

# Information Filtering in Wireless and Mobile Environments

Wang-chien Lee

Dept of Computer and Information Science  
The Ohio State University  
Columbus, Ohio 43210-1277, USA

wlee@cis.ohio-state.edu  
FAX: 614-292-2911

Dik Lun Lee

Department of Computer Science  
University of Science and Technology  
Clear Water Bay, Hong Kong

dlee@cs.ust.hk  
FAX: (852) 2358-1477

## Abstract

This paper describes the issue of power conservation on mobile clients, e.g., palmtop computers, and suggests that signature methods are suitable for real-time information filtering on wireless communication services. Two signature-based approaches, namely simple signature and multi-level signature schemes, are presented. Cost models for access time and tune-in time of these two approaches are developed.

## 1 Introduction

Rapid advances on wireless data networks and personal computing open up new services and activities for mobile users. Various commercial and experimental mobile clients have appeared recently [2,6]. It is envisioned that mobile clients will be as popular as walkmans and portable TVs in the near future and revolutionize the information service market. Within a wireless communication environment, users with mobile clients have the freedom to access information services and communicate with other users without geographical limitations. To meet the market needs, ubiquitous services, such as fax-oriented messengers, nomadic conferencing and computing, mail-triggered applications, and information broadcast, will emerge as some of the most important research topics in the next decade [7].

In the paper, we consider the problems with information broadcast services for users equipped with battery powered mobile clients. In particular, we investigate techniques for mobile clients to filter out unwanted information, thus reducing the amount of information presented to the users and the amount of battery power consumed by the mobile clients.

### 1.1 Information Broadcast and Filtering

In the wireless environment, broadcast is an attractive method of disseminating information to mobile users because base stations are equipped with powerful communication equipments but mobile clients, restricted by cost, portability and power, can afford little or no transmission capability. Moreover, broadcast can scale up to an arbitrary number of users. In contrast to accessing traditional storage media, the performance of accessing information on air does not degrade as the number of users increases.

In broadcast, the base stations sends out a series of *information frames*. (See Figure 1.) A frame is the logical unit

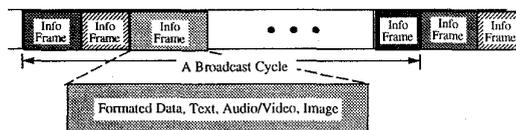


Figure 1: Information Stream.

of information broadcast on the air and may consist of multimedia information, including text, image, audio/video and other related data. It may contain a header (not shown in the figure) for meta-information indicating the type and length of the frame. The sizes of frames may be different. At the receiving end, users are allowed to specify conditions on the frames they are interested in. The mobile clients will only present to the users frames matching the conditions. Since the information frames are periodically broadcast, a complete broadcast of the information frames is called a *broadcast cycle*. From the user's viewpoint, the broadcast information can be treated as a stream of frames flowing along the time axis. There is no specific start and end frames for a broadcast cycle. Logically, a broadcast cycle starts with any frame and ends when the frame appears again. In a broadcast cycle, some important information frames may be replicated (i.e., frames with the same contents but treated as different frames). Information frames may be inserted, deleted, and modified. The updates will be reflected in the subsequent broadcast cycle.

### 1.2 Power Conservation

A major problem with mobile clients is their power supplies. Batteries are the main power source of most mobile clients. In order to make the mobile clients portable, small batteries, such as AA or AAA batteries, are likely to be used. However, these batteries have small capacity and need recharging or replacement after a short period of usage. Although processors and memories consuming less power have been developed, new generations of faster chips with higher clock frequency will continue to demand more and more power. Therefore, power conservation is an important issue for applications on mobile computing environment.

There are two factors affecting power consumption in mobile clients: (1) mobile clients can be switched between *active*

mode (full power) and *doze* mode [5], and (2) receiving messages consumes less energy than sending messages. Methods have been proposed for energy efficient query optimization [1] and information broadcast [8,9].

The duration that a mobile client must stay in active mode to answer a query is called the *tune-in time*, which is proportional to power consumption. *Access time* is the time required to collect all qualified frames. Without any access aid, both the tune-in time and access time are equal to the length of the broadcast cycle, because it is necessary to scan through all of the frames in a broadcast cycle to pick up the qualified frames. This is very inefficient in power consumption, because typically only a few frames in a cycle may qualify.

### 1.3 Indexing Broadcast Information

Access methods can be developed so that the mobile client can be turned off when the frames being broadcast are not qualified. By switching between active and doze modes, power may be saved.

In order to tell which frames would qualify ahead of time, auxiliary information about the contents of the frames must be added. Due to the limited number of broadcast channels, we assume that only one channel may be used for the primary and auxiliary information. Therefore, adding auxiliary information will increase the length of a broadcast cycle and thus increase access time. However, auxiliary information will reduce tune-in time. Thus, we must tradeoff between access time and tune-in time when we consider what auxiliary information is to be used and how it is going to be organized.

Two approaches, namely, hashing and indexing [8,9] have been proposed in the literature for encoding the auxiliary access information for wireless broadcast information services. However, both papers assume indexing based on one single key. Therefore, they won't support general queries involving various attributes of the information frame. To address this problem, we use signature file technique to encode auxiliary access information for information broadcast services.

The rest of the paper is organized as follows. In Section 2, we introduce the signature technique and its application to information broadcast. In Section 3, two signature schemes for information indexing and filtering are presented. In Section 4 and Section 5, we derive cost models for both signature schemes in terms of accessing time and tune-in time and compare their performance. Finally, Section 6 concludes the paper.

## 2 The Signature Technique

Signature methods have been used extensively for text retrieval [4], image database [11], multimedia database [12,14] and other conventional database systems [3]. A signature is basically an abstraction of the information stored in a record or a file. By examining the signature only, we can estimate whether the record contains the desired information. Naturally, the signature technique is very suitable for filtering information frames in a wireless broadcast environment.

A signature of a record,  $S_i$ , is formed by first hashing each value in the record into a bit string and then superimposing together all bit strings generated from the record into

record		IBM	Computer
attribute signatures:			
	IBM	000 010 101 001	
	Computer	001 000 110 010	
record signature ( $\vee$ )		001 010 111 011	
Queries	Query Signatures	Results	
1) IBM	000 010 101 001	$\leftarrow$ true match	
2) GE	010 001 000 011	$\leftarrow$ no match	
3) Ford	001 000 111 000	$\leftarrow$ false drop	

Figure 2: Signature generation and comparison.

the record signature. During filtering, a query signature,  $S_Q$ , is constructed in the same way and then compared to the record signatures. There are three possible outcomes of the comparison: (1) the record matches the query; that is, for every bit set in the query signature, the corresponding bit in the record signature is also set (i.e.  $S_Q \wedge S_i = S_Q$ ); (2) the record doesn't match the query (i.e.  $S_Q \wedge S_i \neq S_Q$ ); and (3) the signature comparison indicates a match but the record in fact does not match the search criteria. The last case is called a *false drop* and, in contrast, the first case is called a *true match*. To eliminate false drops, the record must be compared directly with the query after the record signature signifies a match. A signature failing to match the query signature guarantees that the corresponding record can be ignored. Figure 2 depicts the signature generation and comparison processes of a company record having two attributes, name and type.

For a mobile environment, the signature technique offers the following advantages for information filtering:

- Signature technique may be generally applied to various types of information media.
- Signatures are very easy to generate and search; thus, they are suitable for mobile clients where real-time searching with limited buffer space is required.
- A signature is very short compared to an information frame; therefore, the access time won't be increased drastically.
- The signature technique is particular good for multi-attribute retrieval, which is necessary for specifying precise filtering conditions.
- A signature file is easy to be "linearized" for broadcast on air and scanning by a mobile client, whereas a tree-based access structure will lose the speed benefit because random access cannot be done on a broadcast channel.

## 3 Information Filtering Using Signature Techniques

Signatures are constructed from and broadcast together with the information frames. The signatures may be broadcast as a group before the information frames or interleaved with the corresponding frames. Figures 3 and 4 illustrate these two approaches. For the non-interleaved signature approach, since the mobile users may start monitoring the broadcast channels at any moment, missing the signature

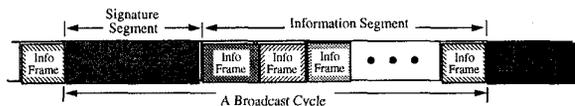


Figure 3: Non-interleaved Signatures.

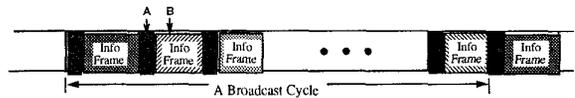


Figure 4: Interleaving Signatures.

segment means the user has to wait until the next broadcast cycle to access the signatures. The period of time from the moment a user tunes in until the first signature is received is called the *initial probe time*. Obviously, the non-interleaved method results in a longer initial probe time. Thus, we only consider interleaving approaches for the signature schemes discussed in this paper. It must be noted that during the initial probe time, the user may choose to switch to doze mode until a signature is encountered or to remain in active mode to scan for qualified information frames. The former will save energy, while the latter may return information frames earlier. Since the focus of the paper is on energy saving, we will assume the mobile client stays in doze mode during the initial probe time throughout this paper.

Different schemes may be used to organize signatures and information for broadcast. In this paper, we discuss two signature methods based on interleaving.

### 3.1 Simple Signature Scheme

The most intuitive signature method is to construct a *simple signature* for each information frame. The signature is broadcast before the corresponding information frame (see Figure 4).

When a mobile user wants to retrieve information from the broadcast channel, she/he specifies a query on a mobile client. A query signature  $S_Q$  is generated based on the specified query. Then the mobile client tunes into the channel and uses  $S_Q$  to compare with the simple signatures received. When a simple signature matches with the query signature, the information frame is received by the mobile client for further checking in order to eliminate false drops. If the frame is not a false drop, it will be included in the result frame set. When a received simple signature does not match with the query signature, the mobile client will switch into doze mode until the next signature frame arrives. If most of the simple signatures don't match with the query signature, the mobile client will stay in doze mode for the most part of a broadcast cycle, which thus saving a lot of energy.

In this scheme, the average initial probe time is half of the average size of an information frame and its signature frame. The access time and tune-in time, however, are dependent on the positions of the initial probe:

**position A:** If the initial probe falls in the middle of a signature frame (point A in Fig. 4), the mobile client has to stay active for the rest of the signature frame in order to detect the beginning of a new frame. Then, the mobile client switches to doze mode. The initial probe time, access time and tune-in time are as follows.

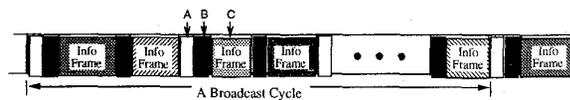


Figure 5: Multi-level Signature Scheme.

**Initial probe time** = the partial signature scanned + the first information frame.

**Access time** = initial probe time + a broadcast cycle.

**Tune-in time** = the partially scanned signature + every signature in a cycle + false drop and true match information frames in the cycle.

**position B:** If the mobile client initially tunes into the middle of an information frame (point B in Fig. 4), it will stay active for the rest of the frame since that's the only way to reach to the next signature.

**Initial probe time** = the partial information frame.

**Access time** = initial probe time + a broadcast cycle.

**Tune-in time** = the partial information frame + every signatures in a broadcast cycle + false drop frames and true match frames in the cycle.

### 3.2 Multi-level Signature Scheme

The multi-level signature scheme consists of multiple levels of signatures [10]. The lowest level contains simple signatures, each of which is generated from an information frame as in the simple signature scheme. Signatures at the upper levels, called *integrated signatures*, are generated from a group of information frames. Figure 5 illustrates a 2-level signature scheme. The white signatures in the figure are integrated signatures. An integrated signature indexes all of the information frames between itself and the next integrated signature of the same or at a higher level. (In the figure, an integrated signature indexes two information frames.) The black signatures are simple signatures for the corresponding information frames.

To reduce the false drop probability, the integrated signatures may use a hashing function which is different from the one used for generating simple signatures.

To answer a query, a query signature is generated for each layer of the signatures. After tuning into the broadcast channel, the corresponding query signatures are used to compare with different levels of signatures. If a signature fails in the comparison, the mobile client switches to doze mode until a signature at the same or upper levels arrives. If the result of the comparison is a match, the mobile client stays in active mode and continues the filtering process. Take the 2-level signatures in Figure 5 as an example. Query signatures  $S_Q$  and  $S_Q'$  are constructed for the integrated and simple signature levels, respectively. When an integrated signature is received,  $S_Q$  is used to match with the signature. If the match fails, the mobile client will go into doze mode until the next integrated signature arrives. If the match is successful,  $S_Q'$  is used to match with the following simple signature. If they match, the corresponding information frame is received; if not, the mobile client may go into doze mode until the next simple or integrated signature arrives.

Compared to the simple signature scheme, the multi-level scheme may have better tune-in time. However, the access time will obviously increase due to the higher signature overhead. The initial probe time, access time and tune-in time are dependent on the position of the initial probe.

**position A:** If the initial probe falls on an integrated signature frame (point A in Fig. 5), the mobile client has to be active for the rest of the integrated signature frame and start the filtering process with the next simple signature.

**Initial probe time** = the partial integrated signature scanned.

**Access time** = initial probe time + a broadcast cycle – the initial integrated signature.

**Tune-in time** = initial probe time + every simple signature in the first frame group + all but the initial integrated signature in a cycle + simple signatures following the qualified integrated signatures + false drop and true match frames associated with the qualified simple signatures.

**position B:** If the initial probe falls on a simple signature frame (point B in Fig. 5), the mobile client has to be active for the rest of the signature frame and then it goes into doze mode. The filtering process starts after the next signature arrives.

**Initial probe time** = the partial simple signature frame and its associated information frame.

**Access time** = initial probe time + a complete broadcast cycle.

**Tune-in time** = the partial simple signature scanned + every simple signatures before the first integrated signature arrives + all of the integrated signature in a cycle + simple signatures associated with the matched integrated signatures + (if the last integrated signature matches) the simple signatures before the initial probe + false drop and true match information frames associated with the matched simple signatures.

**position C:** If the mobile client initially tunes into the middle of an information frame (point C in Fig. 5), it will stay active for the rest of the frame. Then, filtering starts.

**Initial probe time** = the partial information frame scanned.

**Access time** = initial probe time + a broadcast cycle.

**Tune-in time** = the partial information frame scanned + every simple signature before the first integrated signature arrives + all of the integrated signatures in a cycle + simple signatures associated with the qualified integrated signatures + (if the last integrated signature matches) the simple signatures before the initial probe + false drop and true match information frames associated with the matched simple signatures.

## 4 Performance Analysis

There are several factors affecting the tune-in time and access time of the signature schemes. For example, we must consider the number and the size of the signatures, the filtering capability of the signatures, the false drop probability of the signatures, and the initial probe time. A performance evaluation has to take these factors into account. The filtering capability and false drop probability may be controlled by the size of the signatures and the bits set in the signatures. On the other hand, the initial probe time is related to the number of signatures interleaved with the information frames, and the access time and tune-in time depend on the number, size and false drop probability of the signatures.

### 4.1 Symbols and Parameters

$m$ : length of a signature in bits.

$w_b$ : number of 1's in a bit string generated from hashing.

$w_f$ : average number of 1's in a simple signature.

$\bar{w}_f$ : average number of 0's in a simple signature.

$s$ : the number of bit strings which are superimposed into a signature.

$p$ : the number of bits in a packet.

$r$ : the number of packets in a signature ( $r = \lceil m/p \rceil$ ).

$n$ : the average number of packets in an information frame.

$k$ : the number of information frames indexed by an integrated signature.

$l$ : locality of true matches (average number of true matches in a frame group).

$A$ : number of information frames in a broadcast cycle.

$A_f$ : number of information frames received due to false drops.

$A_t$ : number of information frames received due to true matches.

$I$ : number of integrated signatures in a broadcast cycle.

$P_f$ : false drop probability.

$P_f^s$ : false drop probability for simple signatures.

$P_f^i$ : false drop probability for integrated signatures.

$P_s$ : selectivity of a query.

### 4.2 False Drop Probability

The false drop probability is an important factor for the estimation of access time and tune-in time. The false drop probability  $P_f$  is generally defined as follows:

$$P_f = \frac{A_f}{A - A_t}.$$

However,  $P_f$  can also be presented in terms of the length of a signature,  $m$ , the number of bit strings superimposed into a signature,  $s$ , and the number of 1's in a bit string,  $w_b$ . In the following, we derive  $P_f$  assuming an unsuccessful search of a single value query. Multiple value queries which are Boolean combinations of single value queries may be derived similarly. Based on  $P_f$ , the estimation for the access time and tune-in time can be derived.

Assume that a good hashing function is used so that each of the potential bit strings has the same probability of being used in generating simple signatures. Given  $m$  and  $w_b$ , the probability,  $P(x, y)$ , that a particular set of  $y$  bits in a simple

signature is set to 1's by superimposing  $x$  bit strings is:

$$P(x, y) = \sum_{i=0}^y (-1)^i \binom{y}{i} \binom{m-i}{w_b}^x \binom{m}{w_b}^{-x}.$$

If  $x$  is sufficiently large and  $w_b \ll m$ ,

$$P(x, y) \approx (1 - (1 - w_b/m)^x)^y.$$

A simple signature is generated by superimposing bit strings hashed from the values in a frame. Therefore,  $P(s, 1)$  represents the probability of a bit position  $\beta$  to be set to 1 in the simple signature, where  $s$  is the number of distinct values in an information frame. There are  $m$  bits in a signature, so the average number of 1's in a simple signature is:

$$w_f = mP(s, 1) = m(1 - (1 - w_b/m)^s).$$

As a result, the average number of 0's set in a simple signature is:

$$\bar{w}_f = m - w_f = m(1 - w_b/m)^s \approx m e^{-w_b s/m}.$$

A false drop occurs when each of the bits in the simple signature corresponding to the 1's in the query signature is set to 1. In other words, a false drop occurs when the following condition holds:

$$\text{if } \alpha_i = 1 \text{ then } \beta_i = 1, \quad 1 \leq i \leq m$$

where  $\alpha_i$  and  $\beta_i$  are the  $i$ -th bit of the query signature and simple signature, respectively. Therefore, a false drop occurs when the following condition holds:

$$\text{if } \beta_i = 0 \text{ then } \alpha_i = 0, \quad 1 \leq i \leq m$$

For a single value query, the probability of false drops is as follows.

$$P_f \approx (1 - e^{-w_b s/m})^{w_b}.$$

Based on [13], the above formula is optimal when:

$$w_b = w_{opt} = m \cdot \ln 2 / s.$$

Consequently, the minimal false drop probability is as follows.

$$P_f \approx 0.5^{w_{opt}}.$$

### 4.3 Cost Models

In this section, we develop the cost models for the initial probe time, access time and tune-in time for the signature schemes we introduced. We use the number of packets as the unit for time estimation. To simplify our discussion, we assume that every information frame has the same number of packets. Therefore, the total number of packets for the data part is:

$$DATA = A \cdot n.$$

#### 4.3.1 Simple Signature Scheme

The total number of packets for the simple signatures in a cycle is:

$$SIG_s = A \cdot \lceil m/p \rceil = A \cdot r.$$

We use  $CYCLE_s$  to denote the length of a complete cycle for the simple signature scheme:

$$CYCLE_s = SIG_s + DATA.$$

The initial probe time is the period of time before the next signature arrives. Therefore, the average initial probe time is:

$$PROBE_s = (r + n)/2.$$

After the initial probe period, the filtering process will last for a complete cycle of the broadcast. Therefore, the average access time is the sum of the initial probe time and the broadcast cycle.

$$ACCESS_s = PROBE_s + CYCLE_s = (A + 0.5) \cdot (r + n).$$

Let  $PT_s$  denote the period in which the mobile client is active during the initial probe time.

$$PT_s = \frac{r \cdot 1/2 \cdot r + n \cdot 1/2 \cdot n}{r + n} = \frac{r^2 + n^2}{2(r + n)}.$$

To estimate the tune-in time, we have to first estimate the number of true matches. Let  $P_s$  denote the selectivity of a query, the number of true matches is:

$$A_t = A \cdot P_s.$$

In the filtering process, the mobile client has to tune in for all of the signature frames. In addition, it has to tune in for the true matches and false drops as well. Therefore, the tune-in time is:

$$\begin{aligned} TUNE_s &= PT_s + SIG_s + A_t \cdot n + A_f \cdot n \\ &= PT_s + SIG_s + A_t \cdot n + P_f^* \cdot A \cdot n - P_f^* \cdot A_t \cdot n \\ &= PT_s + SIG_s + DATA \cdot (P_s + P_f^* - P_s \cdot P_f^*). \end{aligned}$$

#### 4.3.2 Multi-level Signature Scheme

In this section, we assume a two-level signature scheme, which consists of integrated signatures and simple signatures. Assume that  $k$  information frames are grouped together to form an integrated signature. The total number of integrated signatures is:

$$I = \lceil A/k \rceil.$$

The total number of packets for the integrated signatures in a cycle is:

$$SIG_i = I \cdot r = \lceil A/k \rceil \cdot r.$$

Thus, the length of a complete cycle is:

$$CYCLE_m = SIG_s + SIG_i + DATA.$$

The average initial probe time is derived based on the probability of the initial probe locations and the corresponding probe time:

$$\begin{aligned} PROBE_m &= \frac{k \cdot n}{k \cdot n + k \cdot r + r} \cdot \frac{n}{2} + \frac{r}{k \cdot n + k \cdot r + r} \cdot \frac{r}{2} \\ &\quad + \frac{k \cdot r}{k \cdot n + k \cdot r + r} \cdot \left(\frac{r}{2} + n\right) \\ &= \frac{k \cdot n^2 + (k+1)r^2 + 2 \cdot k \cdot nr}{2(k \cdot n + (k+1)r)}. \end{aligned}$$

The access time is as follows.

$$ACCESS_m = PROBE_m + CYCLE_m.$$

The tune-in time in the initial probe period is:

$$\begin{aligned} PT_m &= \frac{k \cdot n}{k \cdot n + k \cdot r + r} \cdot \frac{n}{2} + \frac{k \cdot r}{k \cdot n + k \cdot r + r} \cdot \frac{r}{2} \\ &\quad + \frac{r}{k \cdot n + k \cdot r + r} \cdot \frac{r}{2} \\ &= \frac{k \cdot n^2 + (k+1)r^2}{2(k \cdot n + (k+1)r)}. \end{aligned}$$

To simplify the formula for our estimate of the tune-in time, we assume that the integrated signature for the group of the initial probe matched with the query. Therefore, the tune-in time can be approximated as follows.

$$\begin{aligned} TUNE_m &= PT_m + SIG_i + k \cdot r + k \cdot P_s \cdot n \\ &\quad + k \cdot P_f^i \cdot n + (I-1) \cdot P_f^i \cdot (k \cdot r + k \cdot P_f^i \cdot n) \\ &\quad + [P_s \cdot A/l](k \cdot r + l \cdot n + (k-l) \cdot P_f^i \cdot n). \end{aligned}$$

## 5 Comparison of the Signature Schemes

Table 1: Parameters of the cost models.

$A = 10000$	$P_s = 0.01$	$k = 4$
$n = 1000$	$p = 128$	$s_s = 100$

Using the formulae developed above, we compare the access time and tune-in time of the two schemes. Table 1 lists the parameter values used in the comparisons. In the comparisons, we assume that the selectivity of a query is 1%. For the multi-level scheme,  $k$  information frames are grouped together to generate an integrated signature. The false drop probability for the simple and integrated signatures can be calculated based on  $m$  and the number of bit strings superimposed,  $s_s$  and  $s_i$ , where  $s_i = s_s \cdot k$ . We vary the signature size,  $m$ , from 1 to 35 packets to observe the changes on access time and tune-in time.

Figure 6 shows that the access times of the two schemes are linearly proportional to the size of signatures. Since the overall size of the information frames is fixed at  $10^7$  packets, the increase in access time over the signature size represents the overheads of the signature schemes. From the figure, we also find that the overall size of signatures plays an important role to the increased access time.

Figure 7 shows the tune-in time for the two schemes. From the figure, we may observe that the tune-in time decreases to the minimal and then increases again as the size of signature increases. From the data collected, we observe that at those minimal points the false drop probability is so low that false drops become insignificant to the tune-in time. The signature sizes and access time corresponding to the minimal tune-in time of the two schemes are listed in Table 2. The table shows the best signature sizes for the parameters used. From the table, we may estimate that the access delays are less than

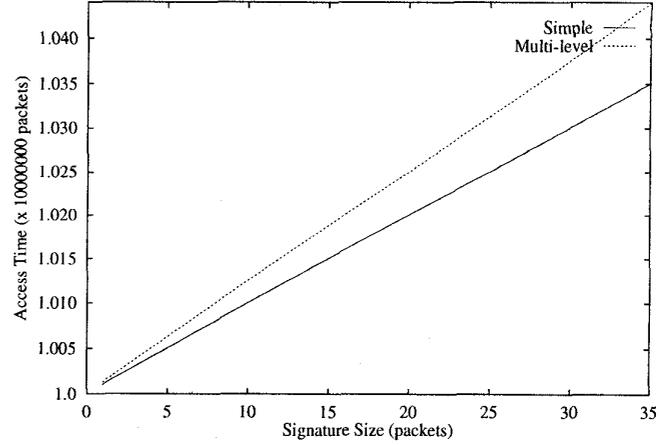


Figure 6: Access time vs Signature size.

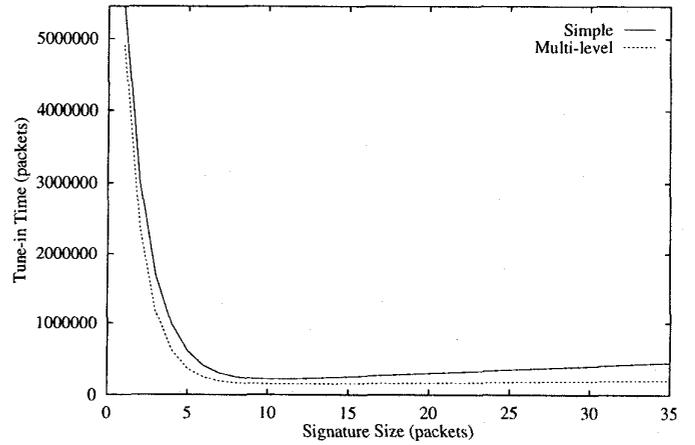


Figure 7: Tune-in time vs Signature size.

1.4% for both schemes, while the tune-in time save are more than 97.7%.

Next, the tune-in times of these two schemes are compared in Figure 8 in terms of access time. We found that with restricted delay, i.e., less than 0.6% of the optimal access time, the simple scheme has a better tune-in time than the multi-level scheme. However, the multi-level scheme's performance is comparable under the circumstances. For access delays of more than 0.6% of the optimal access time, the multi-level scheme has a much better tune-in time than the simple scheme.

However, note that the performance of the multi-level scheme may be affected by the locality of the true matches

Table 2: Sig. Size, Access Time w.r.t. Minimal Tune time

Scheme	Minimal		Sig. Size
	Tune Time	Access Time	
Simple	221620.82	10100505.00	10
Multi-level	155221.61	10138007.58	11

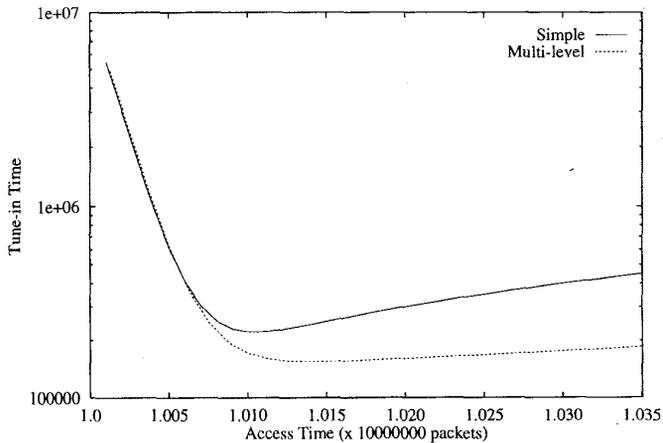


Figure 8: Tune-in time vs Access Time.

in a frame group and the overlaps among the superimposed bit strings. In the comparisons, we assume that the locality is 1 and there is no overlap. In other words, we compare the worst case of the multi-level scheme to the simple scheme. For applications which group together similar information frames (those having the same values for most of the indexed attributes), the performance of the multi-level scheme is expected to be better than as shown in the figure.

## 6 Conclusion

We believe that most queries on broadcast information only select a small number of information frames. With a reasonable false drop probability and small signature overhead, the signature schemes will be the natural choice for information broadcast services. Compared to traditional indexing, the signature method is particularly suitable for mobile clients because it can perform real-time filtering with little processing and memory requirement.

In this paper, we propose two signature schemes, simple and multi-level, for indexing and filtering information broadcast on air. Different from performance evaluation of traditional disk access, tune-in time, corresponding to battery consumption in the operation, and access time are used as performance criteria, which are the factors considered for our signature design. Cost models for tune-in time and access time are developed for both schemes. Finally the tune-in time and access time among the two signature schemes are compared by varying the size of signatures. The tune-in time performance with the same access time among the schemes is also illustrated.

Our analysis and comparison shows that, with fixed signature size, the multi-level scheme has the better tune-in time performance, but has the worse access time than the simple signature schemes. If the access time delay is fixed, the simple scheme has a better tune-in time than the multi-level scheme. For loosed access time delays, the multi-level scheme has a much better tune-in time than the simple scheme. Most importantly, both schemes improve tune-in time performance dramatically over a broadcast channel without providing any auxiliary information with a reasonable access time overhead.

## References

- [1] R. Alonso & S. Ganguly, "Energy Efficient Query Optimization," Matsushita Information Technology Laboratory, MITL-TR-33-92, November 1992.
- [2] B. Barriginer, T. Burd, F. Burghardt, A. Burstein, A. Chandrakasan, R. Doering, S. Narayanaswamy, T. Pering, B. Richards, T. Truman, J. Rabaey & R. Brodersen, "Infopad: A System Design for Wireless Multimedia Access," *Wireless94, Calgary*, July 1994.
- [3] W.W. Chang & H.J. Schek, "A signature access method for the Starburst database system," *Proceedings of the Fifteenth International Conference on Very Large Data Bases*, Amsterdam, The Netherlands, Aug. 1989, 145-153.
- [4] C. Faloutsos & S. Christodoulakis, "Signature files: an access method for documents and its analytical performance evaluation," *ACM Transactions on Office Information Systems*, Vol. 2, No. 4, Oct. 1984, 267-288.
- [5] S. Ganguly & R. Alonso, "Query Optimization in Mobile Environments," Department of Computer Science, Rutgers University, LCSR-TR-219, 1993.
- [6] Wireless Data Group, "Envoy: Personal Wireless Communicator," Motorola, Inc, Brochure, July 1994.
- [7] T. Imielinski & B. R. Badrinath, "Mobile Wireless Computing: Solutions and Challenges in Data Management," Department of Computer Science, Rutgers University, DCS-TR-296, 1993.
- [8] T. Imielinski, S. Viswanathan & B. R. Badrinath, "Power Efficiency Filtering of Data on Air," *Proceedings of the International Conference on Extending Database Technology*, 1994, 245-258.
- [9] T. Imielinski, S. Viswanathan & B. R. Badrinath, "Energy Efficiency Indexing on Air," *Proceedings of the International Conference on SIGMOD*, May 1994, 25-36.
- [10] D.L. Lee, Y.M. Kim & G. Patel, "Efficient signature file methods for text retrieval," *IEEE Transactions on Data and Knowledge Engineering*, Vol. 7, No. 3, June, 1995, 423-435.
- [11] S.-Y. Lee, M.-C. Yang & J.-W. Chen, "Signature file as a spatial filter for iconic image database," *Journal of Visual Languages and Computing*, Vol. 3, No. 4, December 1992, 373-397.
- [12] F. Rabitti & P. Zezula, "A dynamic signature technique for multimedia databases," *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, September 1990, 193-210.
- [13] S. Stiasny, "Mathematical analysis of various superimposed coding methods," *American Documentation*, Vol. 11, No. 2, February 1960, 155-169.
- [14] P. Tiberio & P. Zezula, "Selecting signature files for specific applications," *Information Processing and Management*, Vol. 29, No. 4, 1993, 487-498.