# Indexing Techniques for Wireless Data Broadcast under Data Clustering and Scheduling

**Qinglong Hu***
Dept of Computer Science
Univ. of Science and Technology
Clear Water Bay, Hong Kong
qinglong@cs.ust.hk

**Wang-Chien Lee**
GTE Laboratories Inc.
40 Sylvan Road
Waltham, MA 02451
wlee@gte.com

**Dik Lun Lee**
Dept of Computer Science
Univ. of Science and Technology
Clear Water Bay, Hong Kong
dlee@cs.ust.hk

**Abstract**

This paper investigates power conserving indexing techniques for data disseminated on a broadcast channel. A *hybrid indexing* method combining strengths of the *signature* and the *index tree* techniques is presented. Different from previous studies, our research takes into consideration two important data organization factors, namely, *clustering* and *scheduling*. Cost models for index, signature and hybrid methods are derived by taking into account various data organizations accommodating these two factors. Based on our analytical comparisons, the signature and the hybrid indexing techniques are the best choices for power conserving indexing of various data organizations on wireless broadcast channels.

## 1 Introduction

Wireless broadcasting is an attractive approach for data dissemination in a mobile environment since it tackles both of the bandwidth and power problems [7, 10]. On one hand, data disseminated through broadcast channels allows simultaneous access by an arbitrary number of mobile users and thus allows efficient usage of scarce bandwidth. On the other hand, the mobile computers consume less battery power when passively monitoring broadcast channels than actively interacting with the server by point-to-point communication.

To facilitate efficient data delivery on broadcast channels, *scheduling* and *clustering* are frequently used to select and organize data for broadcast. Broadcast scheduling policies determine the content and organization of data broadcasting based on aggregate user data access patterns [1, 2, 3, 11]. Clustering for data broadcast is to place all frames with the same value of a specific attribute[1] consecutively in a broadcast cycle. Clustering allows continuous reception of data with a specific attribute value.

Power conserving indexing techniques for broadcast channels, with the basic idea of automizing selective tuning, has been actively discussed in the literature [4, 7, 8, 9] in the recent years. Among them, signature and index tree techniques represent two different classes of broadcast indexing methods for power conservation [7, 8]. The basic idea is that, by receiving auxiliary information about the arrival time of the forthcoming data on broadcast channels, a mobile computer may stay with power saving mode and only selectively tune in the broadcast channels when the data of interest actually arrives. However, these early proposals didn't take the factors of clustering and scheduling into consideration. In this paper, we re-examine the index tree and signature methods by taking the clustering factor and two well known broadcast scheduling policies, namely *broadcast disks* and *flat broadcast* [1], into consideration. Moreover, we revised the index and signature methods for improvement where is not appropriate, e.g., we separate the index frames from data frames for the index method to accurately estimate the index overhead.

In order to obtain an inside of these two index methods, we developed cost formulae based on *tune-in time*[2] and *access time*[3] to cover the following organizations on broadcast channels: : i) flat scheduling and clustered; ii) flat scheduling and non-clustered; iii) broadcast disk scheduling and clustered. To our knowledge, this is the first systematical study on power conserving indexing techniques which takes both clustering and scheduling issues into account. Through the analysis and under-

*The author is now with Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada

[1] In this paper, we only consider the case of single attribute indexing and clustering. Issues involving multiple attribute indexing and clustering are addressed in [6].

[2] The period of time spent by a mobile computer staying active in order to obtain the requested data.

[3] The average time elapsed from the moment a query is issued to the moment when all the requested data frames are received.

standing of these two index methods and derivations of their cost models, we come up naturally with a new power conserving index, called *hybrid index*.

Performance evaluation on index tree, signature and hybrid index have been conducted based on the formulae we developed. While access time measures the efficiency of access methods and data organizations for broadcast channels, tune-in time is frequently used to estimate the power consumption by a mobile computer. A good power conserving indexing technique has to balance out the index overhead (in terms of access time increased) and the tune-in time saved.

The rest of the paper is organized as follows. Section 2 gives an informal introduction of the broadcast channels, indexing techniques and system parameters used in performance evaluation and comparisons. In Section 3, indexing techniques based on index tree and signature methods are re-examined by taking the clustering and scheduling factors into consideration. Section 4 proposes the new hybrid index scheme and develop corresponding cost models. In Section 5, performance evaluation of the indexing techniques, in terms of tune-in time and access time are presented. Finally, Section 6 concludes the paper.

## 2 The Data Organization for Wireless Broadcast

We assume that a base station is serving the role of an information server and periodically broadcasts, on a specific channel, data of popular demands as a series of *data frames* to a large client population. The clients retrieve the frames of their interest off the air by monitoring the broadcast channel. In addition, index frames, which contain indexing information such as index tree nodes or signatures, can be inserted into a broadcast cycle to facilitate selective tuning. Those two types of frames are usually interleaved for broadcast. Since these frames is periodically broadcast, a complete broadcast of these frames is called a *broadcast cycle*.

Data organizations on broadcast channels have great impacts on data access efficiency and power consumption. Data frames can be clustered based on attributes. Based on the results in [7], index tree techniques result in more efficient access for clustered information than non-clustered one. In this paper, clustered and non-clustered data organizations are investigated for the other two index methods, the signature and the hybrid methods. Generally speaking, a non-clustered data organization can be divided into a number of segments, called *meta segments* [7], with non-decreasing or non-increasing values of a specific attribute. These meta segments can be considered as clustered and thus the indexing techniques for clustered data can be applied to them. To facilitate our study, we use the *scattering factor* $M$, the number of meta segments in the data organization, to model the non-clusterness of a data organization[4].

Due to some timely events, the client access pattern sometimes shows skewed distributions. In this case, scheduling data frames in broadcast disks (refer to [1] for detail) can achieve a better performance in terms of the access time. Therefore, the application of the index methods to broadcast disks are also addressed in this paper.

The broadcast disk method has better access time when the data frames with the same attribute values are clustered in one of the minor cycles. By receiving the cluster of data frames together, the mobile computer can answer the query without continuing to monitor the rest of broadcast cycle. This can be achieved by placing all of the data frames with the same indexed attribute value as a cluster on the same broadcast disk. The whole cluster of data frames are brought to the broadcast channel as a unit. Depending on speed of broadcast disk where this cluster is located, these data frames may appear several times in minor cycles. Thus, the resulting broadcast cycle is different from the completely clustered broadcast cycle. For broadcast scheduling adopting broadcast disks without using clustering, we simply consider the resulting broadcast cycle as non-clustered. In that case, broadcast disks loses its advantages compared with flat broadcast. Thus, when we consider index techniques for broadcast disks in the later sections of this paper, we only consider clustered case. We assume only one broadcast channel since a channel with large bandwidth is logically the same as multiple channels with combined bandwidth of the same capacity. Moreover, it reduces overheads of administrating multiple channels. With the same token, we assume that index information is disseminated on the same broadcast channel. Finally, we assume that updates are only reflected between cycles. In other words, a broadcast schedule is determined before a cycle begins.

Table 1: General System Parameters

| | |
|---|---|
| $D$ | Number of data frames in a cycle |
| $F$ | Number of distinct data frames in a cycle |
| $P$ | Average number of packets in a data frame |
| $S$ | Selectivity of query: average number of frames containing the same attribute value |
| $M$ | Scattering factor of an attribute, which is the number of meta segments of the attribute |

Table 1 gives the general parameters which describe the characteristics of a broadcast cycle. The cost models for the various index methods discussed later in this paper are derived based on these parameters. Although the sizes of data frames may vary, we assume frames to be a multiple of the packet size. Both access time and tune-in time are measured in terms of number of packets.

[4] To simplify our discussion, we neglect the variance of the meta segment size.

## 3 Classical Indexing Techniques for Broadcast

In this section we describe the distributed indexing and integrated signature techniques because they are the best methods of their class for single attribute indexing. The analytical cost models for the access time and the tune-in time of clustered and non-clustered data broadcast are presented. Moreover, the application of these index techniques to broadcast disks is also considered. Due to space constraints, we do not give all the derivations of these cost models. Interested readers may refer to [5] for more details.

### 3.1 The Index Tree Techniques

As with a traditional disk-based environment, index-tree methods [7] have been applied to data broadcasts on wireless channels. Instead of storing the locations of disk records, the arrival time of the clustered data frames is kept in the leaves of an index tree.

To reduce access time while maintaining a similar tune-in time for the client, the index tree can be replicated and interleaved with the information. *Distributed indexing* is actually an index replication and interleaving method. By dividing a broadcast cycle into $d$ data blocks, an index tree can be broadcasted every $\frac{1}{d}$ of the broadcast cycle. However instead of replicating the entire index tree $d$ times, distributed index only interleaves the part of the index tree which indexes the immediately followed data blocks. The whole index tree is divided into two parts: replicated and non-replicated parts.

Table 2: Parameters for Index Tree

| | |
|---|---|
| $h$ | Height of the whole index tree |
| $t$ | Height of the replicated tree part |
| $T$ | Number of packets in an index tree node |
| $n$ | Number of search keys plus pointers a node holds |

Table 2 lists the parameters for the index tree cost model. The replicated part consists of the upper $t$ levels of the index tree and each node in that part is replicated the number of times as the number of its children, while the non-replicated part consists of the lower $(h-t)$ levels and each node in this part appears only once in a given broadcast cycle. Since the lower levels of an index tree take up much more space than the replicated part, the index overheads can be greatly reduced if the lower levels of the index tree are not replicated. In this way, access time can be improved significantly without much deterioration in tune-in time.

**Flat Broadcast:** Let us derive the access time and the tune-in time estimation for flat broadcast first. With this scheduling policy, each frame is broadcast exactly once in a cycle. Thus, the number of data frames in the broadcast, $D$, is equal to the number of distinct frames $F$. The initial probe period is the time to reach the index frame at the beginning of the next data block and can be estimated as:

$$P_F^T = \frac{((X[h]+L[t+1]-1)\cdot M\cdot T+F\cdot P}{2\cdot M\cdot L[t+1]}$$

For a **clustered** broadcast, the scattering factor, $M = 1$ and the expected number of data frames before the arrival of the desired frames is $F/2$. The access time is:

$$A_{F,C}^T = P_F^T+(X[h]+L[t+1]-1)\cdot T/F+P)\cdot F/2 \\ +S\cdot P$$

Hereafter $A$ and $TN$ denote the access time and tune time. For a **non-clustered** broadcast cycle ($M > 1$),

$$A_{F,NC}^T = P_F^T+(X[h]+L[t+1]-1)\cdot M\cdot T+F\cdot P$$

The tune-in time for both clustered and non-clustered broadcast cycle depends on the initial probe, the scanning of index tree, the extra scanning of index tree in subsequent meta segments, and the retrieval of $S$ data frames. Thus, the tune-in time of the index technique is upper bounded by:

$$TN_F^T = (h+1)\cdot T+(M+S)\cdot P$$

**Broadcast Disks:** Assuming a $N$ broadcast disks system where $F_i$ and $f_i$ denote the size and broadcast frequency of disk $i$, we next derive the cost formulae for broadcast disks. Since data frames with the same attribute values are clustered in one minor cycle, we can treat each minor cycle of the broadcast disks as a meta segment[5], An index tree can be built for each minor cycle. Similar to flat broadcast, the initial probe period, the time to reach the index frame at the beginning of the next data block, can be estimated as:

$$P_B^T = \frac{((X[h]+L[t+1]-1)\cdot M\cdot T+D\cdot P}{2\cdot M\cdot L[t+1]}$$

where the number of data frames in the broadcast is $D=\sum_{i=1}^{N}F_i\cdot f_i$ and the scattering factor, $M$, is equal to the number of minor cycles in the broadcast (i.e., the LCM of the relative frequencies of the disks). Hence, the access time for retrieving a data frame from broadcast disk $i$ is:

$$A_B^T = P_B^T+((X[h]+L[t+1]-1)\cdot M\cdot T/D+P) \\ \cdot D/(2f_i)+S\cdot P$$

Since all the desired data frames are clustered in one minor cycle, the tune-in time is the same as in flat broadcast for a clustered broadcast cycle, i.e.,

$$TN_B^T = (h+1)\cdot T+(1+S)\cdot P$$

[5]Note that it is different from the meta segments for a non-clustered flat broadcast, where frames with the same attribute value may be scattered in several meta segments.

### 3.2 The Signature Technique

Signature methods have been widely used for information retrieval. A signature of a data frame is basically a bit vector generated by first hashing the values in the data frame into bit strings and then superimposing them together [8]. They are broadcast together with the data frames in every broadcast cycle. A query signature is generated in a similar way based on the query specified by the user. To answer a query, the client simply retrieves information signatures from the broadcast channel and then matches the signatures with the query signature by performing a bitwise *AND* operation. When the result is not the same as the query signature, the corresponding data frame can be ignored. Otherwise, there are two possible cases. First, for every bit set in the query signature, the corresponding bit in the data frame signature is also set. This case is called *true match*. Second, the data frame in fact does not match the search criteria. This case is called *false drop*. Obviously the data frames still need to be checked against the query to distinguish a true match from a false drop.

The signature technique interleaves signatures with their associated data frames. By checking a signature, the mobile computer can decide whether a data frame contains the desired information. If it doesn't, the mobile computer turns into doze mode and wakes up again for the next signature. The primary issue with different signature methods is the size and the number of levels of the signatures.

Table 3: Parameters for Signature Scheme

| | |
|---|---|
| $P_f$ | false drop probability for integrated signatures |
| $k$ | number of data frames indexed by a signature |
| $p$ | number of bits in a packet |
| $R$ | the size (number of packets) of a signature |

The number and the size of the signatures and thus the false drop probability of the signatures[6] affect tune-in time and access time. The false drop probability may be controlled by the size of the signatures. The initial probe time is related to the number of signatures interleaved with the data frames. Table 3 defines the parameters for signature cost models. Estimation of the false drop probability is given in the following Lemma [8]:

**LEMMA 1 (optimal false drop probability)** *Given the number of packets in a signature, $R$, the number of bit strings superimposed into the signature, $s$, the false drop probability for the signature is:* $P_f = 2^{-R\cdot(p\cdot ln2)/s}$

The integrated signature method is constructed for a group of consecutive frames and is general enough to accommodate both clustered and non-clustered data broadcast. For clustered data broadcast, a lot of data frames can be indexed by one integrated signature. According to Lemma 1, the smaller the number of bit strings $s$ superimposed into an integrated signature, the lower the false drop probability. The integrated signature generated for a clustered broadcast cycle has the effect of reducing the number of bit strings superimposed. To maintain a similar false drop probability for a non-clustered broadcast cycle, the number of data frames indexed by an integrated signature may be reduced. How to determine the number of data frames for signature generation needs further study.

**Flat Broadcast:** Next, we give the access time and the tune-in time for clustered and non-clustered broadcast cycles [5]. Let $SIG$ be the average signature overhead for each data frame. Then, $SIG = R/k$.

For **clustered** data broadcast, the access time can be derived as follows:

$$A^S_{F,C} = (R + k\cdot P)/2 + (SIG + P)\cdot F/2 + S\cdot P$$

and the tune-in time is:

$$TN^S_{F,C} = P/2 + P\cdot(S + k/2) + (R/k + P_f\cdot P)\cdot F/2$$

For a **non-clustered** broadcast cycle,

$$A^S_{F,NC} = (R + k\cdot P)/2 + SIG\cdot F + P\cdot F.$$

and the tune-in time is:

$$TN^S_{F,NC} = P/2 + F\cdot R/k + (k/l\cdot k/2 + S + P_f\cdot F)\cdot P$$

**Broadcast Disks:** The access time and tune-in time for retrieving a data item from disk $i$ can be derived similarly,

$$\begin{aligned} A^S_B &= (R + k\cdot P)/2 + (SIG + P)\cdot\sum_{j=1}^{N} F_j\cdot f_j/(2f_i) \\ &\quad + S\cdot P \\ TN^S_{F,C} &= P/2 + P\cdot(S + k/2) + (R/k + P_f\cdot P) \\ &\quad \cdot\sum_{j=1}^{N} F_j\cdot f_j/(2f_i) \end{aligned}$$

## 4 The Hybrid Index Approach

Both the signature and the index tree techniques have advantages and disadvantages in one aspect or the other. For example, the index tree method is good for random data access, while the signature method is good for sequentially structured media such as broadcast channels. The index tree technique is very efficient for a clustered broadcast cycle, but the signature method is not affected much by the clustering factor. While the signature method is particularly good for multi-attribute retrieval, the index tree provides a more accurate and complete global view of the data frames based on its

[6] Each data frame may have different false drop probabilities. To simplify the cost model, we use average false drop probability to estimate the access time and the tune-in time when a large number of queries are sampled (i.e., many data frames are retrieved).

indexed value. Since the clients can quickly search in the index tree to find out the arrival time of the desired data, the tune-in time is normally very short. Since a signature does not contain global information about the data frames; it can only help the clients to make a quick decision on whether the current frame (or a group of frames) is relevant to the query or not. The filtering efficiency heavily depends on the false drop probability of the signature. As a result, the tune-in time is normally high and is proportional to the length of the broadcast cycle.

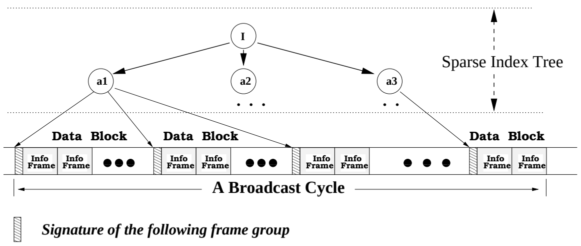


Figure 1: The Hybrid of Index Tree and Signature

In this section, we develop a new index method, called *hybrid index*, which builds on top of signatures a *sparse* index tree to provided global view for data frames and their corresponding signatures. The index tree is called sparse because only the upper $t$ levels of the index tree (the replicated part in distributed index) are constructed. The *key-search pointer* node in the $t$-th level points to a *data block* which is a group of consecutive frames following their corresponding signatures. Figure 1 illustrates a hybrid index. To retrieve information, the client can search the sparse index tree to obtain the approximate location information about the desired data frames. Since the size of the upper $t$ levels of an index tree is usually small, the overhead for this additional index is very small.

Since the hybrid index technique is built on top of signature method, it retains all of the advantages that a signature method has. However, the global information provided by the sparse index tree improves tune-in time considerably.

**Flat Broadcast:** For the **clustered** broadcast cycle with flat broadcast scheduling, the expected access time for hybrid indexing method is:

$$A^H_{F,C} = P^H + (TREE + SIG + P) \cdot F/2 + S \cdot P$$

where the average initial probe time for the index tree is half of the data block:

$$P^H = (TREE + SIG + P) \cdot D[B]/2$$

For the **non-clustered** broadcast cycle, there is one sparse index tree for each meta segment. Index tree technique is applied in each meta segment. Hence, the expected access time is:

$$A^H_{F,NC} = P^H + (TREE + SIG + P) \cdot F$$

For both clustered and non-clustered broadcast cycle with flat broadcast scheduling, the tune-in time primarily depends on the initial probe, the scanning of the sparse index tree, the signatures filtering and data access in the first located data block $B$, the selectivity of a query $S$, and the successive data access in the other $M$ meta segments. Therefore, it is upper bounded by:

$$\begin{aligned} TN^H_F = {} & (t+1) \cdot T + (M+S) \cdot P + \\ & \text{signatures in half data block B} + \\ & \text{false-drop data frames in half data block B} \end{aligned}$$

**Broadcast Disks:** For broadcast disks, the access time and tune-in time for hybrid indexing method can be similarly obtained as follows:

$$\begin{aligned} A^H_B = {} & P^H + (TREE + SIG + P) \cdot \sum_{j=1}^{N} F_j \cdot f_j/(2f_i) \\ & + S \cdot P \end{aligned}$$

$$\begin{aligned} TN^H_B = {} & (t+1) \cdot T + (1+S) \cdot P + \\ & \text{signatures in half data block B} + \\ & \text{false-drop data frames in half data block B} \end{aligned}$$

## 5 Evaluation of Index Methods

In this section, we make analytical comparisons of the access time and the tune-in time of the index tree, the integrated signature, and the hybrid method. To provide a baseline, data broadcast without indexing (denoted as *non-index*) is also included. Our comparisons are based on the cost models developed previously. Independent of the index methods, frames can be broadcast based on scheduling policies such as broadcast disks or flat broadcast. Thus, there are various combinations to be considered.

For flat broadcast, each data frame appears once in a given broadcast cycle. Therefore, the number of data frames in the broadcast $D$ equals the number of distinct frames $F$. For a clustered broadcast cycle (i.e., $M = 1$), half of a broadcast cycle needs to be scanned before the desired frames arrive.

For broadcast disks ($M$=number of minor cycles), $D$ is greater than $F$ due to frame duplication in the broadcast cycle. The access time and the tune-in time on different disks are different and depend on the speed of that disk. We denote the access probability, the access time, and the tune-in time for frames on disks $i$ as $P_i$, $A_i$, and $T_i$, respectively. Hence, the estimations for the average access time and tune-in time are $A_B = \sum_{i=1}^{N} A_i \cdot P_i$ and $TN_B = \sum_{i=1}^{N} TN_i \cdot P_i$.

Table 4: Parameter settings for performance evaluation

| | | |
|---|---|---|
| $F = 10^5$ to $10^6$ | $P = 1000$ | $f_{1,2,3}$ =3,2,1 |
| $n = 10$ | $T = 100$ | $M$ =1 to 200 |
| $k = 4$ | $p$=128 | $P_{1,2,3}$= 1/3 |
| $S = 1$ | $N = 3$ | $F_{1,2,3}$ = $F/10$,$2F/5$,$F/2$ |

Table 4 lists the parameter values used in the comparisons. We use these parameter values so as to maintain consistency with previous work [7, 8]. For some parameters (i.e., $F$ and $M$) a range of values has been used for the sensitivity analysis.

Both access time and tune-in time are measured in number of packets and are compared with respect to the number of distinct frames in a broadcast cycle which is varied from $10^5$ to $10^6$. We made the following assumption in the comparisons: a frame has capacity $P = 1000$ packets and a tree node takes up $T = 100$ packets which can contain $n = 10$ search keys and pointers, the size of a packet is $p = 128$ bits, four frames (i.e., $k = 4$) are grouped together in an integrated signature, the index tree is balanced (all leaves are on the same level) and each node has the same number of children. Since a broadcast cycle with selectivity $S > 1$ is logically equal to a broadcast cycle with the selectivity $S = 1$ and the data frame size $S$ times of the original broadcast cycle. Thus, in this paper, we only explore the case where the query selectivity $S$ is 1.

For broadcast disks, we assume that three disks are adopted (i.e., $N = 3$). The sizes of fast, medium, and slow disk are, respectively, 1/10, 1/2.5, and 1/2 of the total number of frames with relative spin speeds 3, 2, and 1. The aggregate client access probability to each disk is the same (i.e., $P_i = 1/3$, $1 \leq i \leq N$). Within each disk, all data frames have equal access probability. Therefore, the access probability for each data frame is inversely proportional to the size of the disk where the data frame is located. For a non-clustered broadcast cycle, we vary the scattering factor $M$ (i.e., from 1 to 200) to examine its impact on the performance of the index methods.

In what follows, we will evaluate the access time, and tune-in time of index methods for clustered and non-clustered broadcast cycles. For the clustered broadcast cycle, we consider both of the broadcast disks and flat broadcast as broadcast scheduling policies while for the non-clustered broadcast cycle, we only consider the flat broadcast scheduling.

### 5.1 The Clustered Broadcast Cycle

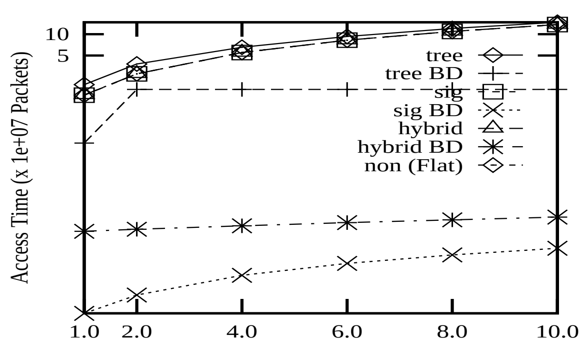


Figure 2: Access Time Overhead for a Clustered Cycle

In this section we evaluate the performance of the index methods for a clustered broadcast cycle. To make the figures clear, the $y$ coordinate is in log-scale and the access time is the overhead with respect to non-index for broadcast disk schedule.

Generally, for any broadcast schedules and indexing methods, the non-index for broadcast disks gives the shortest access time which is proportional to the size of a broadcast cycle (refer to Figure 2). For any particular indexing methods, the access time for broadcast disks (denoted with BD in the figures) is always much better than that for flat broadcast because of the skewed client access pattern.

When we consider flat broadcast only, the access time for the signature and the hybrid methods is similar to the non-index method as indicated by the overlapped curves in Figure 2; while the access time for the index tree gives an obviously worse access time. Compared with the non-index, the index overhead for the index methods (especially the signature and the hybrid methods) does not deteriorate the access time much for a clustered broadcast cycle.

For broadcast disks, due to index overhead, all three index methods give a worse access time than the non-index BD. The signature method performs better than the hybrid and the tree methods. Since the index tree is replicated in every minor cycle, its index overhead for broadcast disks is the highest. The broadcast cycle for broadcast disks is longer than that for flat broadcast. Moreover, the longer the broadcast cycle, the higher the index overhead. As a result, the access time difference between the indexing methods and the non-index method for broadcast disks is much larger than that for flat broadcast.

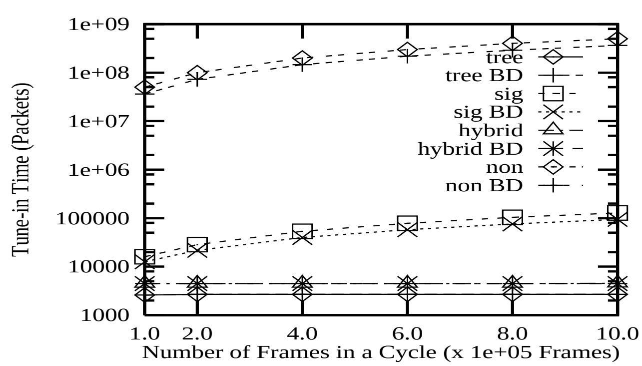


Figure 3: Tune-in Time for a Clustered Cycle

Next, we consider the tune-in time of the index methods. Figure 3 shows that the curves representing the tune-in time of broadcast disks and flat broadcast are overlapped for both the index tree and the hybrid methods. The non-index methods give much worse results than the index methods. This suggests that indexing can improve client tune-in time considerably. If we focus on the index methods only, the index tree method gives the best tune-in time and the signature method has the worst tune-in time. Broadcast disks can also improve the tune-in time of some index methods. For example, the broadcast disks improve the tune-in time of the non-index and the signature methods.

In summary, when a broadcast cycle is clustered based on an attribute, then the index tree scheme shows the best performance in terms of tune-in time, while the signature method has the best access time performance. The hybrid method performance pretty well in both criteria. Moreover, our comparisons show that the broadcast disks approach can improve both the access time and the tune-in time when the client access patterns are skewed, although the improvement for the tune-in time is not as significant as that for the access time.

### 5.2 The Non-Clustered Broadcast Cycle

In this section, we evaluate how these index methods perform with a non-clustered broadcast cycle (i.e., $M > 1$). We first fix the number of frames in a cycle to $10^5$ and vary $M$ from 1 to 200 to examine the influence of scattering factor on the system performance. Later, we will show the index performance against the size of a broadcast cycle. In our comparisons, the access time overhead is obtained with respect to the non-index method.

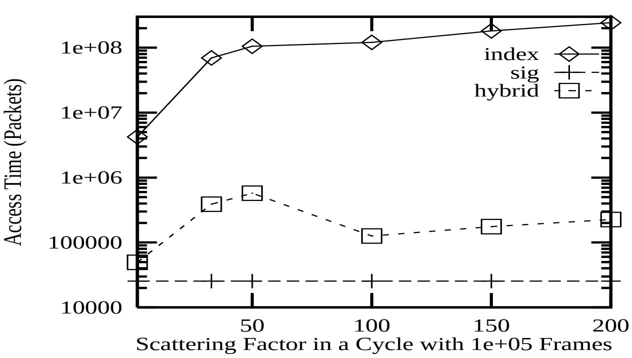


Figure 4: Access Time Overhead vs Scattering Factor

Figures 4 shows that the scattering factor has a great impact on the access time of the index tree method. Since there is an index tree corresponding to every meta segment, the index tree overhead increases rapidly as $M$ is increased. For the hybrid method, the number of the sparse index trees equals $M$. However, the sparse index tree overhead is very small and the initial probe time for index tree node decreases as $M$ is increased. Hence, the access time of the hybrid method is not proportional to $M$. As the flat curve indicates, $M$ has almost no influence on the signature method.

Figure 5 shows that the tune-in time of the index tree and the hybrid methods goes up quickly as $M$ is increased, while the tune-in of the signature index method remains the same. Since both the index tree and the hybrid methods need to probe each meta segment for the

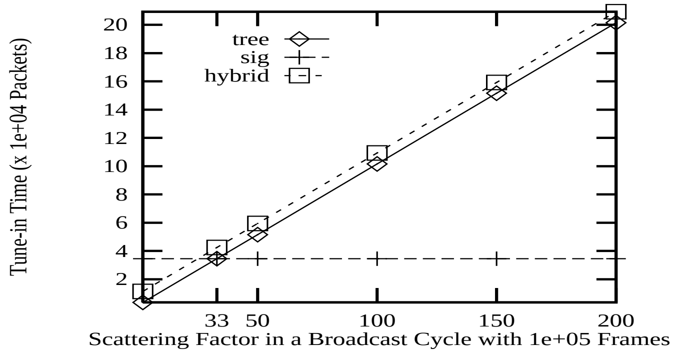


Figure 5: Tune-in Time vs Scattering Factor

possible arrival of the desired frames, the major advantage of the index tree and the hybrid methods, namely, short tune-in time, disappears when $M$ is greater than 33. Similar to the access time, there is no impact on the signature method for the tune-in time when the scattering factor changes. This suggests that the index tree and the hybrid methods are not applicable to a broadcast cycle with large scattering factor.

Next, we vary the size of broadcast cycle to see how these index methods act. In the following comparisons, we assume that the broadcast cycle is non-clustered with a scattering factor set to 100. Figure 6 shows that, similar to the clustered broadcast cycle, the access time of index tree method is much worse than that of the other two index methods. The signature method adds the least access time overhead to the non-index baseline.

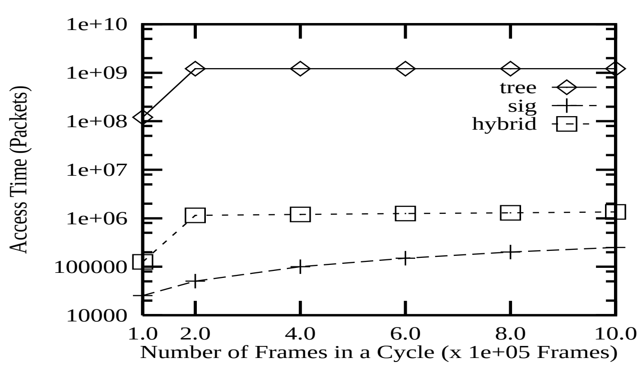


Figure 6: Access Time Overhead (Non-Clustered Cycle)

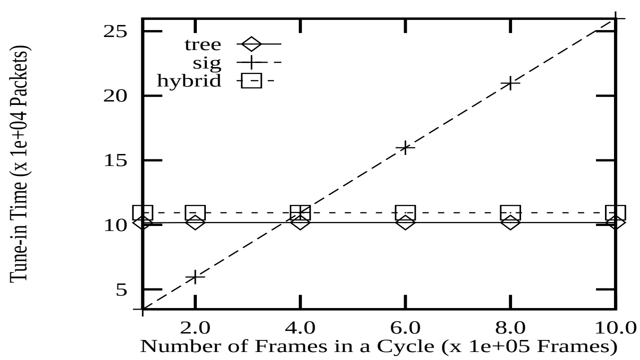


Figure 7: Tune-in Time (Non-Clustered Cycle)

On the other hand, Figure 7 shows that, unlike the clustered cycle, the tune time of the index tree and hybrid methods is not always better than the signature method. For small broadcast cycle (i.e., less than $4 \times 10^5$), the signature methods have the best tune-in time than the other two methods. When the length of a cycle increases, due to false drops, the tune-in time of the signature method increases quickly and becomes worse than the other methods again. As in the case of clustered cycle, the tune-in time of the hybrid method and the index tree method has the same relationship (i.e., the former is always a little bit worse than the latter one).

Considering both of the clustered and non-clustered broadcast cycles, we observe that the tune-in time of the signature schemes is proportional to the size of the broadcast cycle, while the other two methods always have almost the same tune-in time regardless of the size of the broadcast cycle. The reason is that the size of the index tree can be adjusted automatically according to the size of the broadcast cycle $F$ and the height of the index tree $h$, which is the only factor affecting the tune-in time, increases very slowly ($n^h \approx F$).

We can conclude from our comparisons that the signature method is good for non-clustered data broadcast and that the hybrid method, though not as good as the signature method, performs very well under various situations. The index tree method not suitable for non-clustered data, especially with high scattering factor.

## 6 Conclusion

Combining strengths of the previously proposed signature and index tree techniques, a hybrid indexing method is developed in this paper. This method has the advantages of both the index tree method and the signature method and has a good performance under various system conditions. Moreover, a variant of the hybrid indexing method has been demonstrated to be the best choice for multiple attributes indexing organization in wireless broadcast environments [6].

In this paper, we evaluate the performance of power conserving indexes based on index tree, signature and hybrid methods. Our evaluation of the indexing methods takes into consideration the clustering and scheduling factors which may be employed in wireless data broadcast. We develop cost models of access time and tune-in time for these three index methods by taking the following organizations on broadcast channels into consideration: i) flat scheduling and clustered; ii) flat scheduling and non-clustered; iii) broadcast disk scheduling and clustered. Our evaluation shows that the signature and the hybrid indexing techniques are the best choices for power conserving indexing of various data organizations on wireless broadcast channels.

## References


[1] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik. Broadcast disks: Data management for asymmetric communications environments. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 199–210, San Jose, California, May 1995.

[2] S. Acharya and S. Muthukrishnan. Scheduling on-demand broadcasts: New metrics and algorithms. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*, pages 43–54, Dallas, TX, October 1998.

[3] D. Aksoy and M. Franklin. Scheduling for large-scale on-demand data broadcasting. In *Proceedings of IEEE INFOCOM'98*, San Francisco, CA, March 1998.

[4] M.-S. Chen, P. S. Yu, and K.-L. Wu. Indexed sequential data broadcasting in wireless mobile computing. In *The 17th International Conference on Distributed Computing Systems (ICDCS97)*, Baltimore, Maryland, USA, May 1997.

[5] Q. L. Hu, W.-C. Lee, and D. L. Lee. A hybrid index technique for power efficient data broadcast. *Distributed and Parallel Databases Journal*, to appear.

[6] Q. L. Hu, W.-C. Lee, and D. L. Lee. Indexing techniques for power management in multi-attribute data broadcast. *ACM/Baltzer Mobile Networking and Applications (MONET)*, to appear.

[7] T. Imielinski, S. Viswanathan, and B. R. Badrinath. Data on the air - organization and access. *IEEE Transactions of Data and Knowledge Engineering*, June 1997.

[8] W.-C. Lee and D. L. Lee. Using signature techniques for information filtering in wireless and mobile environments. *Distributed and Parallel Databases Journal: Special Issue on Database and Mobile Computing*, 4(3):205–227, July 1996.

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

[10] K. Stathatos, N. Roussopoulos, and J. S. Baras. Adaptive data broadcast in hybrid networks. In *Proceedings of the 23rd VLDB Conference*, pages 326–335, Athens, Greece, August 1997.

[11] C.J. Su and L. Tassiulas. Broadcast scheduling for information distribution. In *Proceedings of IEEE INFOCOM'97*, Kobe, Japan, April 1997.