Data Indexing for Heterogeneous Multiple Broadcast Channel

Andrew Y. Ho and Dik Lun Lee
Department of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
Email: andrewho@cs.ust.hk, dlee@cs.ust.hk

Abstract

This paper studies a heterogeneous multiple channel environment (HMCE), in which the channels are controlled by different wireless operators. To the best of our knowledge, there is no previous research on this scenario. In this paper, we first present the architecture for HMCE which makes use of a centralized index server to broadcast index information about the broadcast data on a dedicated index channel. An analog can be drawn between HMCE and WWW: the wireless operators are web sites and the index channel is Google; Google indexes web pages so that users can find the web pages they want, whereas in HMCE the index channel indexes the data channels to help mobile users to find the data on the air. We propose three indexing methods to reduce the time and energy used to search for data in HMCE. Simulation results are obtained to evaluate the performance of the proposed methods.

1. Introduction

The rapid advance of wireless communication technologies during the last decade has brought a new vision in the computing industry – from traditional wired and stationary desktops to a fast growing area of mobile computing. The trend of using notebook computers, palm-size computers, and personal digital assistants (PDA) is already in full swing. Furthermore, the enhancement in reliability, transmission, and speed of wireless links facilitates the mobility of communication. Usually, a wireless communication environment consists of two sets of entities: a large number of users equipped with mobile devices (*mobile clients – MC*s) and a relatively fewer number of stationary *mobile service stations* (*MSS*) that have *base stations* (*BS*) or *access points* (*AP*) attached to provide wireless communication in geographical areas known as cells. Unlike a *MSS*, a *MC* is able to move freely from cell to cell and poses queries for retrieving data provided by the *MSS*.

There are two major modes for *MC*s to access information on the wireless channels: pull-based on-demand mode, which collects the queries sent by the *MC*s through an uplink channel, and then delivers the requested data through the downlink channel; push-based broadcasting mode, which broadcasts data on the broadcast channel continuously according to some previous data access statistics in order to reduce access latency and

Proceedings of the 2004 IEEE International Conference on Mobile Data Management (MDM'04)
0-7695-2070-7/04 $20.00 © 2004 IEEE

consumption of bandwidth, thus effectively allowing a large number of *MC*s to access information simultaneously. Furthermore, as receiving messages consumes less power than sending messages, *MC*s are able to stay longer with the limited power supplies in the broadcasting environment.

Because of business, economic and technical reasons, some service providers may want to use multiple low-bandwidth channels to achieve high combined bandwidth instead of getting a single high-bandwidth channel. In this ***homogeneous multi-channel*** environment, the server has full control over all channels in terms of scheduling data on the channels and will most likely use a fixed indexing and scheduling scheme. A mobile client typically subscribes to one or a few wireless operators and as such it is easy to identify the available channels and scan them to pick out the interesting information.

In this paper, we study a ***heterogeneous multiple channel environment*** (***HMCE***), which, in contrast, consists of a large number of wireless operators ranging from phone companies to amateurs operating in public radio frequency bands (e.g., Starbucks) [9]. The wireless operators disseminate information on channels that are not related or unorganized. An analog can be drawn between HMCE and WWW. In HMCE, the data channels are the web sites and the broadcast data are the web pages.

As information is disseminated through different service providers on different wireless broadcast channels, it is difficult for mobile users to identify the available channels (cf. web sites), let alone those containing the data they want (cf. web pages). They need to have knowledge about what channels are available and which channels carry their requested data. Since the broadcast pattern may change dynamically over time, any static channel information pre-programmed in the mobile devices may become outdated very soon. To find out where the expected data will be broadcast, the most straightforward method for *MC*s is to search all broadcast channels, but this is very time consuming and uses up a lot of battery power. Another way is to announce the data indices through a dedicated index channel so that *MC*s can identify where the data will be broadcast. In HMCE, we propose to makes use of a centralized index server to broadcast index information about the broadcast data on a dedicated index channel. Mobile clients listen to the index channel and then tune into the data channel according to the index information and scan for the data it wants. In other words, the index server/channel is the "Google" of the data on the air.

Researchers have proposed different broadcast scheduling [1, 2, 3] and indexing schemes [4, 5, 6] in order to reduce power consumption for a single broadcast channel. Yet, scheduling and indexing methods used in a single channel broadcast may not be directly applied to a multi-channel environment. There are also studies on multiple channel scheduling and indexing [3, 4, 7] that focus on a homogeneous multiple channel environment. To the best of our knowledge, there is no previous research on the HMCE scenario.

In this paper, the architecture of HMCE is proposed. We introduce indexing methods to reduce the time and energy used to search for data on multiple data channels. Three indexing models are described. Simulation results are obtained to evaluate the performance of the three proposed methods.

The rest of this paper is organized as follows. Section 2 introduces the background and related work on wireless broadcast. Section 3 describes the proposed methods for data indexing. Section 4 evaluates the performance of the proposed methods. Section 5 concludes the paper.

2. Background

In single channel environment, one way of reducing power consumption is by selective tuning [6, 5], which enables *MC*s to switch into active mode (power consuming) only when the expected data is being broadcast. A server is required to broadcast indexing information to make selective tuning works.

The (1, m) indexing scheme [6, 5] is an index allocation method that involves the complete index being broadcast m times in a broadcast cycle. *MC* traverses the index buckets and determine the offset to the requested data bucket. The tree-based indexing scheme [6, 5] was introduced in which an index is only partially replicated in the broadcast cycle. In this scheme, the data file is associated with a B+-tree index structure. The signature-based indexing scheme was proposed [8] for real-time information filtering. Basically, to access information, a query signature is constructed and compared with the broadcast signature. If the signatures match, all records indexed by the signature will be read until checked for correctness or until the expected record is found in the information frame.

Scheduling and indexing methods used in single channel broadcast may not be directly applied to a multiple channel environment. New algorithms and modified methods were thus proposed, although the algorithms did not address certain issues related to a HMCE. In [3], Hameed and Vaidya integrated the online algorithm with alternate labeling by assigning instances of the data item from a single channel schedule into a multiple channel schedule. In [4], Hsu et al. suggested a method for indexing and scheduling a multiple channel broadcast that considered data access frequencies based on distributed indexing [5]. In [7] Ke et al. proposed a scheduling method that concerned the conflict of data pages. The page-based strategy (PB) aims at allocating the data pages by their own access frequencies. The request-based strategy (RB) is based on user requests rather than pages. The conflict-free version (CFV) of the RB further enhances the schedule by checking if the pages in the same request are assigned in the same time slot.

3. Data Dissemination in Heterogeneous Multiple Channel Environments

Three indexing schemes in the centralized model for data dissemination in a HMCE are proposed.

3.1. Basic Model

The key point in the centralized model is the *central index server* (*CIS*), which is used to manage and broadcast index information about the data being broadcast on all of the broadcasting channels. In the architecture, we define the *broadcast agent* (*BA*) as any individual that has data to be broadcast on the broadcast channel.

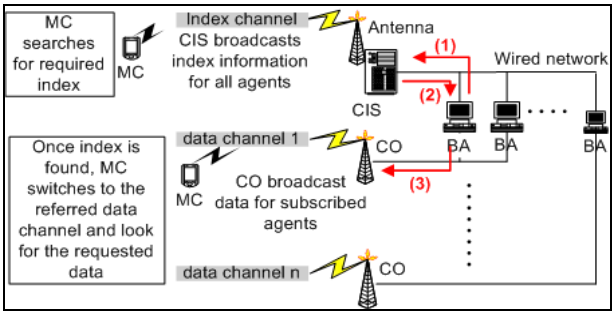

Figure 1. Centralized HMCE

Each *BA* is connected to the *CIS* and *CO* through a wired network (see Figure 1). The *CIS* is responsible for broadcasting index information on a dedicated wireless channel for the whole HMCE, while the *CO* is only responsible for broadcasting a *data message* (*DM*) on its

own wireless channels for *BA*s who subscribe to it.

Assume that a *BA* has data to be broadcast. The first thing for it to do is to send a *"data-to-send" notification* (*DTS*) to the *CIS* through the wired network (Figure 1). The content of the *DTS* follows a standard format as defined by the *CIS*, which includes the *BA*'s identification, the channel ID that the *BA* has subscribed to, the message identification, and a list of key attributes describing the data. When the *CIS* receives the *DTS*, it extracts the information from the *DTS* and converts it into the *index message* (*IM*) format, which contains a header with the *BA*'s ID, the channel ID, the message ID, the *IM* size, the number of attributes, and a pointer to the starting of attributes. Next, the *CIS* puts the *IM* into the broadcast queue for index broadcast.

Once the *CIS* receives the *DTS*, it is required to reply to the *BA* at the earliest time that it can broadcast its data (Figure 1). The replied message is called the *"earliest-send-time" notification* (*EST*). The value of the *EST* is equal to the broadcast end time of the *BA*'s *IM*. The *EST* is very important for serializing the *IM* and the indexed data in the centralized model. Suppose it is omitted and a *BA* broadcasts its data a short while after it has sent the *DTS* to the *CIS*. If there are a lot of *IM*s queued up in the *CIS*, it will take a long time for the *CIS* to broadcast all of them. Therefore, there is a chance that the *BA* broadcasts its data before the *CIS* has broadcast the corresponding *IM*. In this case, the data being broadcast by the *BA* are not properly indexed, and the corresponding *IM* broadcast by the *CIS* will be invalid. For data retrieval, if an *MC* has read this *IM* and tuned into the specified data channel, then it will wait forever or terminate after a timeout period, since the data have already gone on that channel. With the *EST* replied by the *CIS* to the *BA*, they can ensure that the data will have an index to indicate, providing the data are sent at or after the *EST*.

Once the *BA* receives an *EST* from the index server, it will wait until the indicated time. At that moment, the *BA* can send its data to the service provider – the *CO* (Figure 1). Whether the *CO* broadcasts the *BA*'s data immediately or appends it to an internal broadcast queue depends on the traffic of the channel.

Whenever the *MC* wants to retrieve data from the wireless channel, it will tune into the index channel and filter all broadcast *IM*s by listening to the channel until it finds an *IM* containing the attribute that matches its request. Then the *MC* can tune into the data channel indicated by the *IM* header and wait for the requested data to be broadcast. Since each *IM* header contains the corresponding IDs of the *BA* and the message, the *MC* only needs to check both IDs in the *DM* in order to determine if the *DM* is the one indicated by the *IM*. Otherwise, the *MC* can doze off until the end of the incorrect *DM*. The *MC* may also end the retrieval process if no attribute in *IM*s can be found matching its request within a certain period of time.

3.2. Signature Model

The environment of a signature model is the same as the environment used in the basic model. The major difference concerns how indices are constructed. In the basic model, whenever the *CIS* receives a *DTS* sent by *BA*s, all attributes in the attribute list attached to the *DTS* will be used to construct the *IM*. Also, each *IM* is only responsible for one *DM*. If there are a lot of *BA*s and all of them are sending a *DTS* with a long attribute list, then it will take a long period of time for the *CIS* to broadcast all the *IM*s on the index channel. Since the *EST* is equal to the end time of the corresponding broadcast *IM*, the *BA*s also need to

wait for a long period of time to start sending the data to the *CO*s.

In the signature model, to reduce the size of the *IM*, signatures are used instead of real attributes. An *attribute list signature* (*ALS*) is formed by hashing each attribute in an *attribute list* (*AL*) into a random bit string, and then superimposing all bit strings together (Figure 2).

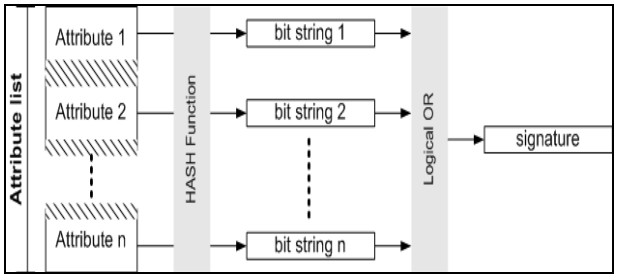

Figure 2. Generation of attribute list signature

During the filtering process, a query signature is constructed in the same way as the *ALS*. Then the query signature will be compared with the *ALS* using the logical AND operation to find if the query is potentially in the *AL*.

The *CIS* maintains two lists for managing signatures: a *channel signature* (*CS*) list for storing all *CS*s with one entry assigned to only one data channel in the system, and an order-array for storing the broadcast order and time for each entry in the *CS* list. Upon reception of the *DTS* from a *BA*, the *CIS* extracts the channel ID and the *ALS* from it. Then the *ALS* is superimposed onto the channel signature (*CS*) of the referring channel. The main purpose of the order-array is to keep track of the broadcast time for each non-empty *CS* (since it is not necessary to broadcast an empty *CS* for indexing purposes). Thus, the *CIS* can respond to the *BA* with the *EST* according to which channel it uses. During signature broadcast, the *CIS* goes through each non-empty entry in the order-array and only those *CS*s in the array will be broadcast. Moreover, after broadcasting, all *CS*s and the corresponding order-array entries will be cleared for future superimpositions.

In the signature model, indices are only signatures that guide the *MC*s to the data channel where the requested data may be found. As a result, the *BA*s are required to send with each data item an *AL* to their *CO* for broadcast, such that the *MC*s can check whether the query attribute is actually in the *AL* attached to the broadcast data.

A false drop can result from a signature comparison. This happens when an *MC* finds a matching *CS* with the query signature, but in fact, the corresponding channel does not contain the requested data item. If this is the case, the *MC* needs to leave the data channel and tune back into the index channel to filter other *CS*s. To determine the occurrence of a false drop, a timeout for searching the data channel is required. Since the *MC* tunes into the indicated data channel after finding a matched *CS* and waits for the data message, if the length of the searching period is not specified and if the *CS* is in fact a false drop, then the *MC* will wait forever on the data channel without finding any useful data.

3.3. Signature Model with Operator's Feedback

An improper length for the searching period can lengthen the retrieval time. No matter how carefully chosen is the length of the searching period, there is still the possibility of an incident of determining an 'incorrect false drop'. An incorrect false drop happens when the *MC* finds a matching *CS* on the index channel but cannot find the correct *DM* within the searching period on the corresponding data channel. In fact, the *DM* is broadcast on the indicated data channel, but after the *MC*'s expiration time. Incorrect false drops occur more frequently when the searching period is too short, or the traffic load of the data channel is heavy, since in both cases, the correct *DM* cannot be broadcast within the *MC*'s searching period.

To eliminate the effect of incorrect false drops, at least one *CS* needs to be broadcast within the length of a searching period before the corresponding *DM* has been broadcast. Hence, when the *MC* reads the *CS* and switches to the data channel, the *DM* will arrive during the searching period. In order to accomplish this method, cooperation from all the *CO*s is needed.

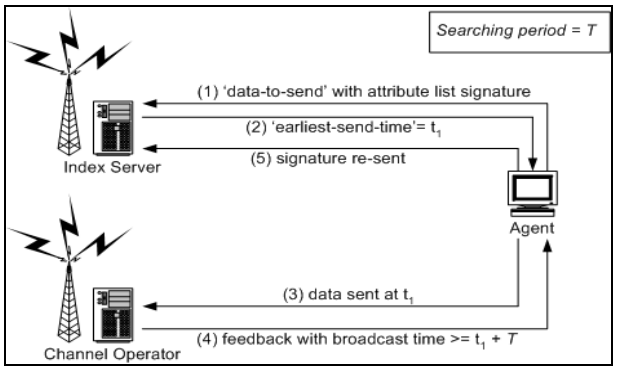

Figure 3. Signature Model with Operator's feedback

In both the basic and signature models, once the *BA* receives an *EST*, it waits until the specified time to send the data. But the *BA* has no idea about the time that the data message will be broadcast on the data channel by its *CO*. As a result, the *BA* has no way of ensuring that at least one *CS* containing the *ALS* of the data will be broadcast within the searching period prior to the broadcast of the data. Feedback from the *CO* plays an important role in notifying the *BA* about the time for the data broadcast. In contrast to the signature model, where the *CO* receives the data from the *BA*, besides appending the data to the broadcast queue, it also gives feedback to the *BA* who requested the data broadcast about the time that the *DM* will be broadcast on the channel. After getting the feedback from the *CO*, if the difference between the *CS* end time and the data broadcast time is more than one searching period, then the *BA* can send the same *ALS* again to the *CIS* one searching period prior to the broadcast time of the data, while avoiding any overlap with the searching period. The reason for this is straightforward: as the *MC* is still searching for the correct attribute on the data channel during the searching period, if the re-sent signature is broadcast within this period, it will be gone before the *MC* switches back to the index channel. The purpose of the re-sent *ALS* is only to prevent an incorrect false drop. Since the corresponding data is already in the *CO*'s internal broadcast queue, the second reply of the *EST* is not required by the *BA*. The message flow in the feedback model is shown in Figure 3. For the *MC*, the same procedure for data retrieval is used, but when an incorrect false drop occurs and the *MC* switches back to the index channel after timeout, the *MC* can use the re-sent signature on the index channel and switch back to the data channel to retrieve the *DM*.

4. Performance Evaluation

This section evaluates the performance of the three proposed heterogeneous multiple channels broadcast models by using simulation. The primary performance metric used for evaluating the models for *MC* is average access time.

For all experiments, CSIM18 [10] was used for implementing the simulation and the same parameters were used in the simulation environment. Besides, we assume that the capacities of the wired networks are much larger than the wireless broadcast channel (1 byte/unit of time), therefore the notification messages, such as *DTS* and *EST*, and any data going through it does not affect the performance of the system. As a result, time spends on wired networks is not counted. For attribute used in the simulation, a vocabulary is made, and each time, *BA* can choose 1 to 25 words from the vocabulary as the data attributes. Similarly, *MC* chooses one word from the list

for its query attribute. The length for each index message in the basic model is added up by the size of the attributes plus a 3 bytes header - *BA* ID, message ID and size of the message, which ranges from 9 bytes to 153 bytes in total. For the signature models, 128-bit (16 bytes) signature is used for each channel. The size of data which *BA* sends to *CO* is ranging from 10 to 10000 bytes. For basic model, 2 bytes IDs for verification is added, while in the signature models, size of attributes is added.

4.1. Impact of the number of channels on *MC*'s data access

MC access time measures the time elapsed from the moment the *MC* poses a request and starts listening on the wireless channels to the moment the requested data is received. The performance on *MC* access time is investigated with respect to the number of channels. The result is shown in Figure 4.

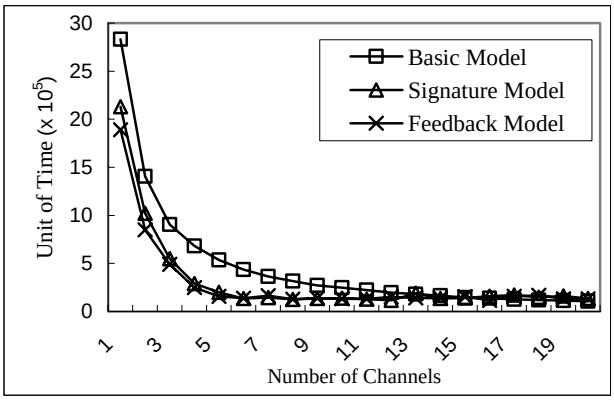

Figure 4. *MC* access time

In the figure, *MC* access time decreases as the number of channels increases. This is because workload on each channel is reduced by using a larger number of channels, thus enabling each data broadcast to start earlier. The figure also shows that *MC* access time in the feedback model achieves the lowest values at the beginning of the experiments, while the signature model has slightly higher values due to the longest time for retrieving data. As the number of channels increases, *MC* access time in all models tends to reach the same values and stay unchanged regardless of further increases in the number of channels. The reasons are as follows. First of all, the time that *CO* receives a data message and the time that *CO* broadcast the data message are getting closer. As *EST* is also the end time of the index broadcast, *MC* access time can be roughly formulated as the summation of the time spent on filtering on index channel, the data queuing time, and the time for broadcasting data message. Since the average data sizes in both models are the same, the filtering time becomes the main factor influencing the access time.

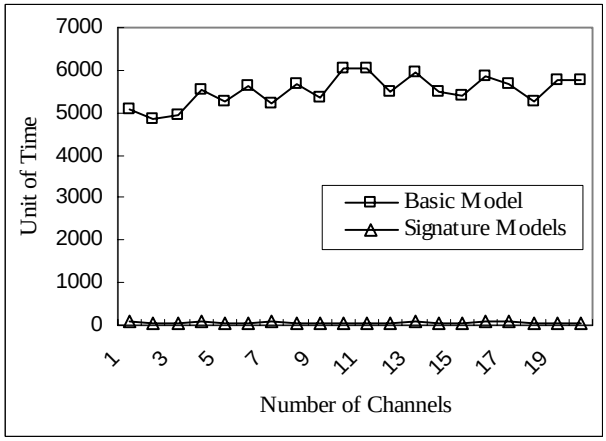

Figure 5. *MC* on index channels

Figure 5 shows the time that *MC* has spent on the index channel. The graph clearly shows that *MC* spends longer time in the basic model than in the signature models, due to the difference in index size. As a result, *MC* access time in the basic model obtains higher values in the experiments.

4.2. Impact of the searching period

Figure 6 shows the *MC* access time with respect to

different length of searching period in signature model and feedback model. Figure 7 shows the number of false drops with respect to the same searching period.

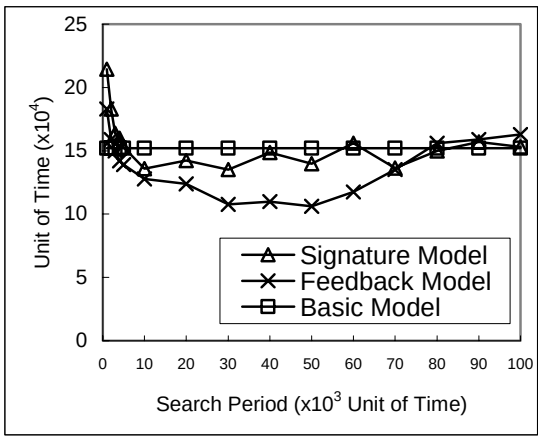

Figure 6. Access time

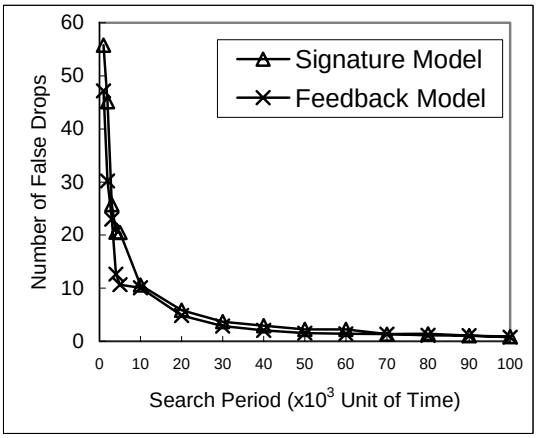

Figure 7. False drops

As shown in the figures, a short searching period causes a lot of false drops, since *MC* does not have enough time to reach the correct data message before the end of the searching period. Therefore, they interpret all the missed data as false drops. In the feedback model, since *ALS* of the data will be resent to *CIS* for broadcast again on the index channel, *MC* is able to retrieve the *CS* the second time. Moreover, as the resent of *ALS* is within one searching period prior to the real data broadcast, once *MC* gets the resent *CS*, it is most likely that it will receive the data message on the data channel. Therefore, *MC* in the feedback model incurs less access time than in the signature model. As the length of the searching period increases, the number of false drops decreases since *MC* is able to eliminate incorrect false drops. This also explains why the access time for both models tends to overlap with each other by increasing the searching period.

4.3. Impact of M-size

The number of false drops can influence the *MC* access time, but false drop itself is also influenced by the m-size. The m-size for a signature stands for the number of 1's generated by the signature generator (usually hash function). If the m-size is too small, then too many 0's will be in the signature, resulting in under utilized signatures. If the m-size is too high, then there will be too many 1's in the signatures, resulting in weakened filtering capabilities. This is because signatures with many 1's are much easier to match a query signature by chance. Figure 8 shows the number of false drops against m-size.

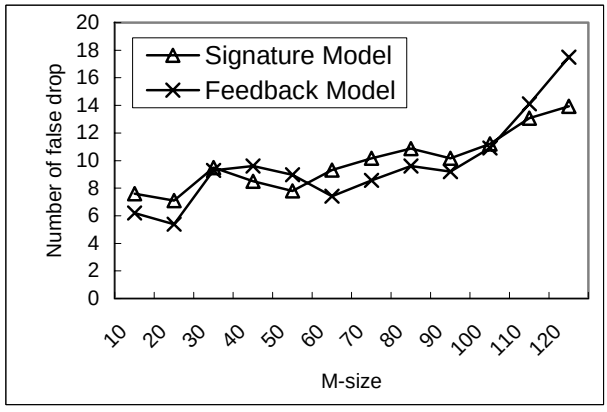

Figure 8. False drops Vs M-size

The figure shows that by using signatures with m-size equal to 20 bits, the lowest false drop rate can be obtained. Ideally, m-size with half signature size (64 bits in this case) will be the best, since the largest number of bit pattern for

a bit stream is constructed with same numbers of 1's and 0's (nCn/2), thus reducing the chance of collision. However, in the proposed models, each *CS* is in fact a superimposition of multiple *ALS*s from different *BA*s, and each *ALS* is in turn generated from a different number of attributes and thus contains different number of 1's. This explains why the minimum m-size is different from the theoretical result, which assumes uniformity in the *ALS*s. In a real operational environment, it is difficult to predict the optimal m-size, because a change in system loading can change the optimal m-size.

4.4. Impact of system loading on *MC* access time

System loading is defined as the fraction that the broadcast channels are occupied for broadcasting data messages. In the experiment, the loading is increased by adding agents to the system in order to increase the usage of each data channel. The maximum value '1' means that all channels are occupied at any instance during the simulation.

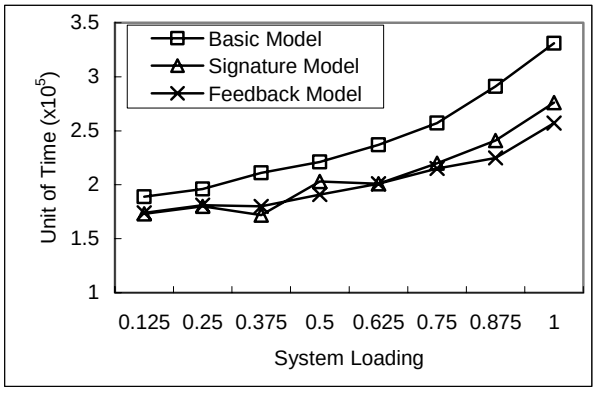

Figure 9. *MC* access time Vs System loading

Figure 9 shows *MC* access time with respect to different system loading. With light channel loading, fewer data will be sent. Hence, the number of false drops is reduced. As a result, *MC* in the signature and feedback models achieves low and similar access time. In addition, as channel loading increases, the number of data broadcasts also increases. Thus, *CO*'s broadcast queue will be filled with awaiting data messages, which also increase *MC* access time by delaying broadcast of the requested data. The fact that the feedback model has the lowest access time is the result of the feedback mechanism with signature resent.

In addition, when loading increases, the differences in *MC* access time between the basic and signature models also increase, which is basically caused by the lengthy *AL* attached to each *IM*. In the basic model, the number of *BA*'s requests is directly related to the number of *IM*s. If there are 300 *BA*'s requests, there will be 300 *AL*s broadcasted on the index channel. Suppose *MC*'s query attribute is in the last *BA*'s request, it is required to scan through all 300 *IM*s before switching to the data channel, which is time consuming compared to the signature models. Although *AL*s exist in the signature models, the number of *AL*s that *MC* has to process is reduced since each channel holds only a fraction of all *AL*s. *MC* is only required to filter the *AL*s that exist on the data channel. Therefore, the difference in access time increases as loading increases.

5. Conclusion and Future Work

In this paper, we propose an HMCE architecture consisting of independent wireless channel operators, broadcast agents and a centralized index server. Compared to the existing data dissemination schemes for a multi-channel environment, HMCE provides a channel through which *MC*s are able to know where to fetch their desired data. Furthermore, three indexing methods applicable to the HMCE architecture were proposed,

namely the basic model, the signature model, and the signature model with channel operator's feedback (feedback model). The basic model mainly uses data attributes from broadcast agents to form index messages. The signature model superimposes attribute list signatures for the purpose of indexing. Finally, the signature model with channel operator's feedback is an enhancement of the signature model for reducing the effect of incorrect false drops. We showed the both the signature model and feedback model are significantly better than the basic model.

Our work on HMCE represents the first attempt, to the best of our knowledge, to address a heterogeneous, autonomous broadcast environment. Much more research needs to be done to investigate different architectures (e.g., *CO*s and *CIS* can directly communicate in scheduling the index and data broadcast [9]) and other index schemes.

Acknowledgment

This work was supported in part by grants from the Research Grant Council of Hong Kong (Grant No. HKUST 6179/03E).

References

[1] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast disks: Data management for asymmetric communication environments," in *Proceedings of the ACM Conference on Management of Data (SIGMOD95)*, pp. 199-210. San Jose, California. May 1995.

[2] V. Gondhalekar, R. Jain, and J. Werth, "Scheduling on airdisks: Efficient access to personalized information services via periodic wireless data broadcast," in *IEEE International Conference on Communications (ICC 97)*, vol. 3, pp. 1276-1280. Montreal. 1997.

[3] S. Hameed and N.H. Vaidya, "Log-time algorithm for scheduling single and multiple channel data broadcast," in *Proceedings of the 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM97)*. Budapest. September 1997.

[4] C.H. Hsu, G. Lee, and A.L.P. Chen, "Index and data allocation on multiple broadcast channels considering data access frequencies," in *Proceedings of the 3rd International Conference on Mobile Data Management (MDM2002)*, pp. 87-93. January 2002.

[5] T. Imielinski, S. Viswanathan, and B.R. Badrinath, "Energy efficiency indexing on air," in *Proceedings of the ACM Conference on Management of Data (SIGMOD94)*, pp. 25-36. May 1994.

[6] T. Imielinski, S. Viswanathan, and B.R. Badrinath, "Data on air: Organization and access," *IEEE Transactions on Knowledge and Data Engineering*, 9(3). May/June 1997.

[7] C.H. Ke, C. Lee, and C.C. Chen, "Broadcast scheduling for multiple channels in wireless information systems," in *Proceedings of National Computer Symposium (NCS99)*, pp. 525-532. Taipei, Taiwan. December 1999.

[8] W.C. Lee and D.L. Lee, "Signature caching techniques for information broadcast and filtering in mobile environments," *ACM/Baltzer Journal of Wireless Networking (WINET)*, 5(1), 57-67. 1999.

[9] A.Y. Ho, "Data indexing in heterogeneous multiple broadcast channels," M.Phil. Dissertation, Department of Computer Science, Hong Kong Unviersity of Science Technology, Hong Kong, HKSAR, 2003.

[10] CSIM18, Mesquite Software Inc.