Log-time Algorithms for Scheduling Single and Multiple Channel Data Broadcast*

Sohail Hameed Nitin H. Vaidya
Department of Computer Science
Texas A&M University
College Station, TX 77843-3112, U.S.A.
E-mail: {shameed,vaidya}@cs.tamu.edu
Web: http://www.cs.tamu.edu/faculty/vaidya/

Abstract

With the increasing popularity of portable wireless computers, mechanisms to efficiently transmit information to such *clients* are of significant interest. The environment under consideration is *asymmetric* in that the information server has much more bandwidth available, as compared to the clients. It has been proposed that in such systems the server should broadcast the information periodically. A *broadcast schedule* determines what is broadcast by the server and when.

This paper makes the simple, yet useful, observation that the problem of broadcast scheduling is closely related to the problem of *fair queueing*. Based on this observation, we present a log-time algorithm for scheduling broadcast, based on an existing fair queueing algorithm. This algorithm significantly improves the time-complexity over previously proposed broadcast scheduling algorithms. Also, for environments where different users may be listening to different number of broadcast channels, we present an algorithm to coordinate broadcasts over different channels. Simulation results are presented for proposed algorithms.

Key Words: Data broadcast, asymmetric communication environments, broadcast scheduling, multiple channel broadcast, simulation results.

1 Introduction

Mobile computing and wireless networks are fast-growing technologies that are making ubiquitous computing a reality. Mobile and wireless computing systems have found many applications, including Defense Messaging System (DMS) [24], Digital Battlefield and Data Dissemination (BADD) [5], and as a general-purpose computing tool. With the increasing popularity of portable wireless computers, mechanisms to efficiently transmit information to such *clients* are of significant interest. For instance, such mechanisms could be used by a *satellite* [25] or a *base station* [3] to communicate information of common interest to wireless hosts. In the environment under consideration, the *downstream* communication capacity, from server to clients, is relatively much greater than the *upstream* communication capacity, from clients to server. Such environments are, hence, called *asymmetric* communication environments [3]. In an asymmetric environment, *broadcasting* the information is an effective way of making the information available simultaneously to a large number of users. For asymmetric environment, several researchers have proposed algorithms for designing *broadcast schedules* [1, 2, 3, 4, 8, 9, 11, 10, 12, 14, 16, 15, 14, 17, 18, 30, 31, 32].

We consider a database that is divided into *information items*. The server periodically broadcasts these items to all clients. A broadcast *schedule* determines when each item is transmitted by the server. We present a new approach to design broadcast schedules that attempts to minimize the average "access time". *Access time* is the amount of time a client has to wait for an information item that it needs. It is important to minimize the *access time* so as to decrease the idle time at the client [4, 9, 15, 14, 17, 18, 11, 10, 3, 2, 31, 32].

This paper makes three contributions:

- We observe that the problem of broadcast scheduling is closely related to *packet fair queueing* [6, 19, 21]. While obvious in the hindsight, this observation has not been exploited before to design efficient broadcasting algorithms.
- Based on the above observation, we present a $O(\log M)$ broadcast scheduling algorithm, where M is the number of information items. Simulations show that this algorithm achieves near-optimal performance.
- In environments where different clients may listen to different number of broadcast channels (depending on how many they can afford), the schedules on different broadcast channels should be coordinated so as to minimize the access time for most clients. We extend the above algorithm to such an environment.

*Research reported is supported in part by Texas Advanced Technology Program grant 009741-052-C.

Rest of this paper is organized as follows. Section 2 introduces terminology, and derives some theoretical results that motivate the proposed algorithms. Section 3 compared packet fair queueing and broadcast scheduling. Section 4 presents proposed scheduling algorithm for single channel. Section 5 presents scheduling algorithms for broadcast on two and three channels. Section 6 evaluates the performance of our algorithms. Related work is discussed in Section 7, including a summary of other scheduling algorithms that we have developed. A summary is presented in Section 8.

2 Terminology and Theoretical Foundation [25]

First we introduce some terminology and notations. Database at the server is assumed to be divided into many *information items*. The items are not necessarily of the same length. l_i represents length of item i. The time required to broadcast an item of unit length is referred to as one *time unit*. Hence time required to broadcast an item of length l is l time units. M denotes the total number of information items in the server's database. The items are numbered 1 through M. An appearance of an item in the broadcast is referred to as an *instance* of the item.

The *spacing* between two instances of an item is the time it takes to broadcast information from the beginning of the first instance to the beginning of the second instance. It can be shown that, for optimal broadcast scheduling, all instances of an item should be equally spaced [18, 25]. Hereafter, for our theoretical development, we assume that all instances of item i are spaced s_i apart. The equal-spacing assumption cannot always be realized in practice [28], however, the assumption does provide a basis for developing the proposed algorithms.

Item Mean Access Time of item i, denoted t_i, is defined as the average wait by a client needing item i until it starts receiving item i from the server. We assume that a client is equally likely to need an item at any instant of time. Then, the *average* time until the first instance of item i is transmitted, from the time when a client starts waiting for item i, is $s_i/2$ time units. Hence, $t_i = \frac{s_i}{2}$.

Demand probability of item i, p_i, denotes the probability that an item needed by a client is item i. *Overall Mean Access Time*, denoted $t_{overall}$, is defined as the average wait encountered by a client (averaged over all items). Thus, $t_{overall} = \sum_{i=1}^{M} p_i\, t_i$. Substituting $t_i = \frac{s_i}{2}$, we obtain $t_{overall}$ as

$$t_{overall} = \frac{1}{2} \sum_{i=1}^{M} p_i\, s_i \qquad (1)$$

The theorem below, proved in [25, 28], provides a theoretical basis for the proposed scheduling scheme.

Theorem 1 Square-root Rule [25, 28]: *Assuming that instances of each item i are equally spaced with spacing s_i, minimum* overall mean access time *is achieved when s_i is proportional to $\sqrt{l_i}$ and inversely proportional to $\sqrt{p_i}$. That is,* $s_i \propto \sqrt{\frac{l_i}{p_i}},\ 1 \le i \le M$

Specifically, it can be shown [25, 28] that, optimal s_i is given by

$$s_i = \left(\sum_{j=1}^{M} \sqrt{p_j\, l_j} \right) \sqrt{\frac{l_i}{p_i}} \qquad (2)$$

Substituting this expression for s_i into Equation 1, the optimal *overall mean access time*, named $t_{optimal}$, is obtained as:

$$t_{optimal} = \frac{1}{2} \left(\sum_{i=1}^{M} \sqrt{p_i\, l_i} \right)^2 \qquad (3)$$

$t_{optimal}$ is derived assuming that instances of each item are equally spaced. As noted before, the *equal-spacing* assumption cannot always be realized [28]. Therefore, $t_{optimal}$ represents a *lower bound* on the overall mean access time. The lower bound, in general, is not achievable. However, as shown later, it is often possible to achieve *overall mean access time* almost identical to the above lower bound.

3 Broadcast Scheduling & Packet Fair Queueing

Consider a switch that has many input channels (queues), but just one output channel. *Packet fair queueing* algorithms [6, 21] determine which packet from the many input queues should be transmitted next on the output channel. The main constraint imposed on the packet fair queueing algorithms is that input queue i should get at least fraction ϕ_i of the output channel bandwidth (assuming that the input queue is not empty). Additional constraints may be imposed to assure other "fairness" conditions.

Now consider broadcast scheduling. As noted above, for an optimal schedule, spacing between consecutive instances of item i should be obtained using Equation 2. Equation 2 can be rewritten as

$$\frac{l_i}{s_i} = \frac{l_i}{\left(\sum_{j=1}^{M} \sqrt{p_j\, l_j} \right) \sqrt{\frac{l_i}{p_i}}} \qquad (4)$$

Let ϕ_i denote the right-hand side of Equation 4. That is, $\phi_i = \frac{l_i}{\left(\sum_{j=1}^{M} \sqrt{p_j} l_j \right) \sqrt{\frac{l_i}{p_i}}}$. Then, we have $l_i/s_i = \phi_i$. Thus, the two conditions for obtaining an optimal schedule are:

- $\frac{l_i}{s_i} = \phi_i$ for each item i. Observe that l_i/s_i is the fraction of broadcast channel bandwidth allocated to item i.
- All instances of each item i should be spaced equally apart with spacing s_i.

These two conditions are similar to those imposed on packet fair queueing, particularly in [6]. Although the problem of *packet fair queueing* is not identical to broadcast scheduling, the similarities between these two problems motivated us to adapt a packet fair queueing algorithm in [6, 20] to broadcast scheduling. The broadcast scheduling algorithm, thus obtained, is presented below.

4 Single Channel Broadcast Scheduling Scheme

In this section, we consider the case when the information items are broadcast on a single channel. Section 5 considers multiple channel broadcast.

For each item i, the algorithm maintains two variables, B_i and C_i. B_i is the earliest time when next instance of item i should *begin* transmission, and $C_i = B_i + s_i$. (It may help the reader to interpret C_i as the "suggested worst-case *completion* time" for the next transmission of item i.)

Single Channel Broadcast Scheduling Algorithm

Step 0: Determine optimal spacing s_i for each item i, using Equation 2.
Current Time is denoted by T. Initially, $T = 0$.
Initialize $B_i = 0$ and $C_i = s_i$ for $1 \leq i \leq M$.
Step 1: Determine set S of items for which $B_i \leq T$.
That is, $S = \{i \mid B_i \leq T,\ 1 \leq i \leq M\}$.
(It can be shown that S is never empty.)
Step 2: Let C_{min} = minimum value of C_i over $i \in S$.
Step 3: Choose any one item $j \in S$ such that $C_j = C_{min}$.
Step 4: Broadcast item j at time T.
$B_j = C_j$
$C_j = B_j + s_j$
Step 5: When item j completes transmission, $T = T + l_j$.
Go to step 1.

The algorithm iterates steps 1 through 5 repeatedly, broadcasting one item per iteration. In each iteration, first the set S of items with begin times B_i smaller than or equal to T is determined. The items in set S are "ready" for transmission. From among these items, the items with the smallest C_i (suggested worst-case completion time) is chosen for broadcast.

Using the *heap* data structure, steps 1 through 4 can be implemented such that, the average time complexity per iteration is $O(\log M)$ [6, 28].

As an illustration, assume that the database consists of 3 items, such that $l_1 = 1$, $l_2 = 2$, $l_3 = 3$, $p_1 = 0.5$, $p_2 = 0.25$, and $p_3 = 0.25$. In this case, by Equation 2, $s_1 = 3.224$, $s_2 = 6.448$ and $s_3 = 7.989$. In the first iteration of the above algorithm, at step 2, $B_1 = B_2 = B_3 = T = 0$, and $C_1 = 3.224$, $C_2 = 6.448$ and $C_3 = 7.989$. During the first iteration, $S = \{1, 2, 3\}$, as $T = 0$ and for all items $B_i = 0$. As C_1 is the smallest, item 1 is the first item transmitted. During the second iteration of the algorithm, $T = 1$, $B_1 = 3.224$, $B_2 = B_3 = 0$, $C_1 = 6.448$, $C_2 = 6.448$ and $C_3 = 7.898$. Now, $S = \{2, 3\}$ (as $B_2 = B_3 = 0 < T = 1$, and $B_1 > T$). As $C_2 < C_3$, item 2 is transmitted next. Figure 1 shows the first few items transmitted using the above algorithm. After an initial transient phase, the schedule became cyclic with the cycle being (1,2,1,3).

Simulations show that the above algorithm attempts to use optimal spacing for each item. Simulation results for the above algorithm (Section 6) show that this algorithm performs close to the optimal obtained by Equation 3.

5 Multiple Broadcast Channels

The discussion so far assumed that the server is broadcasting items over a single channel and all the clients are tuned to this channel. One can also conceive an environment in which the server broadcasts information on multiple channels [29, 28], and different clients listen to different number of channels depending on the desired quality of service (as characterized by the mean access time).

To illustrate how the algorithm in Section 4 may be extended for multiple channels, in this paper, we present algorithms for scheduling broadcast on two and three channels. Assume that the broadcast channels are numbered from 1 to c, where c is the number of channels. We assume that a client listening to j channels, $1 \leq j \leq c$, must listen to first j consecutive channels. Thus, a client listening to, say, 2 channels must listen to channels 1 and 2. Let π_j denote the probability that a client listens to j channels. Trivially, $\sum_{j=1}^{c} \pi_j = 1$.

Optimality Criteria

For single channel scheduling, we attempted to minimize overall mean access time, $t_{overall}$. However, with multiple channels, the overall mean access time experienced by clients listening to different number of channels would be different. Let $t_{overall(i)}$ denote the overall mean access time experienced by clients listening to the first i channels. Then, the performance metric of interest here, called *composite* overall mean access time, denoted $t_{composite\_overall}$, is obtained as

$$t_{composite\_overall} = \sum_{i=1}^{c} \pi_i \, t_{overall(i)} \quad (5)$$

This metric is a special case of a metric presented in [29]. When a client listens to only 1 channel, a lower bound on the overall mean access time $t_{overall(1)}$ is given by $t_{optimal}$ in Equation 3. It is easy to see that, a lower bound on $t_{overall(i)}$ is given by $t_{optimal}/i$. Thus, a lower bound on $t_{composite\_overall}$ can be obtained as

$$t_{composite\_optimal} = \sum_{i=1}^{c} \pi_i \, \frac{t_{optimal}}{i} \quad (6)$$

The objective now is to design multi-channel algorithms that minimize $t_{composite\_overall}$.

Staggered Broadcast Schedules

The main idea here is to schedule broadcast of an item i in such a way that its instances on consecutive channels are "staggered" with some interval. As an example, the Figure 2 shows the scheduling of an item i on three channels. The instances on channel 2 are staggered by an interval of ψ_{i2} and those on channel 3 are staggered by an interval of ψ_{i3} with respect to the corresponding instances on channel 1. Note that the spacing between instances of item i on each channel is s_i.

If we assume that every client is listening to all the three channels, i.e., $\pi_3 = 1$, $\pi_1 = \pi_2 = 0$, then clearly $\psi_{i3} =$

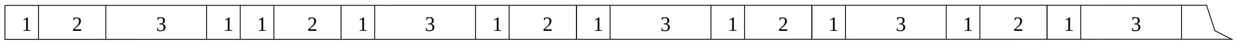


Figure 1: *Illustration of the Single Channel Scheduling Algorithm.*

$2\psi_{i2} = \frac{2}{3}s_i$ would be optimal. With these values, instances of item i are staggered across the three channels such that a client listening to three channels would receive item i every $s_i/3$ time units. In general, however, optimal ψ_{i2} and ψ_{i3} would vary with different π_j distributions.

5.1 2-Channel Scheduling

Let us consider the case when $c = 2$. Hence a client either listens only to channel 1, or to both channels. Appendix A.1 shows that for optimality, $\psi_{i2} = \frac{1}{2}s_i$.

Similar to single channel scheduling, the above proof assumes that the consecutive instances of all items are equally spaced. In addition, the proof also assumes that an instance of item i on channel 2 appears exactly after ψ_{i2} time units from an instance on channel 1. These assumptions may not be realizable in general. However, they provide a theoretical foundation on which algorithms may be developed.

Note that the value of ψ_{i2} is independent of the values of π_1 and π_2. That is every instance of item i on channel 2 should appear exactly midway between every two consecutive instances of item i on channel 1, independent of the values of π_1 and π_2. The following algorithm tries to achieve this result. Similar to the algorithm presented in previous section, for item i, the algorithm below maintains B_i^j and C_i^j, for channel j, $j = 1, 2$.

2-Channel Broadcast Scheduling

Step 0: Determine optimal spacing s_i for each item i, using Equation 2.
Current time is denoted by T. Initially, $T = 0$.
Initialize $B_i^1 = B_i^2 = 0$ and $C_i^1 = C_i^2 = s_i$, $1 \le i \le M$.
Steps below are executed to find an item to transmit on channel h at time T (h may be 1 or 2).
Step 1: Determine set S of items for which $B_i^h \le T$.
That is, $S = \{i \mid B_i^h \le T,\ 1 \le i \le M\}$.
Step 2: Let C_{min} = minimum value of C_i^h over $i \in S$.
Step 3: Choose any one item $j \in S$ such that $C_j = C_{min}$.
Step 4: Broadcast item j at time T.
if $h = 1$ then {
$C_j^2 = T + s_j/2$
$B_j^2 = C_j^2 - s_j/2$ }
$B_j^h = B_j^h + s_j$
$C_j^h = B_j^h + s_j$

5.2 3-Channel Broadcast

Unlike in case of $c = 2$, for three channels ($c = 3$), optimal values of ψ's are dependent on π's. Appendix A.2 shows that for optimality with 3 channels,

$$\psi_{i2} = \frac{2\pi_2 + \pi_3}{4\pi_2 + 3\pi_3} s_i \quad (7)$$

$$\psi_{i3} = \frac{3\pi_2 + 2\pi_3}{4\pi_2 + 3\pi_3} s_i \quad (8)$$

The 2-channel algorithm above can modified for 3 channels, as follows :

3-Channel Broadcast Scheduling

Step 0: Determine optimal spacing s_i for each item i, using Equation 2.
Current time is denoted by T. Initially, $T = 0$.
Initialize $B_i^1 = B_i^2 = B_i^3 = 0$ and $C_i^1 = C_i^2 = C_i^3 = s_i$ for $1 \le i \le M$.
Determine ψ_{ij}, $j = 2, 3$ and $1 \le i \le M$.
Steps below are executed to find an item to broadcast on channel h at time T (h may be 1, 2 or 3).
Step 1: Determine set S of items for which $B_i^h \le T$.
That is, $S = \{i \mid B_i^h \le T,\ 1 \le i \le M\}$.
Step 2: Let C_{min} = minimum value of C_i^h over $i \in S$.
Step 3: Choose any one item $j \in S$ such that $C_j = C_{min}$.
Step 4: Broadcast item j at time T.
if $h = 1$ then {
$C_j^2 = T + \psi_{j2}$
$B_j^2 = C_j^2 - s_j$
$C_j^3 = T + \psi_{j3}$
$B_j^3 = C_j^3 - s_j$ }
else if $h = 2$ then {
$C_j^3 = T + (\psi_{j3} - \psi_{j2})$
$B_j^3 = C_j^3 - s_j$ }
$B_j^h = B_j^h + s_j$
$C_j^h = B_j^h + s_j$

The algorithm can be easily extended for $c > 3$. Section 6 evaluates 2-channel and 3-channel algorithms.

6 Performance Evaluation

In this section, we present simulation results for various algorithms presented above. In each simulation, number of information items M is assumed to be 1000. Each simulation was conducted for at least 8 million item requests by the clients. Other parameters used in the simulation are described below.

We assume that demand probabilities follow the Zipf distribution (similar assumptions are made by other researchers as well [3, 4, 31]). The Zipf distribution may be expressed as :

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

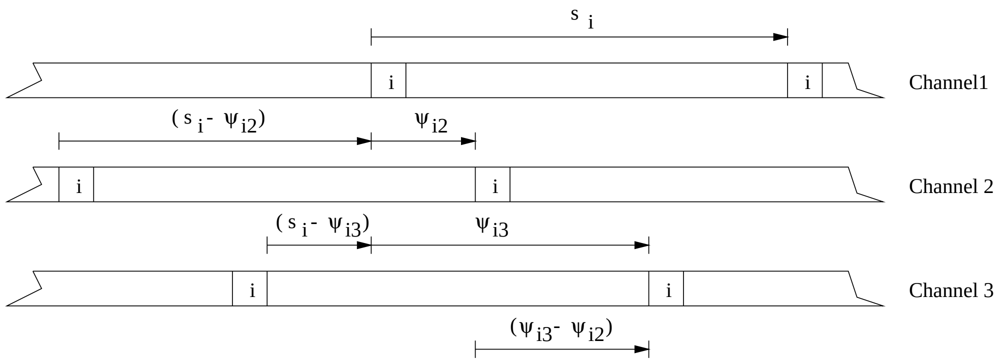


Figure 2: *Schedule for item i on three channels. The instances of item i on channel 2 are staggered by an interval of ψ_{i2} and on channel 3 by an interval of ψ_{i3} with respect to channel 1.*

where θ is a parameter named *access skew coefficient*. Different values of the access skew coefficient θ yield different Zipf distributions. For $\theta = 0$, the Zipf distribution reduces to uniform distribution with $p_i = 1/M$. However, the distribution becomes increasingly "skewed" as θ increases (that is, for larger θ, the range of p_i values becomes larger).

A *length distribution* specifies length l_i of item i as a function of i, and some other parameters. In this paper, we consider the following length distribution.

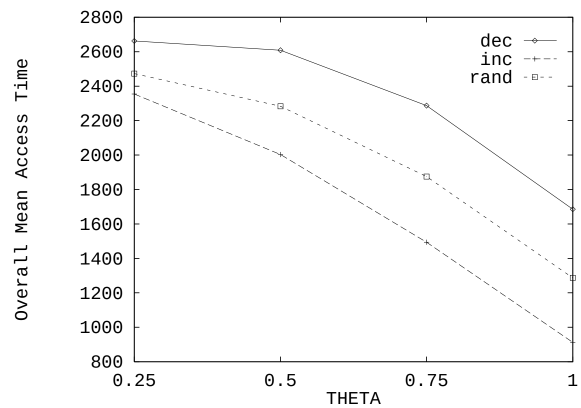
$$l_i = \text{round}\left(\left(\frac{L_1 - L_0}{M-1}\right)(i-1) + L_0\right), \qquad 1 \le i \le M$$

where L_0 and L_1 are parameters that characterize the distribution. L_0 and L_1 are both non-zero integers. `round()` function above returns a rounded integer value of its argument. We consider two special cases of the above length distribution: (i) *Increasing Length Distribution* obtained by $L_0 = 1$ and $L_1 = 10$ and (ii) *Decreasing Length Distribution* obtained by $L_0 = 10$ and $L_1 = 1$. In addition to these length distributions, we also use a *Random* Length Distribution obtained by choosing lengths randomly distributed from 1 to 10 with uniform probability.

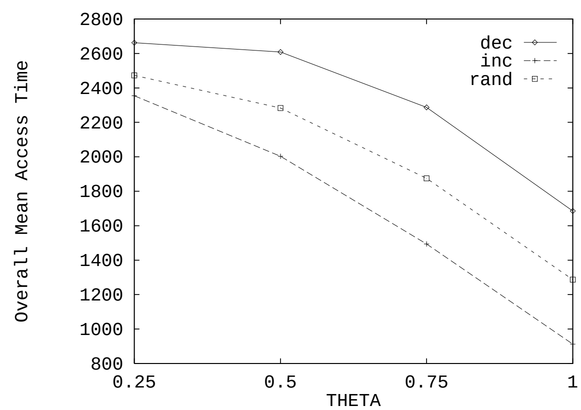
We generated two requests for items per time unit. Simulation time is divided into intervals of unit length; two requests are generated during each such interval. The time at which the requests are made is uniformly distributed over the corresponding unit length interval. The items for which the requests are made are determined using the demand probability distribution.

6.1 Performance Evaluation for Single Channel Broadcast

In this section, we evaluate the Single Channel Scheduling Algorithm explained in Section 4. Figure 3(a) shows the simulation results. It plots *overall mean access time* versus access skew coefficient θ. The curves labeled "dec", "inc" and "rand" respectively correspond to decreasing, increasing and random length distributions defined in Section 6. The corresponding analytical lower bounds obtained from Equation 3 are plotted in Figure 3(b) for comparison.



(a) Simulation results



(b) Analytical lower bounds

Figure 3: Overall mean access time *versus access skew coefficient θ. The simulation curves are obtained using algorithm given in Section 4. The values obtained by simulation are within 0.5% of the corresponding analytical values.*

From the simulation results in Figure 3, observe that the proposed Single Channel Scheduling Algorithm performs very close to optimal (within 0.5% of optimal). These results confirm that the algorithm is able to space instances of each item with approximately ideal spacing, thereby achieving near-optimal *overall mean access time*.

6.2 Performance Evaluation of 2-Channel Broadcast Algorithm

In this section, we evaluate performance of the 2-channel scheduling algorithms in Section 5. Figures 4(a), 4(b) and 5 plot the *overall mean access time* versus access skew coefficient θ for decreasing, increasing and random length distributions respectively. The curves labeled "ch1 sim" and "ch2 sim" are the curves for $t_{overall(1)}$ and $t_{overall(2)}$, respectively, obtained from simulations. Recall that $t_{overall(i)}$ is the overall mean access time experienced by clients listening to first i channels. The curves labeled "ch1 opt" and "ch2 opt" plot $t_{optimal}$ and $t_{optimal}/2$ – recall that $t_{optimal}/i$ is a lower bound on $t_{overall(i)}$ ($t_{optimal}$ is obtained from Equation 3).

The proposed 2-channel algorithm produces same schedule irrespective of the values of π_1 and π_2. Therefore, the above curves in Figures 4(a), 4(b) and 5 are applicable for all π distributions. Observe that, $t_{overall(i)}$ $(i = 1, 2)$ in these curves is very close to $t_{optimal}/i$. Therefore, it follows that the $t_{composite\_overall}$ (for any π distribution) will be very close to $t_{composite\_optimal}$ (see Equations 5 and 6). For brevity, we have not plotted $t_{composite\_overall}$ and $t_{composite\_optimal}$.

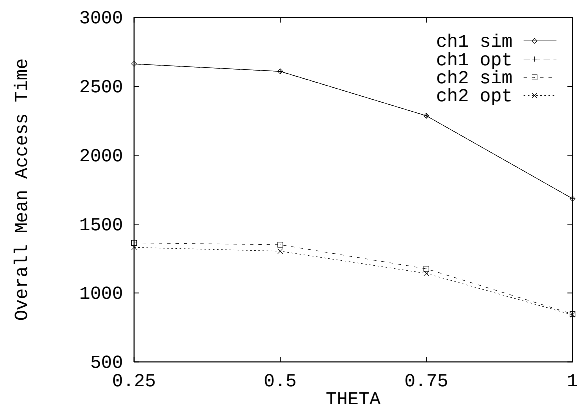
The simulation results above show that the proposed algorithm has near-optimal performance for 2 channels.

6.3 Performance Evaluation of 3-Channel Broadcast Algorithm

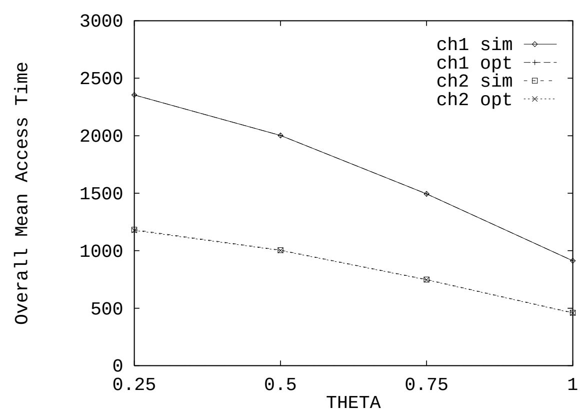
Figures 6(a) and 6(b) show the performance of the 3-channel scheduling algorithm. As noted earlier in Section 5, the values of ψ_{ij}, for $c \geq 3$ depend on π_i's. For $c = 3$, the values of ψ_{i2} and ψ_{i3} as a function of π's are given by Equations 7 and 8.

In each figure in this section, the curves labeled `sim` plot the *composite overall mean access time* $t_{composite\_overall}$ obtained by simulations, and the curves labeled `opt` plot the lower bound $t_{composite\_optimal}$. These curves are plotted for different values of π_3 (horizontal axis) – π_1 and π_2 are defined as functions of π_3 as $\pi_1 = \frac{2}{3}(1 - \pi_3)$ and $\pi_2 = \frac{1}{3}(1 - \pi_3)$. The *Random Length Distribution* is being used in all graphs for 3-channel broadcast.

Figure 6(a) plots the analytical and simulation curves for access skew coefficient, $\theta = 0$ and $\theta = 0.2$, whereas Figure 6(b) plots the analytical and simulation curves for $\theta = 0.5$ and $\theta = 0.75$. In each of these figures, the curves labeled *sim* represent simulation results and those labeled *opt* represent analytical results. The analytical curves plot Equation 6. The figures show that the performance of 3-channel Scheduling Algorithm is fairly close to optimal for some, but not all, values of access skew coefficient θ. The algorithm does not always perform well because of two reasons: (i) the bound $t_{composite\_optimal}$ is not very tight for many values of $c >$



(a) For Decreasing Length



(b) For Increasing Length

Figure 4: Overall mean access time *versus access skew coefficient* θ *for (a) Decreasing Length and (b) Increasing Length Distributions. The simulation results labeled as* sim *are within 3.6% of analytical lower bounds labeled as* opt. *Note that the curves* ch1 sim *and* ch1 opt *are overlapping.*

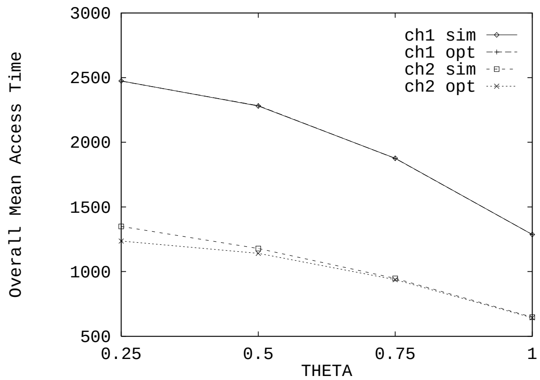


Figure 5: Overall mean access time *versus access skew coefficient* θ *for Random Length Distribution. The simulation results labeled as* sim *are within 9% of analytical lower bounds labeled as* opt*. Note that the curves* ch1 sim *and* ch1 opt *are overlapping.*

2, and (ii) there is still some room for improvement in our algorithm for $c = 3$.

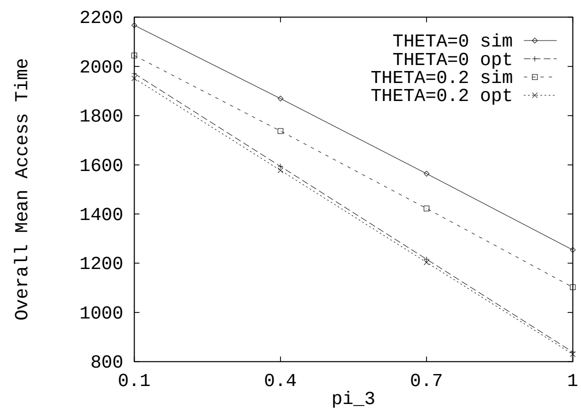
7 Related Work

The algorithms presented in this paper are based on an algorithm proposed previously for "packet fair queueing" [6]. As noted earlier, the problem of optimal broadcast scheduling is closely related to design of good packet fair queueing algorithms.

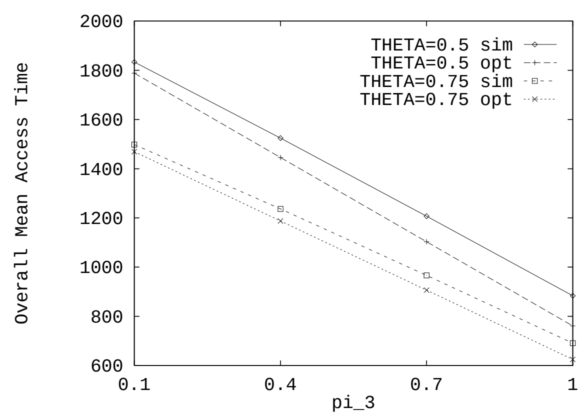
The problem of data broadcasting has received much attention lately. The existing schemes can be roughly divided into two categories (some schemes may actually belong to both categories): Schemes attempting to reduce the *access time* (e.g., [4, 3, 14, 9, 25, 31]) and schemes attempting to reduce the *tuning time*, i.e., the time a client actively listens to the broadcast (e.g., [8, 16, 15, 17, 30]). In this paper, we only consider minimization of access time.

Ammar and Wong [4, 31] have performed extensive research on broadcast scheduling and obtained many interesting results. One of the results obtained by Ammar and Wong is a special case of our square-root rule (Theorem 1). Wong [31] and Imielinski and Viswanathan [14, 30] present a constant-time algorithm that uses a *probabilistic* approach for deciding which item to transmit. The single channel scheduling algorithm presented in this paper results in an improvement by a factor of 2 in the mean access time as compared to the probabilistic algorithm in [14, 30, 31], with a modest increase in time-complexity (logarithmic). Wong also presents a cyclic scheduling algorithm that performs close to the optimal (the schedule needs to be generated *a priori*).

Chiueh [9] and Acharya et al. [3] present schemes that transmit the more frequently used items more often. However, they do not necessarily use optimal broadcast frequencies. Our schemes, on the other hand, tend to use optimal



(a) $\theta = 0, 0.2$



(b) $\theta = 0.5, 0.75$

Figure 6: Composite overall mean access time *versus* π_3*, for Random Length Distribution. The values of* π_1 *and* π_2 *are obtained as* $\pi_1 = 2\pi_2 = \frac{2}{3}(1 - \pi_3)$*. The curves labeled* sim *represent simulation results and* opt *represent analytical results. In (a), the curves shown are for access skew coefficient* $\theta = 0$ *and* $\theta = 0.2$*, whereas in (b), the curves shown are for* $\theta = 0.5$ *and* $\theta = 0.75$*.*

frequencies. (Optimal frequencies are inversely proportional to optimal spacing.)

Gondhalekar et al. [10] have looked at the problem of optimizing mean access time using indexing schemes, and shown that the problem is NP-complete under certain conditions. They also present fast heuristics to achieve a low access time using indexing. The scheduling schemes presented in this paper do not use indexing.

Several researchers, including Su and Tassiulas [23], Acharya et al. [3] and Statathos et al. [22], have considered the possibility of caching information items at the client. With caching, a client need only wait for broadcast if the desired item is not in the cache. Our broadcasting schemes do not consider caching as yet.

We have developed several broadcast scheduling algorithms that are not presented in this paper [13, 27, 28, 29, 25, 26]. This section summarizes some of these algorithms.

- Single channel broadcast: From Theorem 1, it follows that, for an optimal schedule $s_i^2 p_i$ = constant, for all items i. We have developed an $O(M)$ algorithm that attempts to achieve this equality. The simulation results show that this algorithm also results in near-optimal access times [27, 25, 26, 29].

  Su and Tassiulas [23] present a broadcast scheduling scheme for clients that do not have any caches. Although the model used in their work, and the method of arriving at the algorithm are different, it is interesting to note that their algorithm bears resemblance to our $O(M)$ algorithm.

  Based on the $O(M)$ algorithm, we developed another "bucketing" algorithm that can trade time complexity with performance with an appropriate choice of parameter. The bucketing algorithm has some similarities with *broadcast disks* [3], but would typically perform better than broadcast disks [25, 26, 29].

- Multiple channel broadcast: Based on the $s_i^2 p_i$ = constant requirement, we have developed a $O(cM)$ multi-channel broadcast scheduling algorithm for c channels [13, 29]. This algorithm results in near-optimal performance in many cases, and tends to perform better than the multi-channel algorithm presented in this paper. However, the algorithm presented here has lower time complexity.

  In the multi-channel algorithm presented in this paper, each item is *independently* staggered on the multiple channels. Another possible approach is to design a single schedule for one channel, and use staggered (time-shifted) versions of the entire schedule on other channels [26]. We are further investigating this approach at present.

- Broadcast in presence of transmission errors: We have shown that, if probability that an item of length l contains an uncorrectable (detected) error is $E(l)$, then the proportionality in Theorem 1 must be modified as

  $$s_i \propto \sqrt{\frac{l_i}{p_i}\left(\frac{1-E(l_i)}{1+E(l_i)}\right)}$$

  The modified Theorem 1 can then be used to design scheduling algorithms (similar to above algorithms) in presence of transmission errors [13, 25, 29].

8 Conclusions

This paper considers *asymmetric* environments where a server has a much larger communication bandwidth available as compared to the clients. In such an environment, an effective way for the server to communicate information to the clients is to broadcast the information periodically. This paper makes three contributions:

- Observes that broadcast scheduling problem is similar to packet fair queueing.
- Presents a broadcast scheduling algorithm based on a packet fair queueing algorithm.
- Presents algorithms for scheduling broadcasts on multiple channels.

Simulation results suggest that proposed algorithms perform well. Future work includes derivation of a better bound for $t_{composite\_overall}$, particularly, for $c \geq 3$. We believe that the bound $t_{composite\_optimal}$ is quite loose when $c \geq 3$.

This paper does not consider caching of information at a client, or the possibility of combining data broadcast (*push*) with on-demand (*pull*) delivery. These issues are a subject of our on-going work.

Acknowledgements

Thanks are due to P. Krishna for drawing our attention to the papers on packet fair queueing [20]. This work was motivated by discussions with him on the possibility of applying our previous broadcast scheduling algorithms to solve the packet fair queueing problem. This paper presents the converse, application of packet fair queueing algorithms to solve the broadcast scheduling problem.

A Appendix: Optimal Values of Stagger

A.1 Two Channel Broadcast

Figure 7 shows different instances of item i scheduled on two channels. The spacing on each of the channels is s_i. Every instance on channel 2 is staggered by an interval of ψ_{i2} from the corresponding instance on channel 1. Our interest is to determine the value of ψ_{i2} which will result in optimal *composite item mean access time*, denoted t_i, as follows. Note that each composite t_i is being optimized independently – thus, all optimal t_i (or optimal stagger for all items) may not be achievable simultaneously.

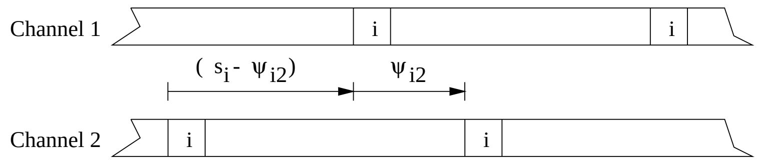


Figure 7: *Schedule for item i on two channels. The instances of item i on channel 2 are staggered from channel 1 by an interval of ψ_{i2}. The value of ψ_{i2} should be $\frac{1}{2}s_i$ for the mean access time for item i to be minimum.*

The *item mean access time*, t_{i1}, for a client listening to channel 1, assuming that a request is equally likely to occur at any time in interval s_i, is clearly

$$t_{i1} = \frac{1}{2}s_i \tag{9}$$

Note that the probability that a client makes a request during a sub-interval of length τ of an interval of length s_i is given by τ/s_i. Therefore, *item mean access time*, t_{i2}, for a client listening to both the channels can be obtained as

$$t_{i2} = \frac{1}{2}\frac{(s_i - \psi_{i2})^2}{s_i} + \frac{1}{2}\frac{\psi_{i2}^2}{s_i} \tag{10}$$

Thus, the composite item mean access time can be obtained as

$$t_i = \pi_1\, t_{i1} + \pi_2\, t_{i2} \tag{11}$$

$$= \frac{1}{2}\pi_1\, s_i + \frac{1}{2}\pi_2\frac{(s_i - \psi_{i2})^2}{s_i} + \frac{1}{2}\pi_2\frac{\psi_{i2}^2}{s_i} \tag{12}$$

For minimum value of t_i, we differentiate Equation 12 with respect to ψ_{i2} and equate it to zero:

$$\frac{dt_i}{d\psi_{i2}} = -\pi_2\frac{(s_i - \psi_{i2})}{s_i} + \pi_2\frac{\psi_{i2}}{s_i} = 0.$$

Solving for ψ_{i2}, we get $\psi_{i2} = \frac{1}{2}s_i$.

Note that the value of ψ_{i2} for optimal *composite item mean access time* is independent of π_1 and π_2 for two channel case. However, as can be seen in the next section, for $c = 3$, value of ψ_{i2} for optimal *composite item mean access time* is a function of π_j's.

A.2 Three Channels Case

Figure 2 shows the schedule for item i on three channels. Let the instances of item i on channel 2 be staggered by an interval of ψ_{i2} and on channel 3 be staggered by an interval of ψ_{i3} with respect to channel 1. A client may listen to channel 1 only, or to channels 1 and 2, or to all the three channels. The *item mean access time* for item i for a client listening to channel 1 and for a client listening to channel 1 and 2, denoted by t_{i1} and t_{i2}, and given by Equations 9 and 10 respectively are still valid, as the scheduling on first two channels in Figure 2 is similar to the scheduling shown in Figure 7. However, the *item mean access time* for item i for the client listening to all the three channels, denoted by t_{i3}, is given by

$$t_{i3} = \frac{1}{2}\frac{(s_i - \psi_{i3})^2}{s_i} + \frac{1}{2}\frac{\psi_{i2}^2}{s_i} + \frac{1}{2}\frac{(\psi_{i3} - \psi_{i2})^2}{s_i} \tag{13}$$

From Equations 9, 10, 13, we get

$$\begin{aligned} t_i &= \pi_1\, t_{i1} + \pi_2\, t_{i2} + \pi_3\, t_{i3} \\ &= \frac{1}{2}\pi_1 s_i + \frac{1}{2}\pi_2\frac{(s_i - \psi_{i2})^2}{s_i} + \frac{1}{2}\pi_2\frac{\psi_{i2}^2}{s_i} \\ &\quad + \frac{1}{2}\pi_3\frac{(s_i - \psi_{i3})^2}{s_i} + \frac{1}{2}\pi_3\frac{\psi_{i2}^2}{s_i} + \frac{1}{2}\pi_3\frac{(\psi_{i3} - \psi_{i2})^2}{s_i} \end{aligned}$$

Again, for optimal t_i, differentiating the above equation with respect to ψ_{i2} and $\psi i3$, we get

$$\begin{aligned} \frac{\partial t_i}{\partial \psi_{i2}} &= -\pi_2\frac{(s_i - \psi_{i2})}{s_i} + \pi_2\frac{\psi_{i2}}{s_i} + \pi_3\frac{\psi_{i2}}{s_i} \\ &\quad -\pi_3\frac{(\psi_{i3} - \psi_{i2})}{s_i} \\ &= 0 \end{aligned} \tag{14}$$

$$\frac{\partial t_i}{\partial \psi_{i3}} = -\pi_3\frac{(s_i - \psi_{i3})}{s_i} + \pi_3\frac{(\psi_{i3} - \psi_{i2})}{s_i} = 0 \tag{15}$$

We assume that π_2 and π_3 are not both 0 – if both are 0, then the 3-channel problem reduces to the single channel broadcast problem. Solving Equations 14 and 15, we get $\psi_{i2} = \frac{2\pi_2+\pi_3}{4\pi_2+3\pi_3}s_i$ and $\psi_{i3} = \frac{3\pi_2+2\pi_3}{4\pi_2+3\pi_3}s_i$. It can be verified that these values of ψ_{i2} and ψ_{i3} represent the point of minima, by applying appropriate checks to second derivatives of t_i [7].

The above proof can be generalized for $c > 3$ also.

References

[1] S. Acharya, M. Franklin, and S. Zdonik, “Prefetching from a broadcast disk,” in *12th International Conference on Data Engineering*, February 1996.

[2] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, “Broadcast disks - data management for asymmetric communications environment,” in *ACM SIGMOD Conference*, May 1995.

[3] S. Acharya, M. Franklin, and S. Zdonik, “Dissemination-based data delivery using broadcast disks,” *IEEE Personal Communication*, pp. 50–60, December 1995.

[4] M. H. Ammar and J. W. Wong, “On the optimality of cyclic transmission in teletext systems,” *IEEE Transactions on Communications*, pp. 68–73, January 1987.

[5] Battlefield awareness and data dissemination (BADD) program, program duration 1996-2000. Web site at http://maco.dc.isx.com/iso/battle/badd.html.

[6] J. C. R. Bennett and H. Zhang, "Wf2q: Worst-case fair weighted fair queueing," in *INFOCOM'96*, March 1996.

[7] W. E. Boyce and R. C. DiPrima, *Calculus*. John Wiley & Sons, Inc., 1988.

[8] M.-S. Chen, P. S. Yu, and K.-L. Wu, "Indexed sequential data broadcasting in wireless mobile computing," in *International Conf. Distributed Computing Systems*, pp. 124–131, 1997.

[9] T. Chiueh, "Scheduling for broadcast-based file systems," in *MOBIDATA Workshop*, November 1994.

[10] V. Gondhalekar, R. Jain, and J. Werth, "Scheduling on airdisks: Efficient access to personalized information services via periodic wireless data broadcast," in *IEEE Int. Conf. Comm.*, June 1997.

[11] V. A. Gondhalekar, "Scheduling periodic wireless data broadcast," December 1995. M.S. Thesis, The University of Texas at Austin.

[12] A. Gurijala and U. Pooch, "Propagating updates in asymmetric channels (a position paper)," in *First International Workshop on Satellite-based Information Services (WOSBIS)*, November 1996.

[13] S. Hameed, "Scheduling information broadcast in asymmetric environment." M.S. Thesis, Dept. of Computer Science, Texas A&M University, May 1997.

[14] T. Imielinski and S. Viswanathan, "Adaptive wireless information systems," in *Proceedings of SIGDBS (Special Interest Group in DataBase Systems) Conference*, October 1994.

[15] T. Imielinski, S. Viswanathan, and B. R. Badrinath, "Energy efficient indexing on air," May 1994.

[16] T. Imielinski, S. Viswanathan, and B. R. Badrinath, "Power efficient filtering of data on air," in *4th International Conference on Extending Database Technology*, March 1994.

[17] T. Imielinski, S. Viswanathan, and B. R. Badrinath, "Data on the air - organization and access," *IEEE Transactions of Data and Knowldge Engineering*, July 1996.

[18] R. Jain and J. Werth, "Airdisks and airraid : Modelling and scheduling periodic wireless data broadcast (extended abstract)," Tech. Rep. DIMACS Tech. Report 95-11, Rutgers University, May 1995.

[19] S. Keshav, "On the efficient implementation of fair queueing," *Journal of Internetworking: Research and Experience*, vol. 2, September 1991.

[20] P. Krishna, "personal communication on packet fair queueing and broadcast scheduling," 1996.

[21] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round robin," in *SIGCOMM'95, Cambridge, MA, USA*, 1995.

[22] K. Stathatos, N. Roussopoulos, and J. S. Baras, "Adaptive data broadcasting using air-cache," in *First International Workshop on Satellite-based Information Services (WOSBIS)*, November 1996.

[23] C.-J. Su and L. Tassiulas, "Novel information distribution methods to massive mobile user populations," Tech. Rep. TR 97-46, ISR, Univ. of Maryland, 1997.

[24] U.S. Navy, "Chips articles on DMS," 1993-96. Web site at http://www.chips.navy.mil/dms/dmsart.html.

[25] N. H. Vaidya and S. Hameed, "Data broadcast in asymmetric environments," in *First International Workshop on Satellite-based Information Services (WOSBIS)*, November 1996.

[26] N. H. Vaidya and S. Hameed, "Data broadcast scheduling: On-line and off-line algorithms," Tech. Rep. 96-017, Computer Science Department, Texas A&M University, College Station, July 1996.

[27] N. H. Vaidya and S. Hameed, "Data broadcast scheduling (part I)," Tech. Rep. 96-012, Computer Science Department, Texas A&M University, College Station, May 1996.

[28] N. H. Vaidya and S. Hameed, "Improved algorithms for scheduling data broadcast," Tech. Rep. 96-029, Computer Science Department, Texas A&M University, College Station, December 1996.

[29] N. H. Vaidya and S. Hameed, "Scheduling data broadcast in asymmetric communication environments," Tech. Rep. 96-022, Computer Science Department, Texas A&M University, College Station, November 1996.

[30] S. Viswanathan, *Publishing in Wireless and Wireline Environments*. PhD thesis, Rutgers, November 1994.

[31] J. W. Wong, "Broadcast delivery," in *Proceedings of IEEE*, pp. 1566–1577, December 1988.

[32] Z. Zdonik, R. Alonso, M. Franklin, and S. Acharya, "Are disks in the air, ' just pie in the sky? '," in *IEEE Workshop on Mobile comp. System*, December 1994.