Proc. of the 10th ACM International Conference on Information and Knowledge Management (CIKM), Nov. 2001.

Binary Interpolation Search for Solution Mapping on Broadcast and On-demand Channels in a Mobile Computing Environment

Jiun-Long Huang, Wen-Chih Peng and Ming-Syan Chen
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan, ROC
E-mail: mschen@cc.ee.ntu.edu.tw, {jlhuang, wcpeng}@arbor.ee.ntu.edu.tw

ABSTRACT

We explore in this paper the problem of dynamic data and channel allocations with the number of communication channels and the number of data items given. It is noted that the combined use of broadcast and on-demand channels can utilize the bandwidth effectively for data dissemination in a mobile computing environment. We first derive the analytical models of the expected delays when the data are requested through the broadcast and on-demand channels. Then, we transform this problem into to a guided search problem. In light of the theoretical properties derived, we devise an algorithm based on binary interpolation search, referred to as algorithm BIS, to obtain solutions of high quality efficiently. In essence, algorithm BIS is guided to explore the solution space with higher likelihood to be the optimal first, thereby leading to an efficient and effective search. It is shown by our simulation results that the solution obtained by algorithm BIS is of very high quality and is in fact very close to the optimal one. Sensitivity analysis on several parameters, including the number of data items and the number of communication channels, is conducted.

Keywords

Data dissemination, dynamic data and channel allocation, mobile computing, data broadcasting

1. INTRODUCTION

In a mobile computing environment, a mobile user with a power-limited mobile computer can access various information via wireless communication. Applications such as stock activities, traffic reports and weather forecast have become increasingly popular in recent years [24][25]. It is noted that mobile computers use small batteries for their operations without directly connecting to any power source, and the bandwidth of wireless communication is in general limited. As a result, an important design issue in a mobile system is to conserve the energy and communication bandwidth of a mobile unit while allowing mobile users of the ability to access information from anywhere at anytime [4][6][13][19].

In order to conserve the energy and communication bandwidth of a mobile computing system, a data delivery architecture in which a server continuously and repeatedly broadcasts data to a client community through a *single* broadcast channel was proposed in [1]. In a push-based information system, a server generates a broadcast program to broadcast data to mobile users. This broadcast channel is also referred to as a *broadcast disk* from which mobile users can retrieve data [1]. The mobile users need to wait for the data of interest to appear on the broadcast channel, and the corresponding waiting time is called the expected delay of that data item. One objective of designing proper data allocation in the broadcast disks is to reduce the average expected delay of data items. The research issues have attracted a considerable amount of attention, including on-demand broadcast [2][3], data indexing [10][12] and client cache management [23][26][27]. In addition, a significant amount of research effort has been elaborated on developing the index mechanisms [17][21] and data allocation schemes [18][20] in *multiple* broadcast channels.

In addition to broadcast mode, channels can operate in an on-demand mode in which a client explicitly sends data requests to retrieve the data items of interest. The major advantage of data broadcast is its scalability since the performance of the system does not depend on the number of clients listening to the broadcast channels. However, the performance degrades as the number of data items being broadcast increases. It has been shown that the combined use of the broadcast and on-demand channels can utilize bandwidth more efficiently for data dissemination [14]. Hence, the problem of channel allocation is to partition a given total number of communication channels into broadcasting ones and on-demand ones.

In this paper, we study the problem of dynamic data and channel allocation. Consider the illustrative example shown in Figure 1. Assume that the data items R_i, $1 \leq i \leq 15$, are of the same size and are sorted by their access frequencies. The number of channels in this example is assumed to be four. In the beginning, two channels are assigned as broadcast channels and the other two are on-demand ones. Five

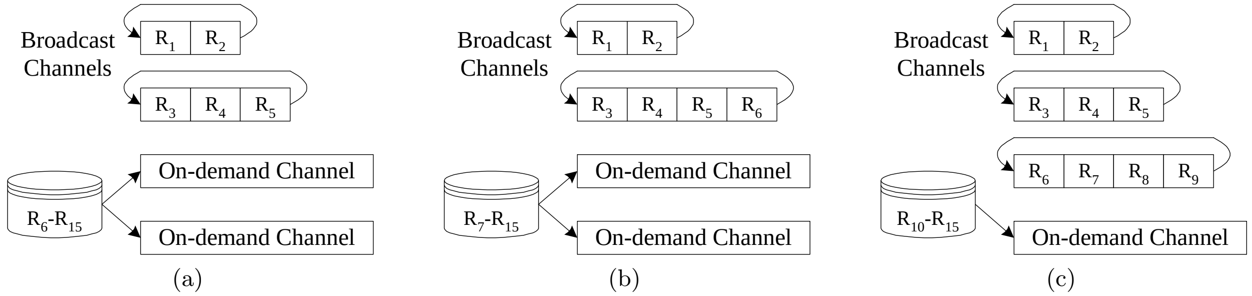


Figure 1: An example scenario of dynamic data and channel allocation

data items are put in broadcast channels and the broadcast program is shown in Figure 1a. When the data request rate increases, R_6 is moved from the on-demand channel to the broadcast channel.[1] This will reduce the data request rate to on-demand channels and the expected waiting time in on-demand channels is hence reduced. The broadcast program is then rescheduled and the new broadcast program is shown in Figure 1b. If the data request rate keeps increasing, as shown in Figure 1c, one channel is re-assigned to be a broadcast one and three data items (R_7, R_8 and R_9) are moved from on-demand channels to broadcast channels. As the partition of broadcast and on-demand channels varies, the number of data items in those channels changes accordingly, showing the dynamic characteristics of this data and channel allocation problem.

Explicitly, we explore in this paper the problem of dynamic data and channel allocations with the number of communication channels and the number of data items given. Gathering the access frequencies of data items is another research issue, since clients do not explicitly send data requests when the data items of interest are put in broadcast channels. Research works [9][28] in gathering or estimating the data access frequencies in broadcast channels can complement our work. We first describe the analytical models of broadcast and on-demand channels. Then, we transform this problem into to a guided search problem. In light of theoretical properties derived, we devise an algorithm based on binary interpolation search, referred to as algorithm BIS, to obtain the solutions of high quality efficiently. In essence, algorithm BIS is guided to explore the solution space with higher likelihood to be the optimal first, thereby leading to an efficient and effective search. It is shown by our simulation results that the solution obtained by algorithm BIS is of very high quality and is in fact very close to the optimal one. Sensitivity analysis on several parameters, including the number of data items and the number of communication channels, is conducted. Moreover, algorithm BIS is of very good scalability which is particularly important for its practical use in a mobile computing environment.

We mention in passing that the authors of [22] provide an adaptive algorithm to allocate data items on broadcast and on-demand channels with a fixed ratio for on-demand and broadcast bandwidth. In [15], the optimal channel allocation is calculated when the access delay of data items is formulated. The work in [16] is designed to keep the load of on-demand channels in a predetermined region and to shuffle the loads among broadcast and on-demand channels when so proper. Both works [15] and [16] employed flat broadcast programs which broadcast data items with the same frequencies. In contrast, algorithm BIS employs a binary interpolation search technique to dynamically partition the channels into broadcast and on-demand ones in accordance with the incoming requests.

The rest of this paper is organized as follows. In Section 2 we briefly describe the problem and analytical models of broadcast and on-demand channels. In Section 3, we transform the data and channel allocation problem into a search problem and develop an efficient algorithm. The performance evaluation of the proposed algorithm is presented in Section 4. This paper concludes in Section 5.

2. PROBLEM DESCRIPTION

To facilitate the presentation of this paper, some preliminaries are given in this section. We first describe the system architecture and the problem of data and channel allocation in Section 2.1. The analytical models of broadcast channels, on-demand channels and the overall expected delay of the system are provided in Section 2.2.

2.1 System Description and Problem Formulation

In our companion work [18], without considering the use of on-demand channels, we explored the problem of generating hierarchical broadcast programs with the number broadcast channels given. Specifically, we transformed the problem of generating hierarchical broadcast programs into the one of constructing a data allocation tree with variant-fanout. The depth of the data allocation tree corresponds to the number of broadcast channels, and those leaf nodes in the same level of the data allocation tree correspond to those data items to be put in the same broadcast channel. The work in [18] devised two algorithms, OPT and VF^K, to generate broadcast program. Algorithm OPT is an A*-like algorithm which is able to generation the optimal broadcast program. VF^K is a greedy, heuristic algorithm can efficiently obtain a hierarchical broadcast program.

Denote the total number of data items as n, and data items as R_i, $1 \leq i \leq n$. Naturally, the n_B frequently accessed data items are placed in broadcast channels and the other $n_O = n - n_B$ data items are in on-demand channels. Let $K = K_B + K_O$ represent the total number of channels where K_B and K_O are the numbers of broadcast and on-

---

[1] The criterion for data movement will be given in Section 3 later.

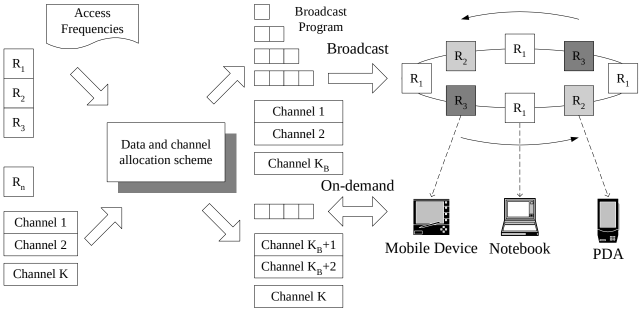


Figure 2: The architecture of a data dissemination system

demand channels, respectively. The size of each data item is assumed to be equal.[2] The problem of generating broadcast programs for K_B broadcast channels can be viewed as a discrete minimization problem: Given a set of n_B data items with their access probabilities, partition them into K_B parts so that the average expected delay of all data items is minimized. Note that once K_B is decided, K_O follows. Such a minimization problem is known to be NP-hard [5].

Figure 2 shows the architecture of a data dissemination system. We assume that each data item is the same size and read-only [16]. Without knowing the placement of the requested data item, a mobile user has to send a data item request via on-demand channels. If the requested data item is placed in an on-demand channel, the server will reply the data item directly. If the data item is in a broadcast channel, the server replies the broadcast information such as the channel frequencies, the data identifiers, the data index information, and other auxiliary information [15]. After receiving the broadcast channel access information, the client will listen to the broadcast channel and wait for the request data item. If a client already has the broadcast information and the requested data item is placed in broadcast channels, this client does not send a data request to servers via on-demand channels unless the broadcast information is expired or invalidated.

With the above model, the problem of data and channel allocation we consider in this paper is formulated as follows:

Problem of data and channel allocation: Given K broadcast channels, n data items and their access frequencies, we shall do the following tasks so that the average expected delay of all data items is minimized.

1. Determine the numbers of broadcast and on-demand channels (i.e., K_B and K_O), where $K = K_B + K_O$.
2. Determine the numbers of data items allocated to broadcast and on-demand channels (i.e., n_B and n_O), where $n = n_B + n_O$.
3. Construct the broadcast program in the K_B broadcast channels with the n_B most frequently accessed data items.

2.2 Analytical Models

Table 1: Description of symbols

Description	Symbol
Number of channels	K
Number of broadcast channels	K_B
Number of on-demand channels	K_O
Number of data items	n
Number of data items in broadcast channels	n_B
Number of data items in on-demand channels	n_O
The jth data item	R_j
The access frequency of data item R_j	$Pr(R_j)$
The size of each data item	s
The size of each data request	r
The channel bandwidth	b
The data request rate	λ
Avg. service time per on-demand channel	$\frac{1}{\mu}$

2.2.1 *Broadcast Channels*

Since there is more than one data broadcast program for given K_B and n_B, we use $W_B(K_B, n_B)$ to represent the *minimal* expected delay when the requested data item is put in broadcast channels. Let $C(K_1, n_1)$ be a configuration where $K_B = K_1$ and $n_B = n_1$. The optimal broadcast program can be obtained by executing one broadcast program generation algorithm.

2.2.2 *On-demand Channels*

Let $W_O(K_O, n_O)$ denote the expected delay when the requested data item is in on-demand channels. Let $P_O^n(n_O)$ be the probability that the requested data item is in on-demand channels when there are n_O data items placed in on-demand channels. We assume that the arrival process of data item requests is a Poisson process with the arrival rate λ. It follows that the arrival process of requests received by on-demand channels is also a Poisson process with arrival rate $\lambda_O = P_O^n(n_O)\lambda$. Same as in [11], we assume the queueing buffer is infinite. Thus, the on-demand channels are modeled as an M/M/c queueing system [8] with the arrival rate λ_O, the service rate μ and the channel number c. The average service time is then $\frac{1}{\mu}$. Table 1 describes the symbols used in this paper. Let the sizes of data items and data requests be s and r, respectively. Hence, the average service time of on-demand channels can be formulated as:

$$\mu = \frac{b}{s+r}.$$

Omitting the equation manipulation which can be found in [8], the average expected delay of the on-demand channels (i.e., the M/M/c queueing system where $c = K_O$) when $\rho < 1$ is

$$\begin{aligned} &\text{Average expected delay} \\ &= \frac{1}{\mu} + \left(\frac{r^c}{c!(c\mu)(1-\rho)^2}\right) p_0, \text{ where} \end{aligned} \tag{1}$$

$$\rho = \frac{\lambda_O}{c\mu}, r = \frac{\lambda_O}{\mu}, \text{ and } p_0 = \left(\sum_{n=0}^{c-1} \frac{r^n}{n!} + \frac{r^c}{c!(1-\rho)}\right)^{-1}.$$

2.2.3 *Overall Expected Delay*

The probability that a client requests a data item in the broadcast channels is $P_B^n(n_B) = \sum_{i=1}^{n_B} Pr(R_i)$. On the other

---

[2] Note that this assumption is made for ease of presentation, and is not a restriction for the use of the algorithm proposed.

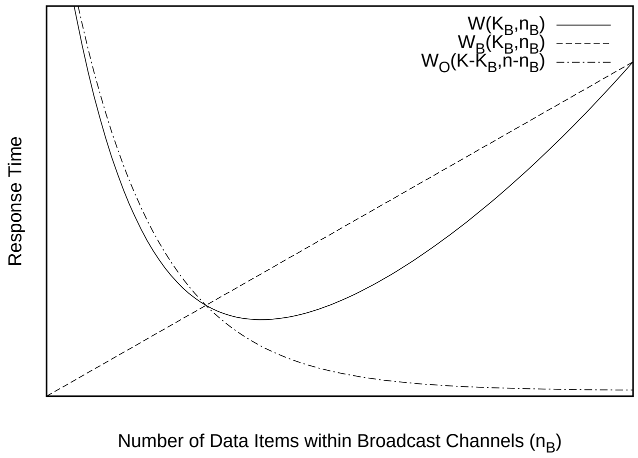


Figure 3: Trade-off for dynamic data dissemination

hand, the probability that a client requests a data item in the on-demand channels is $P_O^n(n_O) = \sum_{i=n-n_O+1}^{n} Pr(R_i) = 1 - \sum_{i=1}^{n_B} Pr(R_i) = 1 - P_B^n(n_B)$. The minimal expected delay of a data dissemination system can then be formulated as follows:

$$\begin{aligned} & W_{optimal}(K,n) \\ = \; & \min_{0 \le K_B \le K, 0 \le n_B \le n} \{W(K_B,n_B)\}, \text{ where} \qquad (2) \\ & W(K_B,n_B) \\ = \; & P_B^n(n_B) \times W_B(K_B,n_B) + (P_O^n(n_O)) \times W_O(K_O,n_O) \\ = \; & P_B^n(n_B) \times W_B(K_B,n_B) + \\ & (1 - P_B^n(n_B)) \times W_O(K - K_B, n - n_B). \end{aligned}$$

With K_B predetermined, Figure 3 shows the relationship among $W(K_B,n_B)$, $W_B(K_B,n_B)$ and $W_O(K_B,n_B)$. Note that $W_O(K_B,n_B)$ increases exponentially when n_O increases (i.e., n_B decreases). It is evident that with too few data items in broadcast channels, the volume of requests at the servers may increase beyond their capacity, making the service practically infeasible. On the other hand, the change of the response time for the broadcast data is smoother than that for the on-demand data since the expected delay is proportional to the number of data items allocated to broadcast channels. In this study, the data dissemination scheme designed will determine the proper K_B and n_B with the objective of minimizing the average expected delay of all data items.

3. BIS: SOLUTION MAPPING ON BROADCAST AND ON-DEMAND CHANNELS

In this section, we devise an algorithm based on the analytical results in Section 2.2. In Section 3.1, we first transform the problem of data and channel allocation into a search problem. A generic algorithm to solve this search problem is also derived. Then, an efficient algorithm based on binary interpolation search is devised in Section 3.2.

3.1 Problem Transformation

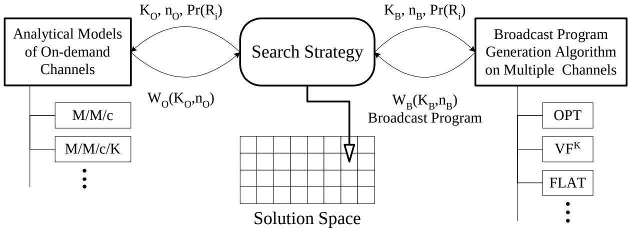


Figure 4: A generic framework of a combined use of broadcast and on-demand channels

Given K and n, for an arbitrary configuration $C(K_B,n_B)$, $W_B(K_B,n_B)$ can be obtained by executing a generation algorithm for broadcast programs, and $W_O(K-K_B,n-n_B)$ can be calculated by the analytical model of on-demand channels. As a result, the problem can be transformed into a search problem: to find the configuration with the minimal expected delay by searching all given configurations $C(K_B,n_B)$, where $0 \le K_B \le K$ and $0 \le n_B \le n$. Figure 4 shows the generic framework of algorithms to solve the problem of data and channel allocation. In this paper, since on-demand channels are modeled as an M/M/c queueing system, the expected delay of on-demand channels can be derived by Equation (1). The search strategy determines the set of configurations to be checked. Note that some configurations can be pruned by the following properties:

Property 1 All configurations that $1 \le K_B \le K-1$ and $n_B < K_B$ are pruned since those configurations will not be the optimal.

Proof: Consider an arbitrary configuration C which $1 \le K_B \le K-1$ and $n_B < K_B$. Since $n_B < K_B$, at least one broadcast channel does not be contain any data item. We can get another configuration C$'$ by reassigning the broadcast channel(s) without any data item in on-demand channel(s). P_B^n is equal to $P_B'^n$ since no data item is reassigned. Since these reassigned broadcast channels contain no data item, the expected delays in broadcast channels of C and C$'$ are equal (i.e., $W_B = W_B'$). Since C$'$ has more on-demand channels than C, W_O' is smaller than W_O. By Equation (2), we have $W' < W$, and as a result, C is not the optimal since C$'$ is better than C. **Q.E.D.**

Analogously, we have the following property.

Property 2 All configurations that $n_B = n$ and $K_B < K$ are pruned, since those configurations will not be the optimal.

Omitting straightforward proofs, we also have the following three properties.

Property 3 All configurations that $K_B = 0$ and $n_B > 0$ are pruned, since if there is no broadcast channel, neither is the data in the broadcast mode.

Property 4 All configurations that $K_O = 0$ and $n_O > 0$ are pruned, since if there is no on-demand channel, neither is the data in the on-demand mode.

Property 5 All configurations that $\rho = \frac{\lambda_O}{K_O \mu} \geq 1$ are pruned. When ρ of an M/M/c queueing system is larger than or equal to 1, the system is unstable. That is, the expected delay does not converge and will increase drastically as time advances.

3.2 Binary Interpolation Search

In light of analytical models and problem transformation described, we devise algorithm BIS to minimize the overall expected delay. BIS is a greedy algorithm to find the sub-optimal solution of the solution space. In essence, algorithm BIS is guided to explore the solution space with higher likelihood to be the optimal first. A configuration $C(K_1, n_1)$ is the local optimal in $K_B = K_1$ when $W(K_1, n_1 - 1) \geq W(K_1, n_1)$ and $W(K_1, n_1 + 1) \geq W(K_1, n_1)$. The function *LocalOptimalCheck* presented below is employed to determine whether the input configuration is the local optimal.

Function LocalOptimalCheck(K_B, n_B)

```
Calculate($K_B$,$n_B - 1$)
Calculate($K_B$,$n_B + 1$)
if ($W(K_B, n_B - 1) < W(K_B, n_B)$) then
    return MINUS
else if ($W(K_B, n_B + 1) < W(K_B, n_B)$) then
    return PLUS
else /* $W(K_B, n_B - 1) \geq W(K_B, n_B)$ and
      $W(K_B, n_B + 1) \geq W(K_B, n_B)$ */
    return LOCALOPTIMAL
end if
```

Procedure Calculate(K_B,n_B)

```
Calculate and store $W_B(K_B, n_B)$ and the
  corresponding broadcast program by employed
  broadcast program generation algorithm if they had
  not been calculated
Calculate and store $W_O(K - K_B, n - n_B)$ by Equation
  (1) if it had not been calculated
Calculate and store $W(K_B, n_B)$ by Equation (2) if it
  had not been calculated
```

LocalOptimalCheck(K_1, n_1) returns `LOCALOPTIMAL` to notify BIS to search another value of K_B when the input configuration $C(K_1, n_1)$ is the local optimal. Otherwise, it returns `MINUS` to show that $W(K_1, n_1 - 1) < W(K_1, n_1)$ and the search strategy will check another value of $n_B < n_1$. Similarly, if *LocalOptimalCheck*(K_1, n_1) returns `PLUS`, the search strategy will check another value of $n_B > n_1$.

Note that each invocation of *LocalOptimalCheck* will cause at least one execution of the broadcast program generation algorithm. That is costly. Predicting the local optimal solution is able to reduce the number of invocation of *LocalOptimalCheck*, thus reducing the total execution time. Suppose that the prediction algorithm predicts that $C(K_1, n_2)$ has the high probability to be the local optimal when $K_B = K_1$. *LocalOptimalCheck* will then check whether $W(K_1, n_2)$ is the local optimal. If $W(K_1, n_2)$ is the local optimal, BIS will search another value of K_B. Otherwise, BIS repeats the similar procedure until the configuration predicted by local optimal prediction algorithm is indeed the local optimal (i.e., *LocalOptimalCheck* returns `LOCALOPTIMAL`).

Denote the approximation of $W_B(K_B, n_B)$ and $W(K_B, n_B)$ as $W'_B(K_B, n_B)$ and $W'(K_B, n_B)$, respectively. Consider an arbitrary configuration $C(K_1, n_1)$. Suppose that *LocalOptimalCheck*(K_1, n_1) returns `MINUS`. In the beginning, the prediction algorithm checks whether $W'(K_1, n_2 - 1) \geq W'(K_1, n_2)$ where $n_2 = n_1 - 1$. If $W'(K_1, n_2 - 1) \geq W'(K_1, n_2)$, *LocalOptimalPrediction* reports n_2 as the predicted place of the local optimal. Otherwise, it checks another $n_2 = n_1 - 2$ and repeats the similar procedure until $W'(K_1, n_2 - 1) \geq W'(K_1, n_2)$. Then, *LocalOptimalPrediction* reports $C(K_1, n_2)$ as the possible configuration of the local optimal solution. The function *LocalOptimalPrediction* is as follows.

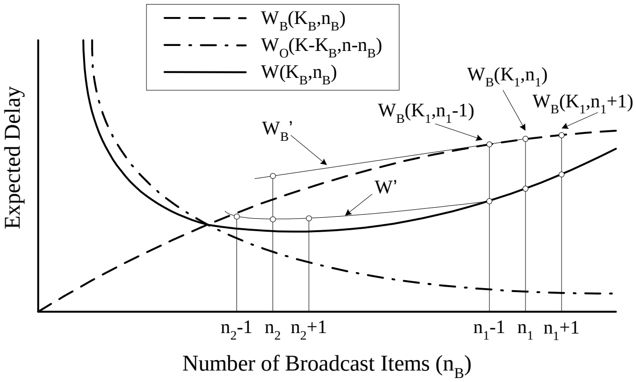


Figure 5: Execution scenario of the function *LocalOptimalPrediction* in BIS

Function LocalOptimalPrediction(K_B, n_B, α)

```
repeat
    $n_B \leftarrow n_B + \alpha$
    Calculate $W'(K_B, n_B)$ and $W'(K_B, n_B + \alpha)$ by
      Equation 3
until ($W'(K_B, n_B + \alpha) \geq W'(K_B, n_B)$)
return $n_B$
```

Figure 5 shows the method to calculate $W'_B(K_B, n_B)$ and $W'(K_B, n_B)$ by interpolation. Suppose that BIS selects an configuration $C(K_1, n_1)$ and *LocalOptimalCheck*(K_1, n_1) returns `MINUS` since $W(K_1, n_1 - 1) < W(K_B, n_1)$. The $W'_B(K_1, n_2)$ is obtained by the interpolation of $W_B(K_1, n_1)$ and $W_B(K_1, n_1 - 1)$. Therefore, we have the following equation:

$$\frac{W'_B(K_1, n_2)}{n_2 - n_1} = \frac{W_B(K_1, n_1 + \alpha) - W_B(K_1, n_1)}{\alpha}, \text{ where}$$

$$\alpha = \begin{cases} 1 & : \text{ if } \textit{LocalOptiamCheck(}K_1, n_1\textit{)} \text{ returns } \texttt{PLUS} \\ \text{-1} & : \text{ if } \textit{LocalOptiamCheck(}K_1, n_1\textit{)} \text{ returns } \texttt{MINUS} \end{cases}$$

Solving the above equation, we have $W'_B(K_1, n_2)$ as:

$$W'_B(K_1, n_2) = \frac{1}{\alpha} \times (n_2 - n_1) \times (W_B(K_1, n_1 + \alpha) - W_B(K_1, n_1))$$

With $W'_B(K_1, n_2)$, $W'(K_1, n_2)$ can be obtained by the following equation:

$$\begin{aligned} & W'(K_1, n_2) \\ = \; & P^n_B(n_2) \times W'_B(K_1, n_2) + \\ & (1 - P^n_B(n_2)) \times W_O(K - K_1, n - n_2) \qquad (3) \end{aligned}$$

The detailed steps of algorithm BIS are shown below. The complexity of the algorithm BIS depends on the used broadcast program generation algorithm and the accuracy of *LocalOptimalPrediction*. If VF^K is used, the complexity

Table 2: System parameters used in the simulation

Parameters	Values
Channel bandwidth (b)	8000 bps
Data item size (s)	1000 bytes
Data request size (r)	10 bytes
Data request rate for each user	1/sec
Number of mobile users (N)	100
Parameter of Zipf distribution (θ)	2

of average case is the product of $O(K \log n)$ and the complexity of VF^K.

Algorithm BIS

Input: The data items sorted by their access frequencies and the number of communication channels.
Output: The number of broadcast channels and on-demand channels, the number of data items within broadcast and on-demand channels, and the resulting broadcast program.

```
Construct the solution space and prune configurations according to the properties 1-5
Mark the unavailable configurations (i.e., $K_B > K$ or $K < 0$ or $n_B > n$ or $n_B < 0$) as calculated and set $W_B(K_B, n_B)$, $W_O(K - K_B, n - n_B)$ and $W(K_B, n_B)$ to be $\infty$.
for all pruned configuration $C(K_B, n_B)$ do
    Set $W_B(K_B, n_B)$, $W_O(K - K_B, n - n_B)$, and $W(K_B, n_B)$ to be $\infty$ and mark them as calculated
end for
for ($K_B \leftarrow 0$ to $K$) do
    Let $n_B$ be the middle of un-pruned configurations
    Calculate($K_B, n_B$)
    while (LocalOptimalCheck($K_B, n_B$)$\neq$LOCALOPTIMAL) do
        if (LocalOptimalCheck($K_B, n_B$)=PLUS) then
            $n'_B \leftarrow$ LocalOptimalPrediction($K_B, n_B, 1$)
        else /* LocalOptimalCheck(i, j)=MINUS */
            $n'_B \leftarrow$ LocalOptimalPrediction($K_B, n_B, -1$)
        end if
    end while
    Keep track of the optimal $W_{optimal}(K, n) \leftarrow W(K_B, n_B)$, the corresponding configuration $C(K_B, n_B)$ and broadcast program in the broadcast disk array.
end for
```

4. PERFORMANCE EVALUATION

In order to evaluate the performance improvement achieved by algorithm BIS, we have designed a simulation model of a data dissemination system which is described in Section 4.1. To compare the quality of solutions of all algorithms, two experiments are conducted and compared in Section 4.2. The impact of employing BIS is evaluated in Section 4.3.

4.1 Simulation Model

Similarly to the work in [15], we set the system parameters as shown in Table 2. Also, the access frequency of ith data item is assumed to be $Pr(R_i) = \frac{(\frac{1}{i})^\theta}{\sum_{j=1}^{n}(\frac{1}{j})^\theta}$ where θ is the parameter of the Zipf distribution [7]. The access frequencies become increasing skewed as θ increases. Let N denote the number of mobile users. The total request arrival rate λ is equal to N since the request arrival rate for each client is one per second. The simulator is coded in C++.

To compare the effect of BIS on the quality of solutions and execution time, we conduct two experiments with the values of n and K varied. Flat broadcast program (denoted as FLAT), which allocates data items within broadcast channels with equal appearance frequencies, is also implemented in order to evaluate the benefit of using hierarchical broadcast program. For each configuration, since the optimal broadcast program can be obtained by OPT, the optimal data and channel allocation can be obtained by collecting all optimal broadcast program of all possible configurations and taking the optimal one among them. In addition to BIS, we also implement the exhaustive search (abbreviated as ES) for comparison purposes.

4.2 Comparison of Quality of Solutions

Figure 6a and 6b show the expected delays of (1) ES with OPT, (2) BIS with OPT, (3) ES with VF^K and (4) BIS with VF^K on these two experiments. As shown in Figure 6, the expected delays of all algorithms employing hierarchical broadcast generation program are better than those employing FLAT broadcast program in all experiments, showing the advantage of using hierarchical broadcast program generation algorithms. The solutions obtained by VF^K-based algorithms are close to OPT-based algorithms because the results of VF^K are close to OPT. It is seen that the solutions obtained by algorithm BIS are all of very high quality. In fact, in our experiments, the solutions of BIS-based algorithms are all equal to the solutions of ES-based.

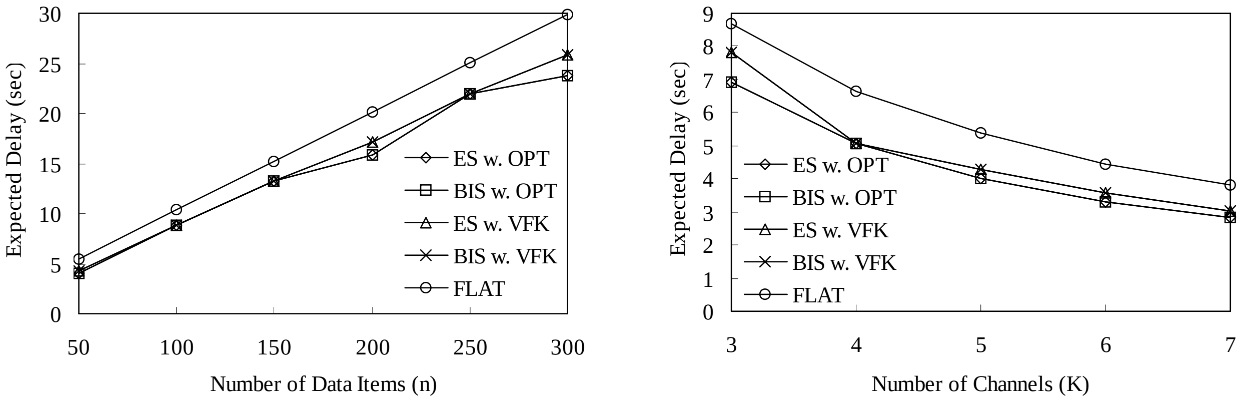
4.3 The Effect of BIS

Figures 7 and 8 show the execution time of each algorithm with the values of n and K varied, respectively. The execution times of all algorithms are proportional to the size of solution space. The size of solution space increases as the n and K increase since the size of the solution space is proportional to $K \times n$. Since the execution times of ES-based algorithms are more sensitive to the size of the solution space than the BIS-based algorithms, BIS-based algorithms are more scalable when the values of K and n become large.

In all, we observed that (1) the execution time of BIS-based algorithms is much faster than that of ES-based algorithms when the same broadcast program generation algorithm is employed, and (2) BIS-based algorithms are more scalable than ES-based algorithms.

5. CONCLUSIONS

In this paper, we explored the problem of dynamic data and channel allocation with the number of communication channels and the number of data items given. We first derived the analytical models of the expected delay on broadcast and on-demand channels. Then, we transformed this problem into to a guided search problem. In light of the theoretical properties derived, we devised algorithm BIS based on binary interpolation search to obtain solutions of high quality efficiently. Our simulation results showed that the solution of our algorithm is of very high quality and is in fact very close to the optimal one. Sensitivity analysis on several parameters, including the number of data items and the number of communication channels, was conducted.



(a) (b)

Figure 6: The expected delay with the value of (a) n and (b) K varied

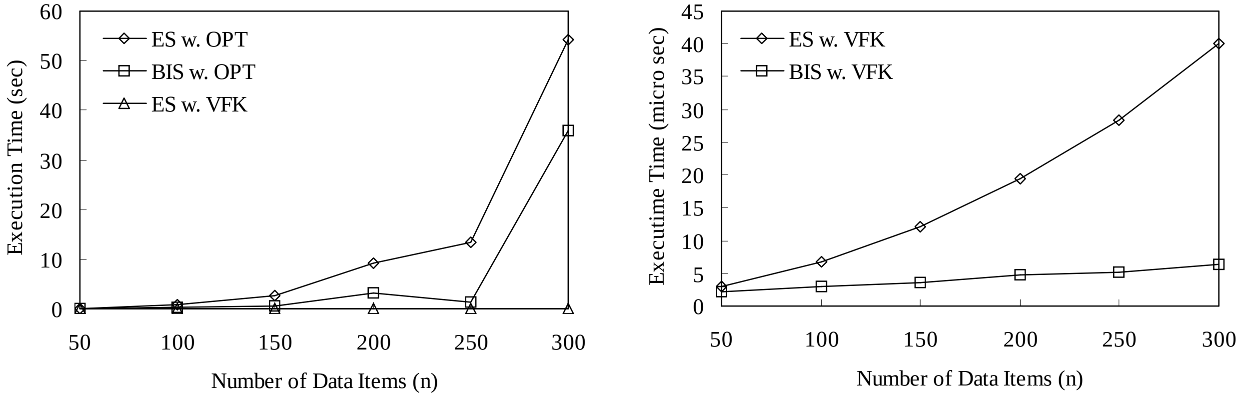


Figure 7: The execution time with the value of n varied

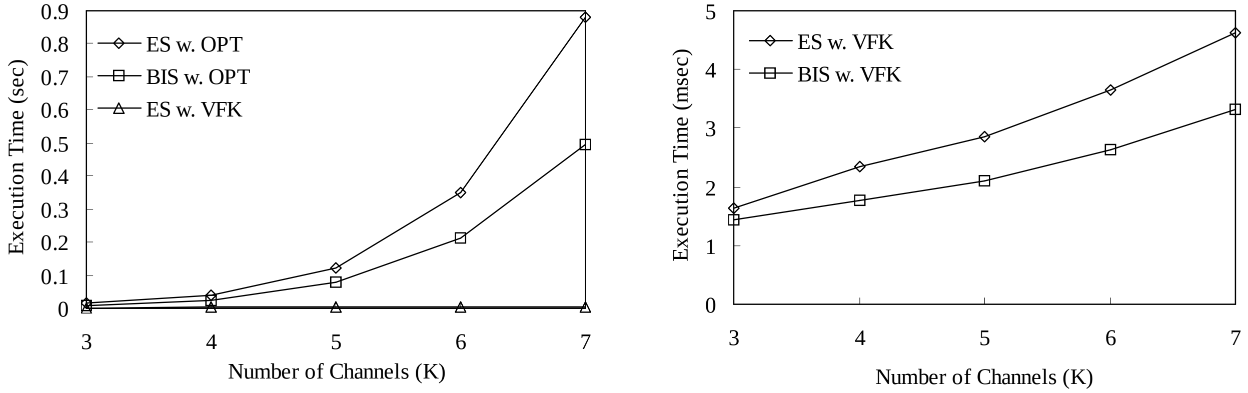


Figure 8: The execution time with the value of K varied

ACKNOWLEDGEMENT

The authors are supported in part by the Ministry of Education Project No. 89-E-FA06-2-4-7 and the National Science Council, Project No. NSC 89-2219-E-002-028 and NSC 89-2218-E-002-028, Taiwan, Republic of China.

6. REFERENCES

[1] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik. Broadcast Disks: Data Management for Asymmetric Communication Environments. In *Proceedings of the ACM SIGMOD Conference*, pages 198–210, March 1995.

[2] S. Acharya and S. Muthukrishnan. Scheduling on-demand Broadcasts: New Metrics and Algorithms. In *Proceedings of the 9th ACM International Conference on Mobile Computing and Networking*, pages 43–94, October 1994.

[3] D. Aksoy and M. J. Franklin. Scheduling for Large-Scale On-Demand Data Broadcasting. In *Proceedings of INFOCOM Conference*, pages 651–659, March 1998.

[4] D. Barbara. Mobile Computing and Databases - A Survey. *IEEE Transactions on Knowledge and Database Engineering*, 11(1):108–117, January/February 1999.

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* The MIT Press, 1996.

[6] D. H. Dunham. Mobile Computing and Databases. *Tutorial of the 14th International Conference on Data Engineering*, February 1998.

[7] J. Gray, P. Sundaresan, S. Englert, and K. Baclawski. Quickly Generating Billion-Record Synthetic Databases. In *Proceedings of ACM SIGMOD Conference*, pages 243–252, March 1994.

[8] D. Gross and C. M. Harris. *Fundamentals of Queueing Theory.* John Wiley & Sons,Inc., 3rd edition, 1998.

[9] Q.L. Hu, D.L. Lee, and W.-C. Lee. Dynamic Data Delivery in Wireless Communication Environments. In *Proceedings of International Workshop on Mobile Data Access*, pages 218–229, November 1998.

[10] Q.L. Hu, W.-C. Lee, and D.L. Lee. Indexing Techniques for Wireless Data Broadcast under Data Clustering and Scheduling. In *Proceedings of the 8th International Conference on Information and Knowledge Management*, pages 351–718, November 1999.

[11] T. Imielinski and S. Viswanathan. Adaptive Wireless Information Systems. In *Proceedings of SIGDBS (Special Interest Group in DataBase Systems) Conference*, October 1994.

[12] T. Imielinski, S. Viswanathan, and B.R. Badrinath. Data on Air: Organization and Access. *IEEE Transactions on Knowledge and Data Engineering*, 9(9):353–372, June 1997.

[13] J. Jing, A. Helal, and A. Elmagarxid. Client-Server Computing in Mobile Environments. *ACM Computing Surveys*, 31:117–157, June 1999.

[14] W.-C. Lee, Q.L. Hu, and D.L. Lee. Channel Allocation Methods for Data Dissemination in Mobile Computing Environments. In *Proceedings of 6th IEEE International Symposium on High Performance Distributed Computing*, pages 274–281, August 1997.

[15] W.-C. Lee, Q.L. Hu, and D.L. Lee. A Study on Channel Allocation for Data Dissemination in Mobile Computing Envirolments. *ACM/Baltzer Mobile Networks and Applications*, 4(5):117–129, May 1999.

[16] C.-W. Lin and D.L. Lee. Adaptive Data Delivery in Wireless Communication Environments. In *Proceedings of the 20th IEEE International Conference on Distributed Computing Systems*, pages 444–456, April 2000.

[17] S.-C. Lo and A. L. P. Chen. Optimal Index and Data Allocation in Multiple Broadcast Channels. In *Proceedings of the 16th International Conference on Data Engineering*, pages 293–702, March 2000.

[18] W.-C. Peng and M.-S. Chen. Dynamic Generation of Data Broadcasting Programs for a Broadcast Disk Array in a Mobile Computing Environment. In *Proceedings of the 9th ACM International Conference on Information and Knowledge Management*, pages 38–45, November 2000.

[19] W.-C. Peng and M.-S. Chen. Exploring User Moving Patterns to Improve the Allocation of Shared Data in a Mobile Computing Environment. In *Proceedings of 3rd International Conference on Mobile Data Management*, January 2002.

[20] K. Prabhakara, K. A. Hua, and J.H. Oh. Multi-Level Multi-Channel Air Cache Designs for Broadcating in a Mobile Environment. In *Proceedings of the 16th International Conference on Data Engineering*, pages 167–186, February-March 2000.

[21] N. Shivakumar and S. Venkatasubramanian. Efficient Indexing for Broadcast Based Wireless Systems. *ACM/Baltzer Mobile Networks and Applications*, 4(6):433–446, January 1996.

[22] K. Stathatos, N. Roussopoulos, and J. S. Baras. Adaptive Data Broadcast in Hybrid Networks. In *Proceedings of the 23rd International Conference on Very Large Data Bases*, pages 326–335, 1997.

[23] C.-J. Su and L. Tassiulas. Joint Broadcast Scheduling and User's Cache Management for Efficient Information Delivery. In *Proceedings of the 4th ACM International Conference on Mobile Computing and Networking*, pages 33–42, October 1998.

[24] WAP application in Nokia. http://www.nokia.com/corporate/wap/future.html.

[25] WAP application in Unwired Planet, Inc. http://phone.com.

[26] J.L. Xu, Q.L. Hu, D.L. Lee, and W.-C. Lee. SAIU: An Efficient Cache Replacement Policy for Wireless On-demand Broadcasts. In *Proceedings of the 9th ACM International Conference on Information and Knowledge Management*, pages 46–53, November 2000.

[27] J.L. Xu, Q.L. Hu, W.-C. Lee, and D.L. Lee. An Optimal Cache Replacement Policy for Wireless Data Dissemination under Cache Consistency. In *Proceedings of the 30th International Conference on Parallel Processing*, September 2001.

[28] J. X. Yu, T. Sakata, and K.L. Tan. Statistical Estimation of Access Frequencies in Data Broadcasting Environments. *ACM/Baltzer Wireless Networks*, 6(2):89–98, March 2000.