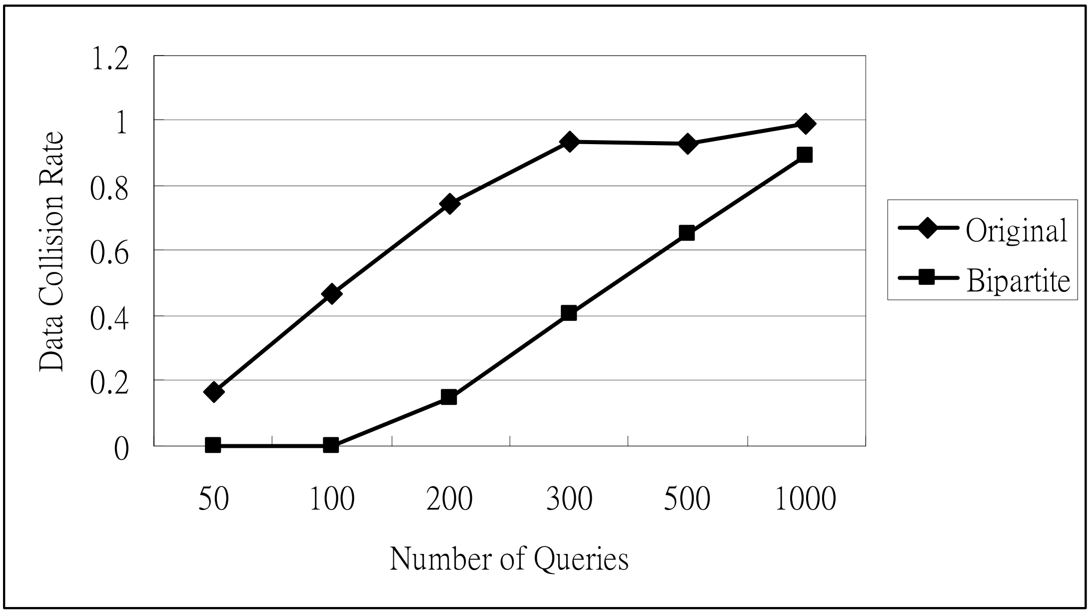


Figure 4-2, the comparison of the data collision rate of the original and the bipartite with various numbers of queries

4.2.3 Ratio of items in a query to data items (R)

To find the effect of the number of queries, we let the parameter N = 500 and Q = 1000. We obtained a set of results by applying various numbers of ratios. In Figure 4-3, we come to a result that for each number of ratios, the data collision rate of the bipartite matching sequence is lesser than the original sequence. As shown in Table 3, there is a significant decrease of the improvement rate when the number of ratios increased. The data collision rates on both original and bipartite sequence are increased as the number of ratios increased. This result indicates that when we increase the number of ratios, the probability of finding a corresponding data match in the broadcast sequence is decreased.

Table 3, the effect of the number of ratio parameter

N	Q	R	Original collision rate	Bipartite collision rate	Improve (%)
500	1000	0.05	0.464	0	100.00%
500	1000	0.1	0.788	0.07	91.12%
500	1000	0.15	0.88	0.352	60.00%
500	1000	0.2	0.96	0.608	36.67%
500	1000	0.25	0.977	0.736	24.67%
500	1000	0.3	0.984	0.816	17.07%
500	1000	0.35	0.98	0.864	11.84%
500	1000	0.4	0.992	0.92	7.26%

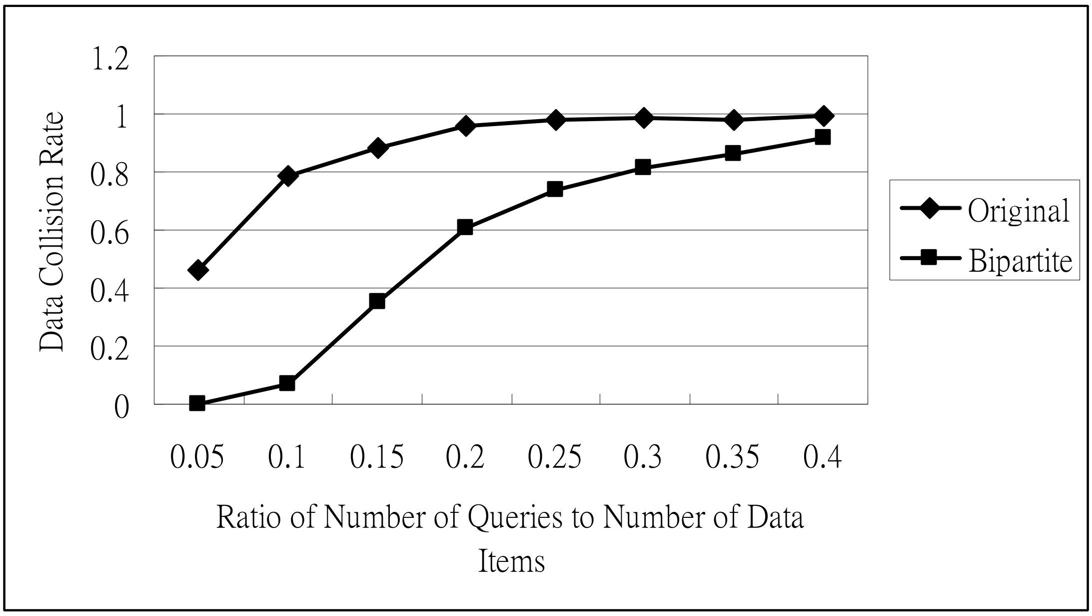


Figure 4-3, the comparison of the data collision rate of the original and the bipartite with various numbers of ratios

Chapter 5 Conclusion

In this paper, we have proposed a method that based on the technique of maximum bipartite matching to solve the multiple-channel broadcast scheduling problem. From the simulation analysis, we found that our method has a significant impact on the reduction of the data collision. For every set of parameters been used in our simulation, the data collision rate was reduced tremendously. For the future work, we would have to consider the possibility of applying our method to the expansion of the number of broadcast channels to 2^n.

References

[1] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, “Broadcast disks: Data management for asymmetric communication environments,” in Proceedings of ACM SIGMOD Conerence, pp. 199-210, 1995

[2] S. Acharya, M. Franklin, and S. Zdonik, “Disseminating updates on broadcast disks,” in Proceedings of Very Large Data Bases Conference, pp. 354-365, 1996

[3] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, “Balancing Push and Pull for Data Broadcast,” Proc. ACM SIGMOD Int’l Conf. Management of Data, Phoenix, Ariz., pp. 183-194, May, 1997

[4] S. Acharya and S. Muthukrishnan, “Scheduling On-Demand Broadcasts: New Metrics and Algorithms,” MOBICOM, pp. 43-54, 1998

[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] D. Barbará, “Mobile Computing and Databases – A Survey,” IEEE Transactions on Knowledge and Data Engineering, Vol.11, No.1, Jan/Feb 1999

[7] T. F. Bowen, G. Gopal, G. Herman, T. Hickey, K. C. Lee, W.H. Mansfield, J, Raitz, and A. Weinrib, “The DATACYCLE architecture,” Comm. ACM, vol. 35, no. 12, pp. 71-81, Dec. 1992

[8] Y. I Chang and W. H Hsieh, “An Efficient Scheduling Method for Query-Set-Based Broadcasting in mobile Environments,” Proc. 24th International Distributed Computing Systems Workshops, pp. 478-483, 2004

[9] Y. D. Chung and M. H. Kim, “An index replication scheme for wireless data broadcasting,” Journal of Systems and Software, vol. 51, no. 3, pp. 191-199, 2000

[10] Y. D. Chung and M. H. Kim, “Effective Data Placement for Wireless Broadcast,” Distributed and Parallel Databases, 9, pp. 133-150, 2001

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, “Introduction to Algorithms, Second Edition, The MIT Press, pp. 664-669

[12] Q. Fang, S. V. Vrbsky, Y. Dang, and W. Ni, “A Pull-Based Broadcast Algorithm that Considers Timing Constraints,” Proc. International Conference Parallel Processing ICPP 2004 Workshops, pp. 46-53, 2004

[13] D. K. Gifford, “Polychannel Systems for Mass Digital Communication,” Comm.

ACM, vol. 33, no. 2, Feb. 1990

[14] C. L. Hu and M. S. chen, “Adaptive Balanced Hybrid Data Delivery for Multi-Channel Data Broadcast,” Proc. IEEE Int’l Conf. Communications, pp. 960-964, 2002

[15] Q. L. Hu, D. L. Lee, and W. C. Lee, “Dynamic Data Delivery in Wireless Communication Environment,” Proc. Workshop on Mobile Data Access, pp. 351-358, 1999

[16] J. J. Hung and Y. Leu, “Efficient Index Caching Schemes for Data Broadcasting in Mobile Computing Environments,” IEEE DEXA, pp. 1-7, 2003

[17] IEEE 802.11 Working Group, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ANSI/IEEE std. 802.11, Sept. 1999

[18] 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

[19] T. Imielinski, S. Viswanathan, and B. R. Badrinath, “Energy efficient indexing on air,” in Proceeding of ACM SIGMOD Conference, pp. 25-36, 1994

[20] S. J. Kim, W. J. Kim, and Y. J. Suh, “Efficient Broadcast Schemes with Transmission Power Control in Mobile Ad Hoc Networks,” IEEE International Conference on Communication, vol. 7, pp. 3859-3863, 2004

[21] P. Kyasanur and N. H. Vaidya, “Routing in multi-channel multi-interface ad hoc wireless networks,” Technical report, University of Illinois at Urbana-Champaign, December 2004

[22] B. Lee and S. Jung, “An Efficient Tree-Structure Index Allocation Method over Multiple Broadcast Channels in Mobile Environments,” Database and Expert Systems Applications, pp. 433-443, 2003

[23] G. Lee, S. C. Lo, and A. L. P. Chen, "Data Allocation on the Wireless Broadcast Channel for Efficient Query Processing," IEEE Trans. On Computers, vol. 51, no. 10, pp. 1237-1252, October 2002

[24] S. C. Lo and A. L. P. Chen, “Optimal Index and Data Allocation in Multiple Broadcast Channels,” Proc. International Conference on Data Engineering, pp. 293-302, 2000

[25] K. Stathatos, N. Poussopoulos, and J. S. Baras, “Adaptive Data Broadcast in Hybrid Networks,” Proc. International Conference on Very Large Data Bases, pp. 326-335,

1997

[26] C. Tang, C. S. Raghavendra, and V. Prasanna, “Energy Efficient Adaptation of Multicast Protocols in Power Controlled Wireless Ad Hoc Networks,” Proc. Parallel Architectures, Algorithms and Networks I-SPAN ’02, International Symposium, pp. 80-85, 2002

[27] C. Su, L. Tassiulas, and V. J. Tsotras, “Broadcast scheduling for information distribution,” Wireless Networks, vol. 5, no. 2, pp. 137-147, 1998

[28] Nitin H. Vaidya and Sohail Hameed, “Scheduling data broadcast in asymmetric communication environments,” Wireless Networks 5 (1999), pp. 171-182

[29] J. Wong, “Broadcast Delivery,” Proceeding of IEEE, vol. 76, no. 12, Dec. 1988