TOSA: A Near-Optimal Scheduling Algorithm for Multi-Channel Data Broadcast *

Baihua Zheng
School of Information Systems
Singapore Management University
bhzheng@smu.edu.sg

Xia Wu Xing Jin Dik Lun Lee
Hong Kong University
of Science and Technology
{sallywu,csvenus,dlee}@cs.ust.hk

ABSTRACT

Wireless broadcast is very suitable for delivering information to a large user population. In this paper, we concentrate on data allocation methods for multiple broadcast channels. To the best of our knowledge, this is the first allocation model that takes into the consideration of items' access frequencies, items' lengths, and bandwidth of different channels. We first derive the optimal average expected delay for multiple channels for the general case where data access frequencies, data sizes, and channel bandwidths can all be non-uniform. Second, we develop TOSA, a multi-channel allocation method that does not assume a uniform broadcast schedule for data items on the same channel. TOSA is based on the idea of two-level data allocation, i.e., a high-level optimization step for allocating data to the channels, followed by a low-level optimization step to schedule data within a channel. We show that TOSA achieves near-optimal performance in terms of average waiting time and significantly outperforms the existing algorithms.

Keywords

Multiple channels, wireless broadcast, mobile computing, scheduling

1. INTRODUCTION

With the development of wireless communication technologies and the popularity of mobile devices, more and more people are accessing information from remote servers without maintaining physical connections. India is expected to have more mobile subscribers than fixed subscribers during 2004. Nokia predicts that the number of mobile users in Russia will exceed 60 million by 2008, a 200% increase over the subscriber base of 2003. The explosive growth in the number of mobile users has created a very strong demand for real-time information access.

The wireless broadcast mode is a common wireless information dissemination method. It is especially efficient for delivering information to a large number of clients simultaneously, since the cost at the server site will not change as the number of clients increases. Actually, wireless broadcast, which has long been used for radio and TV signal transmission, is a natural solution to address the scalability issue. The smart personal objects technology (SPOT), recently announced by Microsoft at 2003 International Consumer Electronics Show (CES), has further exploited the feasibility of using wireless data broadcast in the pervasive computing era. With a continuous broadcast network using FM radio subcarrier frequencies, SPOT-based devices such as watches and alarms, can continuously receive timely, location-aware, personalized information.

Although wireless connection provides users with unlimited mobility and hence the users can enjoy the cable-free world, it also introduces some limitations unique to the wireless world. The portability of mobile devices makes the resources available at the client's side very scarce, and the communication ability of clients and servers is asymmetric. The applications developed in mobile environments have to cater to all these limitations. Since the power supply is the key resource without which the device cannot work, energy efficiency is one of the major requirements of the wireless information delivery method.

Scheduling is a natural solution to address the efficiency issue. Given the fact that users may have different requests, scheduling algorithms can determine the broadcast sequence of data items to minimize the access time, reduce power consumption, or increase bandwidth utilization. In brief, scheduling algorithms for wireless broadcasting are motivated by the desire to satisfy clients' requirements efficiently with as little consumption of energy as possible.

Lots of scheduling algorithms for single-channel environments have been proposed in the literature. However, in some situations, it is not always possible to combine multiple low-bandwidth physical channels into a single channel [9]. Even when a public channel is available, sometimes users

*MDM 05, Ayia Napa Cyprus, 2005

prefer partitioning it into several independent channels. For example, several information service providers may share one channel while they are targeting different customers. By breaking down the channel into several smaller ones, they can deliver information to their potential customers more efficiently without bothering other customers. Consequently, this work focuses on multi-channel data dissemination in a wireless broadcast system.

In this paper, we study data allocation and scheduling algorithms for multiple channels. We derive the optimal average expected delay for multiple channels ($MCAED$) and the conditions under which optimality can be achieved. Our derivation is based on non-uniform data access frequencies, data size, and channel bandwidths. Furthermore, we propose a novel two-level scheduling strategy, called TOSA, to achieve a near-optimal performance. Simulation results show that TOSA outperforms the existing methods and achieves a performance that is less than 1% shy of the theoretical optimal value.

The rest of this paper is organized as follows. Section 2 provides the background information, including a description of the system models, a definition of the problem, and a review of related work. Section 3 presents a detailed analysis of the optimal scheduling algorithm in terms of minimizing clients' average waiting time. As the optimization algorithm is NP-complete, an approximation algorithm, named $TOSA$, is proposed to approach the optimum performance. Simulation results are shown in Section 4. Finally, we conclude this paper in Section 5.

2. BACKGROUND

2.1 System Model and Problem Definition

A cellular mobile network similar to that in [1] is adopted in this paper as the mobile computing infrastructure. It consists of two distinct sets of entities: mobile clients and fixed hosts, as shown in Figure 1. Some of the fixed hosts, called mobile support stations (MSS), are augmented with wireless interfaces. An MSS can communicate with mobile clients within its radio coverage area called a *wireless cell*. MSS have either a local or a remote database which contains all the data items, and publish data via wireless broadcast channels. All the clients within the wireless cell can receive the broadcast data as long as they are actively listening to that channel. In a wireless environment, there are K independent channels available in the broadcast system, each denoted by C_i, $1 \leq i \leq K$. The available bandwidth of channel C_i is denoted by B_i, and maybe B_i is different from B_j ($i \neq j$, $i, j \in [1, K]$). We are interested in the situation where K is larger than one.

Multi-channel broadcast provides information to clients via multiple channels. For instance, radio systems allow clients to tune into different channels to enjoy various programs. A client can only listen to one wireless channel at one time and it can switch from one channel to another freely. The *average wait time*, i.e., the time duration between a client's issuing a query and the client's receiving the result, is a common metric to evaluate the performances of different broadcast schemes. Consequently, the main objective of this paper is to provide a near-optimal scheduling algorithm in order to enable clients to receive items of interest efficiently.

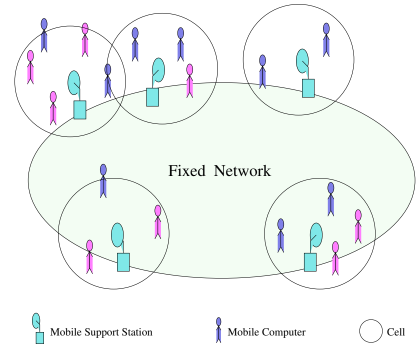


Figure 1: Mobile Computing System Models

Suppose the database contains N data items, denoted by d_j, with $j \in [1, N]$. Each item is allocated to only one channel and interleaved with other data items. The minimal periodic broadcast duration within which each item is broadcast at least once is defined as a *broadcast cycle*. The time difference between two continual broadcast slots for the same item is called the *spacing* s_i of that item d_i. Figure 2 shows a schedule with eight data items on two broadcast channels with spacing s_1, s_2, and s_3 for d_1, d_2, and d_3, respectively.

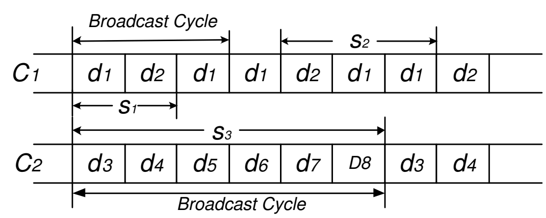


Figure 2: Multi-Channel Broadcasting

2.2 Related Work

Lots of research has been done on generating broadcast programs on multiple channels. In terms of index mechanisms, Shivakumar et al. studied efficient indexing methods for broadcast-based wireless systems, using binary alphabetic-huffman trees to make index blocks quickly accessible [11]. Lo and Chen proposed a parameterized scheduling algorithm for optimal index and data allocation, subject to the constraint that no data item is replicated in a broadcast cycle [8]. Besides developing the index mechanisms, scheduling is also an important area. Some work focused on broadcasting dependent data, whose access sequence is pre-ordered. Hurson et al., trying to minimize the overall access time, proposed an allocation algorithm of dependent data based on some heuristics for multiple-channel environments [6]. Lee and his colleagues introduced an efficient algorithm for mobile environments to answer queries asking for multiple data items [7]. Recently, Huang and Chen proposed a scheme based on a *generic algorithm* to handle a similar problem [5].

In this paper, we conduct a study on allocating independent data items into multiple channels whose bandwidth may be different. The queries issued by clients are assumed to only request a single item. Lots of work has been done to minimize the average expected delay, i.e., the average wait time of clients for their requested data items. A brief summary is provided as follows.

The FLAT algorithm adopts a simple yet not so efficient allocation algorithm, equally allocating the items to each channel. Consequently, the expected delay for the most popular item is the same as that for an item seldom asked for by clients [12]. Peng and Chen proposed VF^k algorithm, which skews the amount of data allocated to each channel [4]. Prabhakara et al. proposed a skewed-allocation algorithm, called *BP*, which guarantees an equal access probability of each channel based on the concept of Bin Packing [9]. Yee and Navathe proposed the *DP* algorithm, which uses dynamic programming to partition the data items on multiple channels and achieves an optimal solution [14]. *GREEDY*, which will be explained in detailed in the next section, can achieve a similar performance to *DP* while at a much cheaper cost [14]. The authors have further extended the work to take into consideration the hopping cost [13]. Although these five algorithms share the same objective, they assign different weights to complexity and performance. It has been proven that GREEDY has the best tradeoff between performance and simplicity [14].

3. MULTI-CHANNEL BROADCAST SCHEDULING ALGORITHMS

Although lots of related algorithms have been proposed to schedule the broadcast problems in a multi-channel environment, none could guarantee an optimal or a near-optimal performance. To remedy this situation, we derive a solid theoretical model which gives the lower bound of $MCAED$, i.e., the *multi-channel average expected delay*. Since the optimization problem is NP-complete, we propose an approximation algorithm, $TOSA$, to achieve near-optimal $MCAED$. In this section, we first introduce two existing algorithms, *Log-time* and *GREEDY*, which prompted the work presented in this paper. Thereafter, the theoretical model and TOSA are presented.

3.1 Preliminary

This work is based on several assumptions. First, the access probability p_i of each item d_i is known and stays the same during the broadcast. Otherwise, the scheduling algorithm has to be re-run to produce a new broadcast program. Second, each query only requests one data item. The time needed to transmit one data item of unit length per unit bandwidth is defined as one time tick. The average wait time is evaluated in the unit of time tick and the data item size is represented by the ratio of the size to the unit length. In order to facilitate the description, the terminologies defined in Table 1 are used in the rest of this paper.

The multi-channel average expected delay (MCAED), defined in Equation 1, is the major performance metric for nearly all the scheduling algorithms in the broadcast environment.

$$MCAED = \sum_{i=1}^{K} \sum_{d_j \in C_i} w_j p_j \tag{1}$$

In the following, two typical strategies, log-time and GREEDY, are described.

Notation	Description
K	the number of available channels
C_i	the ith channel, $(0 < i <= K)$
B_i	the bandwidth of channel C_i, $(0 < i <= K)$
N	the number of available data items
d_i	the ith data item, $(0 < i <= N)$
l_i	the length of data item d_i compared with unit length, $(0 < i <= N)$
N_i	the number of items allocated to channel C_i
p_i	the access probability of data item d_i
w_i	expected wait time for item d_i
s_i	spacing between two continual broadcast slots of item d_i
A_i	the summation of $\sqrt{p_j \times l_j}$ for all the items d_j allocated to channel C_i

Table 1: Terminology Description

Log-time algorithm: The log-time algorithm was proposed by Hameed and Vaidya to efficiently schedule data items on single and multiple channels. One of the most significant results of log-time is that AED, the average expected delay in a single channel, is minimized when the instances of each data item d_i are equally spaced with spacing $s_i = (\sum_{j=1}^{N} \sqrt{p_j \times l_j})\sqrt{\frac{l_i}{p_i}}$. The optimal AED, denoted by $AED_{optimal}$, can achieve its optimal value as follows.

$$AED_{optimal} = \sum_{i=1}^{N} \frac{s_i}{2} \times p_i = \frac{1}{2}(\sum_{i=1}^{N} \sqrt{p_i \times l_i})^2$$

The real scheduling algorithm proposed needs to maintain two parameters, B_i and C_i, for each item d_i. B_i is the earliest time when the next instance of item d_i should begin transmission and its initial value is 0. C_i equals the summation of B_i and s_i. A parameter T is also maintained to simulate the current broadcast time, which is the broadcast time for all the scheduled items. Initially, T equals 0, and it will be increased by l_j after the broadcast of each instance of d_j. The scheduling algorithm is repeated. For each iteration, all the items with B_i smaller than T are selected as the candidates. The candidate with the smallest C_i is chosen for broadcast. At the same time, B_i and C_i of that item are updated correspondingly. $AED_{optimal}$ provides a lower-bound performance, and AED under a real broadcast program is in most cases larger than $AED_{optimal}$.

Hameed and Vaidya also investigated scheduling algorithms for multiple channels [2]. However, their study was based on the assumption that a client could only listen to the first j consecutive channels simultaneously, with $j \in [1, k]$. This assumption limited the results of the log-time algorithm and was inconsistent with a lot of real situations.

GREEDY algorithm: The GREEDY algorithm is a more recent algorithm proposed to achieve a near-optimal performance in scheduling data items on multiple channels[14]. It assumed that each item had a unit length and each channel had the same bandwidth. Consequently, parameters l_i and B_i were ignored. It also assumed that N_i items were cyclically broadcast on channel C_i, and the expected delay w_j

of receiving d_j on C_i was $N_i/2$. Given this assumption, the optimal $MCAED$ of GREEDY can be derived as follows.

$$MCAED = \frac{1}{2}\sum_{i=1}^{K}(N_i \sum_{d_j \in C_i} p_j)$$

Given K channels, the GREEDY algorithm aims at partitioning the whole dataset into K clusters. Suppose there are two Channels, C_i and C_j, and $\sum_{d_l \in C_i} p_l > \sum_{d_m \in C_j} p_m$. It has been proven that in the optimal solution, $\forall d_l \in C_i, \forall d_m \in C_j$, $p_l \geq p_m$. Based on this theorem, the GREEDY algorithm employs a recursive approach to achieving the near-optimal performance. Initially, all the items sorted according to their access probabilities form a candidate set. Each item excluding the first and the last items within this set is a potential split point to partition the set, and the item who brings the best MCAED is chosen as the real split point. This step continues until the original dataset is partitioned into K sets.

The major factor impacting the average performance of different scheduling algorithms is the various access frequencies of data items. The GREEDY algorithm takes this into account and groups the items into clusters with similar access probabilities. However, it still employs the FLAT method for each channel to cyclically broadcast all the items allocated to it. The demands for the items of the same channel can still be very different, especially when the number of channels is small and the access frequencies of data items are really skewed. Therefore, an ideal scheduling algorithm should have two-level clustering. The higher level assigns items to different channels, and the lower level determines the broadcast programs for each channel. Access frequency has to be considered during both steps. The research presented in this paper is motivated by this intention. The second problem with GREEDY is that it assumes each channel has the same bandwidth and each item the same length. Our algorithm will assume a more general scenario, where the bandwidth of channels and the item size of the data could be different.

d_i	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8
p_i	0.5	0.2	0.1	0.1	0.07	0.01	0.01	0.01

Table 2: Access Probabilities

In order to facilitate understanding and comparisons of different algorithms, a running example is introduced. Table 2 shows the access probabilities of all the data items, where K equals two, N equals eight, and two channels have equal bandwidth. Table 3 shows the broadcast program produced by the GREEDY algorithm. Obviously, this is not the optimal solution. If the broadcast program of the first channel is like (d_1, d_2, d_1) as shown in Figure 2, the overall MCAED can be reduced to 1.575. This example further demonstrates the inherent problem of treating all the items at each channel equally.

3.2 Theoretical Lower Bound of MCAED

In this subsection, we derive the theoretical lower bound for MCAED that caters to non-uniform data access frequencies, non-uniform data sizes, and non-uniform channel bandwidths.

GREEDY	
Allocation:	$C_1 : d_1, d_2$
	$C_2 : d_3, d_4, d_5, d_6, d_7, d_8$
Cyclically Broadcast in each channel:	
$MCAED$=$\frac{1}{2} \times \sum_{i=1}^{K}(N_i \sum_{d_j \in C_i} p_j)$=1.600	

Table 3: The Running Example under GREEDY

Without loss of generality, the arrival of client requests can be simulated by a Poisson process, where $w_j = s_j/2$. Replacing w_j, the expected wait time for data item d_j, with d_j's spacing and the bandwidth of the channel that d_j is assigned to, we can rewrite Equation 1 as follows.

$$MCAED = \sum_{i=1}^{K}(\frac{\sum_{d_j \in C_i} s_j p_j}{2 \times B_i}) \quad (2)$$

As we mentioned before, both the channel allocation and the intra-channel broadcast program affect the final performance. Therefore, the optimal allocation method has to consider different factors in both processes. Given the access probabilities and lengths of all the data items, and the bandwidth of each channel, the theoretical optimization can be defined by Theorem 1.

Theorem 1 Assuming that the instances of each item d_i are equally spaced, $MCAED$ is minimized when the following condition holds:

$$\sum_{d_j \in C_m} \sqrt{p_j \times l_j} = \frac{\sqrt{B_m}}{\sum_{l=1}^{K}\sqrt{B_l}} \times \sum_{i=1}^{N}\sqrt{p_i \times l_i}, \quad 1 \leq m \leq K$$

and the optimal $MCAED$, denoted by $MCAED^*$, is:

$$MCAED^* = \frac{K}{2} \times (\frac{\sum_{i=1}^{N}\sqrt{p_i \times l_i}}{\sum_{l=1}^{K}\sqrt{B_l}})^2 \quad (3)$$

Proof: In a single-channel environment, the average waiting time can achieve its optimal value $\frac{1}{2}(\sum_{i=1}^{N}\sqrt{p_i \times l_i})^2$ when the instances of each item d_i are equally spaced with spacing $(\sum_{j=1}^{N}\sqrt{p_j \times l_j})\sqrt{\frac{l_i}{p_i}}$. Consequently, based on Equation 2 and letting $A_i = \sum_{d_j \in C_i}\sqrt{p_j \times l_j}$, we can obtain:

$$\begin{aligned} MCAED &= \frac{1}{2}\sum_{i=1}^{K}(\sum_{d_j \in C_i}\frac{s_j p_j}{B_i}) \\ &= \frac{1}{2}\sum_{i=1}^{K}(\sum_{d_j \in C_i}(\frac{A_i \times \sqrt{p_j \times l_j}}{B_i})) \\ &= \frac{1}{2}\sum_{i=1}^{K}\frac{A_i^2}{B_i} \end{aligned}$$

In accordance with our assumption, l_j, p_j, and B_i are fixed during the broadcast. According to Cauchy Formula [3], $MCAED$ is minimized when

$$\frac{A_i}{\sqrt{B_i}} = \frac{1}{K}\sum_{j=1}^{K}\frac{A_j}{\sqrt{B_j}}$$

Let ϕ denote $\frac{1}{K}\sum_{j=1}^{K}\frac{A_j}{\sqrt{B_j}}$, we have

$$\frac{A_i}{\sqrt{B_i}} = \phi \implies \sum_{i=1}^{K} A_i = \phi \sum_{l=1}^{K} \sqrt{B_l}$$

$$\implies \phi = \frac{\sum_{i=1}^{K} A_i}{\sum_{l=1}^{K}\sqrt{B_l}} = \frac{\sum_{i=1}^{N}\sqrt{p_i \times l_i}}{\sum_{l=1}^{K}\sqrt{B_l}}$$

Therefore, $MCAED$ is minimized when

$$A_i = \sqrt{B_i} \times \phi = \sqrt{B_i} \times \frac{\sum_{j=1}^{N}\sqrt{p_j \times l_j}}{\sum_{l=1}^{K}\sqrt{B_l}}$$

and

$$MCAED^* = \frac{K}{2} \times (\frac{\sum_{i=1}^{N}\sqrt{p_i \times l_i}}{\sum_{l=1}^{K}\sqrt{B_l}})^2$$

□

According to Theorem 1, $MCAED$ is optimized only when the instances of each item are equally spaced and $\frac{A_i}{\sqrt{B_i}}$ is a constant ϕ. Obviously, these two conditions cannot always be satisfied. Therefore, $MCAED^*$, which in general is not achievable, represents the *lower bound* of $MCAED$. Based on this, we propose TOSA, an approximation algorithm, to approach $MCAED^*$. The details of TOSA will be presented in the next section.

3.3 TOSA: A Two-Level Optimization Scheduling Algorithm

A *two-level optimization scheduling algorithm (TOSA)* is proposed in this paper to achieve the near-optimal performance based on Theorem 1. It is a hybrid scheduling algorithm. Its high-level optimization strategy clusters data items into channels, and a low-level optimization schedules the items on each channel in order to guarantee the average performance.

The partitioning of N items, given their access probabilities and item length, on K channels such that $\sum_{i=1}^{K}\frac{A_i^2}{B_i}$ is minimal is an NP-Complete problem. However, it is observed that $\sum_{i=1}^{K}\frac{A_i^2}{B_i}$ is minimized when each channel C_i shares the same $\frac{A_i}{\sqrt{B_i}}$. Therefore, the major objective of high-level scheduling of TOSA is to schedule items such that $\frac{A_i}{\sqrt{B_i}} \approx \frac{A_j}{\sqrt{B_j}}, \forall i, j \in [1, K]$ and $i \neq j$.

Firstly, the *initialization* step allocates N items to K channels according to the items' access probabilities, the items' lengths, and the channels' available bandwidth. The basic idea is to balance the ratio of As to $\sqrt{B}$s. The items are sorted based on the product of the access probabilities and the items' lengths, and the channels are sorted according to the available bandwidth. Let B equal $\frac{\sum_{i=1}^{K}\sqrt{B_i}}{\sqrt{B_K}}$, it sequentially groups every $2B$ items into one set. For each set, it adopts a circuitous allocation strategy. Initially, B items are assigned to channels C_1 to C_K and the number of items allocated to the channel is proportional to the available bandwidth. The distribution of the second B items are from channel C_K down to channel C_1. Algorithm 1 provides the pseudo-code, with time complexity *O(NlogN)*. Like in our example, initially, items d_1, d_4, d_5, and d_8 are allocated to channel C_1, and C_2 has the rest, as shown in Table 4.

Algorithm 1 Initialization

Input: a set of N items with access probabilities, available channels with bandwidth;
Output: the partition of the N data items;
Procedure:

sort items so that $\forall i <= j$, $\sqrt{p_i l_i} >= \sqrt{p_j l_j}$;
sort Channels so that $\forall i <= j$, $B_i >= B_j$;
let $B = \sum_{i=1}^{K}\sqrt{B_i}/\sqrt{B_K}$; $T_i = sqrtB_i/B_K$;
for $(i = 1;\ i \leq N;\ i+ = 2B)$ **do**
  for $(j = 1,\ c_i = 0;\ j \leq K;\ j++,\ c_i+ = T_j)$ **do**
    allocate items d_l $(l \in [i + c_i, MIN(i + c_i + T_j, N)])$ to Channel C_j
    if $(i + c_i + T_j) \geq N$ **then**
      return;
    end if
  end for
  for $(j = K,\ c_i = B;\ j \geq 1;\ j--,\ c_i+ = T_j)$ **do**
    allocate items d_l $(l \in [i + c_i, MIN(i + c_i + T_j, N)])$ to Channel C_j
    if $(i + c_i + T_j) \geq N$ **then**
      return;
    end if
  end for
end for

Secondly, the *permutation* step modifies the initial allocation. It finds the channel C_j having maximal $\frac{A_j}{\sqrt{B_j}}$ and channel C_m having minimal $\frac{A_m}{\sqrt{B_m}}$. By moving the item d_{min} that has the smallest product of access probability and item length from channel C_j to channel C_m, the permutation step improves the performance and hence moves the scheduling towards the optimization. Algorithm 2 describes the detailed code. The high-level allocation will be completed when the permutation is finished. Lastly, the low-level scheduling algorithm will produce the detailed broadcast programs for each channel C_j according to the Log-time algorithm.

In the running example shown in Table 4, it is found that after the initial step, $\frac{A_1}{\sqrt{B_1}} \geq \frac{A_2}{\sqrt{B_2}}$. Therefore, item d_8, as d_{min} in channel C_1, is moved to C_2. Since this movement improves the overall performance, i.e., $MCAED$ is reduced, the permutation step is successful. For the second permutation, the adjustment on the item d_{min} cannot improve the performance, and the permutation is stopped. Finally, the broadcast program of each channel is worked out based on the Log-time algorithm, and the final $MCAED$ is 1.475.

In summary, TOSA has three steps: i) an initialization step to allocate data items to different channels, ii) a permutation step to adjust the allocation to approach the optimal assignment, and iii) a log-time algorithm to determine the broadcast program for each channel. These first two steps are for the high-level allocation of data items into channels, whereas the last step focuses on low-level optimization within a channel. Compared with GREEDY, TOSA achieves a much better performance and it is much closer

Algorithm 2 Permutation

Input: the initial partition of N items;
Output: the approximate partition of these N data items;
Procedure:

while *true* **do**
  find two channels C_j and C_m such that $\frac{A_j}{\sqrt{B_j}} \geq \frac{A_i}{\sqrt{B_i}} \geq \frac{A_m}{\sqrt{B_m}}$, $i, j, m \in [1, K]$
  find d_{min} from C_j such that $\sqrt{p_{min} l_{min}} <= \sqrt{p' l'}$, $\forall d' \in C_j$;
  if $(\frac{A_j^2}{B_j} + \frac{A_m^2}{B_m} > \frac{(A_j - \sqrt{p_{min} l_{min}})^2}{B_j} + \frac{(A_m + \sqrt{p_{min} l_{min}})^2}{B_m})$ **then**
    move item d_{min} from channel C_j to channel C_m;
  else
    return;
  end if
end while

to the optimal value (1.382). Its advantages will be further demonstrated in the next section.

TOSA	
Initialization:	C_1: d_1, d_4, d_5, d_8
	C_2: d_2, d_3, d_6, d_7
Permutation:	
first iteration:	C_1:d_1, d_4, d_5, d_8
	C_2:d_2, d_3, d_6, d_7
search:	$A_1^2/B_1 > A_2^2/B_2$
	$d_{min} = d_8$
Evaluation:	$\frac{A_1^2}{B_1} + \frac{A_2^2}{B_2} > \frac{(A_1 - \sqrt{d_8})^2}{B_1} + \frac{(A_2 + \sqrt{d_8})^2}{B_2}$
Action:	move item d_8 from C_1 to C_2
	continue Permutation step
second iteration:	C_1: d_1, d_4, d_5
	C_2: d_2, d_3, d_6, d_7, d_8
search:	$A_1^2/B_1 > A_2^2/B_2$
	$d_{min} = d_5$
Evaluation:	$\frac{A_1^2}{B_1} + \frac{A_2^2}{B_2} < \frac{(A_1 - \sqrt{d_5})^2}{B_1} + \frac{(A_2 + \sqrt{d_5})^2}{B_2}$
Action:	permutation step is stopped
Log-time algorithm to schedule items in each channel	
	$s_l = \frac{d_j \in C_i \sqrt{p_j}}{\sqrt{p_l}}$, $i \in [1, K]$
	MCAED=1.475

Table 4: The Running Example under TOSA

4. PERFORMANCE EVALUATION

This section describes the simulation model used to evaluate the performance of the proposed TOSA against the existing algorithms, together with the simulation results. The discrete-time simulation package *CSIM* [10] is used to implement the model. In our simulations, the default size of the database is 10000, and the presented result is the average performance of 2 million requests. We assume that the access probabilities of data items follow the Zipf distribution, which can be expressed as follows.

$$p_i = \frac{(1/i)^\theta}{\sum_{i=1}^{N} (1/i)^\theta}, \quad 1 \leq i \leq N$$

Parameter θ is the *access skew coefficient* and N is the database size. The bigger the θ, the more skewed is the distribution of clients' requests. When θ is 0, it is equivalent to uniform distribution. The default value of θ is set at 0.75 in the following simulations, unless otherwise specified.

In addition to TOSA, GREEDY is implemented for comparison purpose. As mentioned before, the major disadvantage of GREEDY is the flat broadcast of the items in each channel. As Log-time can provide optimal scheduling for the single-channel environment given the access probabilities and lengths of the data items, an intuitive solution is to adopt GREEDY for the allocation of items into different channels and employ Log-time to schedule the broadcast of each channel. This intuitive solution, denoted as *COMBI*, is also implemented. However, TOSA is proposed based on the solid theoretical result and hence guarantees a superior performance to COMBI. Simulation results will verify this statement later.

In the rest of this section, we will present the results of the simulation conducted in two different scenarios. The first scenario assumes that each data item has the same length and each channel has the same bandwidth. These are the assumptions made in GREEDY. Therefore, we can compare TOSA with GREEDY and COMBI. In this scenario, it is convenient to adopt the time required to transfer one data item as the unit for the average wait time.

In the second scenario, we evaluate TOSA under variable item lengths and variable channel bandwidths. In the simulations, we use two parameters, namely, $MaxItemLength$ and $MaxBandwidth$ to control the range of the item lengths and channel bandwidths, respectively. Both the item length and available bandwidth follow the uniform distribution between the unit value and the maximum values set forth for them.

Like the simulation conducted in [2], two requests are issued per unit of simulation time. The time to submit requests is uniformly distributed over the unit time interval, and the requested items are determined by the access probability distribution.

4.1 Scenario 1: Unit Item Length and Unit Bandwidth

In this subsection, a set of experiments is conducted assuming that the data item length and channel bandwidth are constant. Three algorithms are compared under various θ values, a various number of channels, and various database sizes. Figures 3 and 4 show their performances under different access distributions with the number of channels K ranging from 2 to 5, and 10,000 items in the database.

It is obvious that GREEDY performs the worst among the three algorithms. Its flat broadcast scheme results in a longer average wait time. As θ increases, the improvement of TOSA and COMBI over GREEDY becomes more significant. Compared to GREEDY, TOSA improves the performance by 18.34% on average, and COMBI increases the performance by 13.03%. Furthermore, TOSA achieves a better performance than COMBI, with an average improvement of 6.46%, which falls within our expectation.

COMBI employs GREEDY and Log-time without considering the dependency between them. However, as we observed from the theoretical analysis in Section 3.2, the inter-channel and intra-channel data allocations are mutually dependent and must be considered together in order to achieve the optimal performance in terms of average wait time. Although TOSA is not guaranteed to achieve the optimal performance, it outperforms COMBI significantly by considering the inter-channel and intra-channel together.

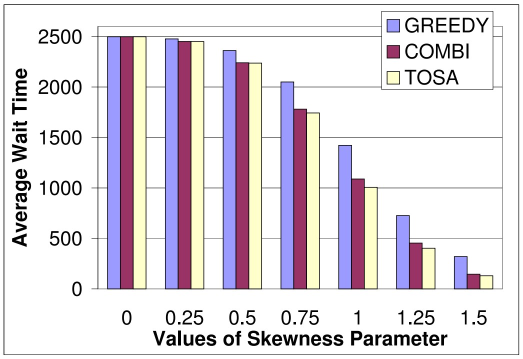


Figure 3: Performance vs. θ ($K = 2, N = 10000$)

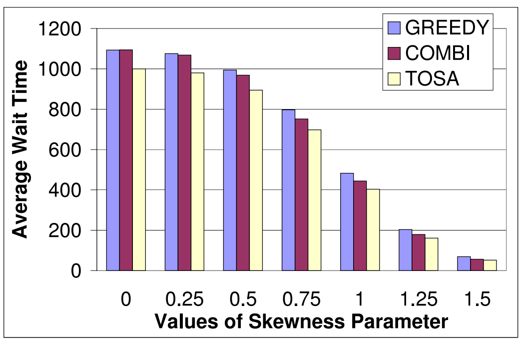


Figure 4: Performance vs. θ ($K = 5, N = 10000$)

In order to provide a complete comparison, two more experiments are conducted with a various number of channels and various database sizes. The simulation results are shown in Figures 5 and 6.

With different numbers of channels, TOSA and COMBI still perform better than GREEDY. However, the degree of improvement decreases as the number of channels increases. The improvement of COMBI against GREEDY drops from 13.2% to 0.5%, when the number of channels increases from 2 to 16. This is because the larger number of channels reduces the differences between the items allocated to the same channel. Therefore, the side effect caused by the flat broadcast of the GREEDY algorithm becomes less. On the other hand, TOSA always surpasses COMBI in performance, with the average improvement around 3.7%. The performances of the three algorithms show a similar behavior under different database sizes.

In conclusion, GREEDY is out-performed by TOSA because it ignores the access probability differences of the data items

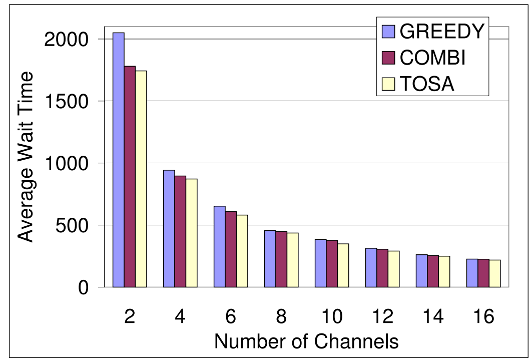


Figure 5: Performance vs. K ($N = 10000, \theta = 0.75$)

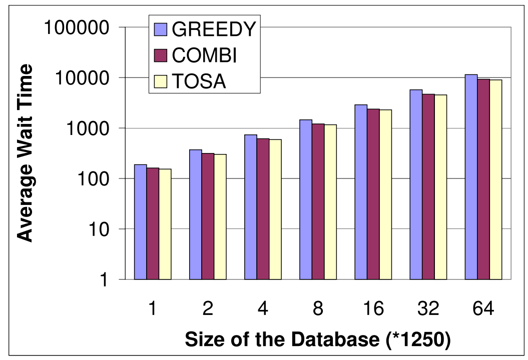


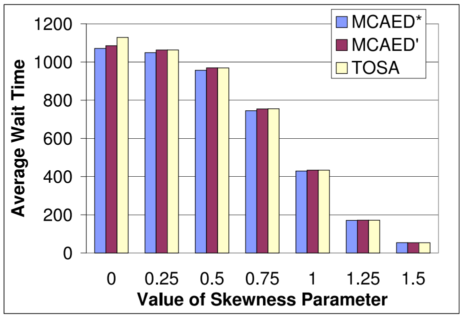
Figure 6: Performance vs. N ($K = 3, \theta = 0.75$)

allocated to the same channel. COMBI improves the performance since the log-time algorithm takes into consideration the different access probabilities of items allocated to one channel. TOSA has the best performance since it is designed based on the conditions for theoretical optimization. The performances of these algorithms are consistent across various numbers of channels and database sizes.

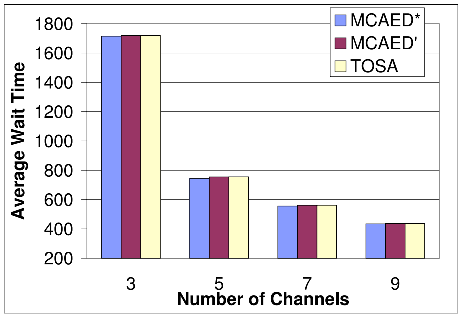
4.2 Scenario 2: Variable Item Length and Variable Bandwidth

Experiments are conducted assuming that items may have different sizes and channels may have different bandwidths. Since GREEDY cannot deal with variable item lengths and bandwidths, it is not used in the comparison. The default values of $MaxItemLength$ and $MaxBandwidth$ are both 5 units. Unless otherwise specified, the default settings are applied. The default number of channels is five.

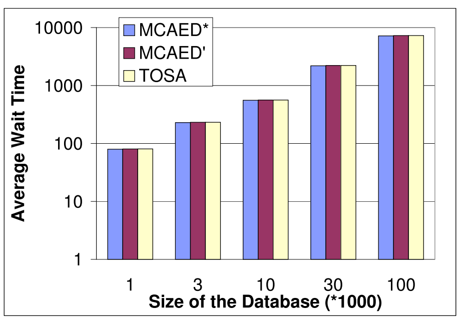
In the following descriptions, three notations are employed to distinguish the performance obtained from different approaches. $MCAED^*$ is the theoretically optimal value of MCAED, which has been defined in Equation 3. $MCAED'$ is the performance obtained by evaluating Equation 2 after applying the TOSA scheduling algorithm. $TOSA$ is the performance measured from the simulations. The comparison between these three values could further verify the accuracy of the simulation. All the performances denoted by TOSA in the previous figures are in fact the real MCAED values from the simulations.



(a) vs. θ ($N = 10000, K = 5$)



(b) vs. K ($N = 10000, \theta = 0.75$)

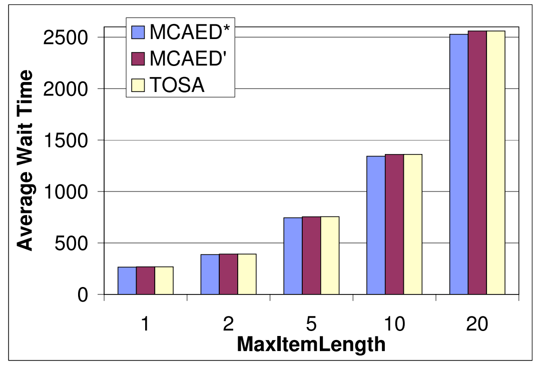


(c) vs. N ($\theta = 0.75, K = 5$)

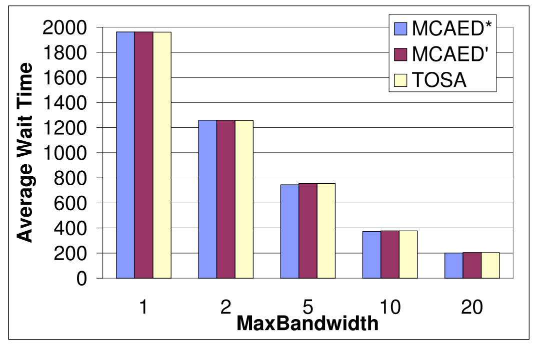
Figure 7: Performance vs. Different Parameters ($MaxItemLength = 5, MaxBandwidth = 5$)

Figure 7(a) presents the performance with different θ values ranging from 0 to 1.5. It is observed that $MCAED'$ approaches to the theoretical optimum, with a difference of around 1%, showing that TOSA is a near-optimal solution to the multi-channel scheduling problem. Secondly, the difference between $MCAED'$ and $MCAED^*$ becomes smaller and smaller as the distribution of access frequencies becomes more and more skewed. This further demonstrates that flat broadcasting cannot provide the optimal performance when the items are not uniformly accessed. Thirdly, $TOSA$ is almost the same as $MCAED'$, which validates the implementation of the simulation.

In the second set of experiments, the performance under a variable number of channels is evaluated. The number of channels varies from three, five, and seven, to nine. As depicted in Figure 7(b), the performance under TOSA again approaches to $MCAED^*$ perfectly. The average difference is only 0.7%. Again, the average value of $MCAED'$ is only 1% worse than that of $MCAED^*$ under different database sized, as shown in Figure 7(c). Therefore, it is safe to conclude that TOSA can achieve the near-optimal performance in multi-channel environments.



(a) vs. $MaxItemLength$ ($MaxBandwidth = 5$)



(b) vs. $MaxBandwidth$ ($MaxItemLength = 5$)

Figure 8: Performance Stability ($N = 10000, \theta = 0.75, K = 5$)

Furthermore, Figures 8(a) and 8(b) show the impact caused by the item length and bandwidth. By fixing the value of $MaxBandwidth$, Figure 8(a) represents the result of varying $MaxItemLength$ from 1, 2, 5, and 10, to 20 units. Similarly, Figure 8(b) depicts the performance under a fixed $MaxItemLength$ and varied $MaxBandwidth$. Consistently, TOSA achieves the near-optimal performance in all cases.

In summary, the performance of broadcast systems could be determined by multiple factors, such as length of items, available bandwidth of channels, and access frequencies of items. The neglect of any of them will impact $MCAED$. Furthermore, both inter-channel and intra-channel scheduling impact the final performance. Like in scenario 1, TOSA improves the waiting time because of the consideration of both aspects. In scenario 2, the strength of TOSA has been further shown. It can achieve the near-optimal performance for the general case where data access frequencies, data sizes, and channel bandwidth can all be non-uniform.

5. CONCLUSION AND FUTURE WORK

This work takes three elements, access frequencies, data sizes, and channel bandwidth, into consideration to schedule the broadcast programs in a muitl-channel environment. The main contributions of this paper are as follows:

1. We derived the optimal value of the average wait time for multiple channels and the condition under which optimality can be achieved.

2. We proposed an approximation algorithm, TOSA, to achieve the near-optimal performance and constructed a set of experiments to evaluate the performance, including the application of two existing algorithms, i.e., GREEDY and Log-time, and the two-level optimization methodology.

GREEDY employs a flat broadcast scheme for data items allocated to the same channel, without considering the difference in access frequency among different items. Our proposed method, TOSA, is based on an efficient, two-level allocation algorithm to first partition the data items over multiple channels and then schedule the data items within each channel. TOSA employs the cost functions developed in our theoretical analysis, which considers non-uniformity in data accesses, channel bandwidths, and data sizes. As such, it achieves a near-optimal performance that closely approximates the optimal performance.

In future research, we plan to extend our work to allow client requests for multiple data items. Furthermore, we will consider situations when uncorrectable errors occur in the broadcast, and how to address the security issue in broadcast environments.

6. ACKNOWLEDGMENTS

Baihua Zheng's work was supported by Wharton-SMU Research Center, Singapore Management University (Grant No. C220/T050011).

7. REFERENCES

[1] D. Barbara and T. Imielinski. Sleepers and workaholics: Caching strategies for mobile environments. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'94)*, pages 1–12, Minneapolis, MN, USA, May 1994.

[2] S. Hameed and N. H. Vaidya. Log-time algorithms for scheduling single and multiple channel data broadcast. In *Proceedings of the 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'97)*, pages 90–99, Budapest, Hungary, September 1997.

[3] G.H. Hardy, J.E. Littlewood, and G. Plya. *Further Remarks on Method: The Inequality of Schwarz.* Cambridge University Press, 1952.

[4] C.H Hsu, G. Lee, and A. Chen. Index and data allocation on multiple broadcast channels considering data access frequencies. In *Proceedings of 3rd International Conference on Mobile Data Management (MDM'02)*, pages 71–78, Singapore, January 2002.

[5] J.-L. Huang and M.-S. Chen. Dependent data broadcasting for unordered queries in a multiple channel mobile environment. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 16(9):1143–1156, September 2004.

[6] A.R. Hurson, Y.C. Chehadeh, and J. Hannan. Object organization on parallel broadcast channels in a global information sharing environment. In *Proceedings of 19th IEEE International Performance, Computing, and Communications Conference (IPCCC'00)*, February 2000.

[7] G. Lee, M.S. Yeh, S.C. Lo, and A. Chen. A strategy for efficient access of multiple data items in mobile environments. In *Proceedings of 3rd International Conference on Mobile Data Management (MDM'02)*, pages 71–78, Singapore, January 2002.

[8] S-C Lo and L.P. Chen. Optimal index and data allocation in multiple broadcast channels. In *Proceedings of the 16th International Conference on Data Engineering (ICDE'00)*, San Diego, CA, USA, February 2000.

[9] K. Prabhakara, K. A. Hua, and J. Oh. Multi-level multi-channel air cache designs for broadcasting in a mobile environment. In *Proceedings of the 16th IEEE International Conference on Data Engineering (ICDE'00)*, pages 167–176, San Diego, CA, USA, February 2000.

[10] H. Schwetman. Mesquite Software, Inc, http://www.mesquite.com, 1998.

[11] N. Shivakumar and S. Venkatasubramanian. Efficient indexing for broadcast based wireless systems. *ACM/Baltzer Mobile Network and Application (MONET)*, 1(4):433–446, December 1996.

[12] N.H. Vaidya and S. Hameed. Data broadcast in asymmetric environments. In *Proceedings of the 1st International Workshop on Satellite-based Information Services (WOSBIS'96)*, Rye, NY, USA, November 1996.

[13] W.G. Yee and S.B. Navathe. Efficient data access to multi-channel broadcast programs. In *Proceedings of International Conference on Information and Knowledge Management, (CIKM'03)*, pages 153–160, New Orleans, Louisiana, USA, November 2003.

[14] W.G. Yee, S.B. Navathe, E. Omiecinski, and C. Jermaine. Efficient data allocation over multiple channels at broadcast servers. *IEEE Transactions on Computers*, 52(10):1231–1236, 2002.