Reducing Latency through Efficient Channel Allocation Methods for Multiversion Data Dissemination In Mobile Computing Environments

Chitra Manikandan
Thiagarajar School of Management
Madurai

Abstract

This paper discusses several channel allocation methods for Multiversion data dissemination in mobile computing systems in which maximizing data currency (minimizing staleness) and meeting transaction deadlines are of equal importance as providing consistent data items to transactions. We consider two basic multiversion data organizations, namely vertical and horizontal data organizations and it uses an efficient mobile data compression technique, with minimal impact on the client, to efficiently use the low bandwidth of mobile communication systems and to reduce the mean access delay for data items and minimize the response times of the requests from the mobile clients. The broadcast and on-demand channels have different access performance under different system parameters and that a mobile cell should use a combination of both to obtain optimal access time for a given workload and system parameters. We study the data access efficiency of three channel configurations: i) all channels are used as on-demand channels (exclusive on-demand), ii) all channels are used for broadcast (exclusive broadcast) and iii) some channels are on-demand channels and some are broadcast channels (hybrid). Simulations on obtaining the optimal channel allocation, with two different Multiversion data organizations, were conducted and the result shows that an optimal channel allocation significantly improves the system performance.

Key words: multiversion data, broadcast, multiversion organizations, on-demand and Channel allocation.

1. Introduction

A mobile computing system consists of a set of mobile computers and stationary hosts connected by a fixed network. The mobile computers communicate with other computers through wireless communication, while the stationary hosts communicate with each other through the wired network. In order to support wireless communication for mobile computers, the geographical area covered by the system is divided into *cells*. Multiple versions of data are maintained in the server. The wireless communication in a cell is supported by a *mobile support station* (MSS), which is one of the stationary computers on the fixed network. The MSS controls the wireless communication channels in a cell and directly or indirectly provides various information services to the mobile users. As a result, it may be considered a logical data server providing and disseminating Multiversion data to its mobile clients in the cell.

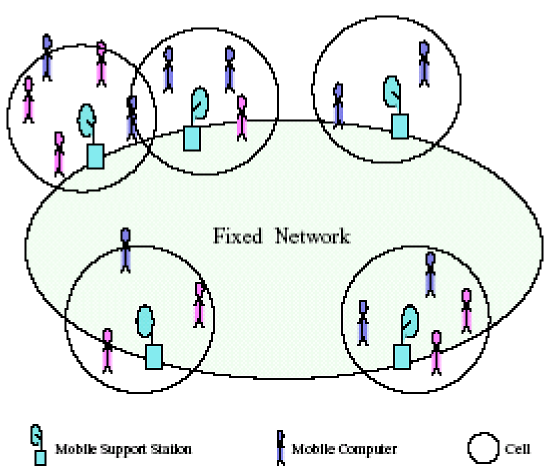

Figure 1. Wireless System Architecture.

Figure 1 shows the architecture for supporting mobile computing and illustrates the terminology such as mobile computers, MSSs, and cells. The performance objective is to reduce the mean access delay for Multiversion data items and minimize the response times of the requests from the mobile clients. The proposed modification on network layer protocols [7] will enable mobile computers to access information on Internet transparently, giving mobile users virtually unlimited amount of information at their fingertips. However, the limited bandwidth of the wireless channels between mobile computers and the MSSs will become the bottleneck when the service load is high. Thus, it is crucial to develop data access methods to efficiently utilize the channel bandwidth under different system loads.

We have two data organization methods to efficiently organize the Multiversion data to be transmitted to mobile clients and to reduce access time. It is shown in this paper that with these Data organization methods, a mobile cell can obtain optimal data access time under different system parameters by delivering frequently accessed data items (hereafter called *hot data items*) to broadcast channels and less frequently accessed data items (hereafter called *cool data items*) on on-demand, point-to-point channels. In this paper, we compare the data access efficiency among the channel allocation methods: i) all channels are used as on-demand channels (exclusive on-demand), ii) all channels are used for broadcast (exclusive broadcast) and iii) some channels are on-demand channels and some are broadcast channels (hybrid).

The rest of the paper is organized as follows. In the next section, we will discuss how to organize the Multiversion data to reduce access time. In section 3, we will discuss the data access methods for wireless channels, broadcast scheduling strategy and the criteria for data access efficiency. In Section 4, we provide analytical and simulation models to evaluate the data access time for the on-demand and broadcast channels, respectively. In Section 5, we evaluate the performance of channel allocation methods by varying different system factors and finally, Section 6 concludes the paper.

2. Multiversion Data Organization

2.1 Basic Organization

A set of multiversion data to be transmitted can be represented as a two-dimensional array, where dimensions correspond to transmission version numbers (Vno) and data ids (Did), and the array elements are the data values (Dval) of the items. That is, Dval[i,k]=v means that the k version of the i-data item is equal to v. Data items can appear in any order. Versions appear in descending order with the most recent versions appearing in the left most columns and the oldest version in the right most columns. This data representation can be extended to any number of data items and versions.

	Vno=3	Vno=2	Vno=1	Vno=0
Did=3	Dval=11	Dval=11	Dval=11	Dval=11
Did=5	Dval=18	Dval=18	Dval=18	Dval=15
Did=8	Dval=16	Dval=11	Dval=11	Dval=12
Did=9	Dval=15	Dval=14	Dval=14	Dval=14

Fig 2. Example of an array, representing a sequence of four versions of four data items.

A simple sequential transmission can be generated by linearizing the two-dimensional array in two different ways: horizontally and vertically. In horizontal data organization, a server transmits all versions (with different Vno) of a data item with a particular Did, then all versions (with different Vno, then all data items (with different Did) having the next Vno and so on. Formally, the Horizontal data organization transmits [Did [Vno,Dval]*]* sequences, whereas the Vertical data organization transmits [Vno[Did,Dval]*]* sequences. We expect that different strategies are more appropriate for different applications. If users require different versions of a particular data (for example, the history of a stock index change), Horizontal data organization is preferable. If users need the most recent data (for example, the current stock indexes), vertical data organization is expected to be more efficient.

2.2 Compressed Organization

Using some compression scheme, we can reduce the transmission data size and consequently the access time. A good compression scheme should reduce the transmission data size as much as possible with minimal impact on the client. Our compression scheme is based on the observation that data values do not always change from one version to another. In other words, Dval[i,k]=Dval[i,k-1]=…=Dval[i,k-N]=v, where the value of the item having Did=I remains equal to v for N consequent versions of a data item if its Dval does not change. Instead, the compressed scheme transmits a Dval only if it is different from the Dval of the previous version. In order not to lose information (as well as to support selective tuning), it also transmits the number of versions having the same Dval.

The generation of a compressed data organization proceeds in two steps. In the first step the compressed values of the data elements are produced and in the second phase, these values are transmitted based on the selected organization. In the first step, any sequence of elements with repetitive data values is replaced by the first element of the sequences and the length of the sequence.

Did=3	11*3			
Did=5	18*2			15
Did=8	16	11*1		12
Did=9	15	14*2		

Fig 3. The example array after the first step of the compression.

In the second step of the compression, the Vertical data organization is produced by using vertical linearization of the array. Formally, the Vertical data organization creates [Vno[Did(Dual * Repetition)]*]* sequences. Obviously we do not include into the transmission those versions, which already have been included "implicitly" with other versions.

In the case of the vertical data organization, it also makes sense to arrange the sequence of transmission of data elements within a single version weep and, make them dependent, not on Did but on the number of implicitly transmitting elements. Applying this recording to our examples, Vertical data organization is:

11x3 18x2 16 15 14x2 11x1 15 12

In this example, 14x2 and 11x1 switch their positions, because we transmit implicitly two 14s and only one 11. The idea is that we transmit first as "dense" data as possible, because when a client begins to road the transmission, the client has higher chances to find the necessary data elements in "more dense" data. Of course, this optimization works under the assumption that client access data uniformly.

For providing a rough estimation of the reduction of the transmission size (and, consequently, of the duration of the bcycle) due to our compression scheme, let us introduce Randomness and Randomness Degree to represent the repetitiveness of data from one version to the other. Let Randomness[i,k] = 0 if Dual[I,k] = Dual[I,k-1] and Randomness[i,k] = 1 otherwise. Then, the average randomness Randomness Degree[i] of the data element i all its versions represents how frequently the value of this element changes. We can use this average randomness as a parameter describing the probability that Dual [i,m] is not equal to Dual[i,m-1]. For instance, Randomness degree [i] = 0 means that Dual[i,m]=Dual[i,m-1], for any m. The smaller the degree of randomness, the higher the gain of our compression scheme. Hence, we can expect that the transmission of data having or a stock index of infrequently traded companies, etc) improves the "density" of transmission data.

As an example, consider the transmission of data item j assuring Randomness Degree [j] = 0.1. Then in average out of 100 versions we have 10 versions with values different from the values of the previous versions and 90 versions repeating their values. This means that instead of transmitting 100 data values we transmit only 10. We can roughly estimate that overhead created by the auxiliary symbols will not exceed 1 symbol per "saved" data item from the transmission. Assuming, one data item consumes 16 bytes and one auxiliary symbol consumes 1 byte, the gain is 100*16/(10*16+90*1) = 6.4, which corresponds to a 84% reduction of the transmission length. Similarly, the transmission shrinks about 9%, for Randomness degree=0.9. This reduction holds for both Vertical and Horizontal data organization.

2.3 Transmission Encoding

In order to make our transmission fully self-descriptive, we add all necessary information about version numbers and data items. One-design principles has been to make system flexible, allowing a client to understand the content of a transmission without requiring the client explicitly to be told of the organization of the transmission. For this purpose, we use four auxiliary symbols:

(Did), V (Vno), = (Assignment to Dval), ^ (Number of repetitions)

For the Vertical data organization

V3#3=11#5=18#8=16#9=15v2#3=11#5=18#8=11#9=14V1#3=11#5=11#5=18#8=11#9=14V0#3=11#5=15#8=12#9=14

V3, V2, V1 and V0 are the version numbers. They determine Vno of the data element, which follows it in the transmission. #3=1 means the element having Did=3 of the corresponding version (transmitted before) is equal to 1. So, V3#3=11#5=18 means Dval[did=3,Vno=3]=11 and Dval[Did =5,Vno=3]=18.Note that for Vertical data organization we do not need to include the version number in the transmission before each data element.

In the case of compressed beast, the symbol ^ is used to specify that the following versions of a data item have the same value. The other auxiliary symbols are also used to give a client the complete information about Did, Vno, and Dval in a uniform format for both the compressed and uncompressed multiversion data organizations. This is applicable to Horizontal data organization also.

The compressed Vertical data organization as

V3^3#3=11^2#5=18^0#8=16#9=15V2^2#9=14^1#8+1V1V0^0#5=15#8=12

3. Modeling the Wireless Channels

In this section, we develop models for the data access performance of the exclusive on-demand and exclusive broadcast channel allocation methods. Let us consider a cell in a mobile computing system. The cell consists of a MSS and mobile computers. The MSS maintains multiple versions of n data items, D_1,D_2, - - *Dn*, with average size *s*. The mobile computers may issue read request *s2*, which have size *r* each, to the MSS. Let λ_1, λ_1, λ_2, -- λ_n be the numbers of read requests generated from the mmobile computers per unit time on D_1,D_2, - - *Dn*. We assume that the MSS supports *c* wireless channels and that each channel has the same bandwidth *b*.

3.1 On-Demand Channels

If users require different versions of a data (for example, the history of a stock index change), Horizontal Data Organization is efficient. Horizontal data organization is preferred through On-Demand Channels. We first consider the access time for a mobile computer to retrieve a data item from the MSS through the on-demand channels. The mobile computer has to establish a connection to the MSS before submitting a request. Upon receiving the request, the MSS returns the requested data items to the user. Since there are c channels in the cell, there are at most c simultaneous communication sessions. We assume that a mobile computer will not issue a new request until the previous request is completed. Therefore, there are at most m requests to be serviced in the system. To simplify our analysis, we do not consider the time for hand shaking in order to establish the connections and the process time for MSS to retrieve the requested data items from the disk. Therefore, the service rate for an on-demand channel is:

$$\mu = b / (r+s)$$

The aggregate arrival rate of requests from the mobile computers is:

$$\lambda = \lambda_1 + \lambda_2 + \text{--} + \lambda_n$$

3.1 Broadcast Channels

If users need the most recent data (for example, the current stock indexes), vertical data organization is expected to be more efficient. Next we develop the cost model for the access time of the broadcast channels, given that the mobile computer has valid broadcast channel access information about how to access the data. Since there are *c* channels used for broadcast, the aggregate bandwidth for broadcast channels is *bc*. Assume that there are *x* data items broadcast periodically in the broadcast channels. The size of a broadcast cycle in the aggregate broadcast channel is *xs / bc*. The average probe time is xs / 2bc. Thus, the access time for retrieving data through monitoring broadcast channels is:

$$B = xs / 2bc + s/b$$

4. Performance Evaluation

In this section, we compare the access time performance of exclusive on-demand, exclusive broadcast and hybrid channel allocation methods. Since there are many factors, e.g., sizes of data items and messages, frequency of data requests, number/bandwidth of the channels, number of users, and number of data items, affecting data access efficiency, we vary them to observe the characteristics of access time performance for the methods. The access time of on-demand channels is obtained by simulation, while the access time of broadcast channels is evaluated using the cost model developed in the last section.

Queuing system is used to simulate the data access activities on the on-demand channels. The simulation is written in C and CSIM, which is a process-oriented simulation toolkit [10]. Each of the simulations lasts for 4,000 seconds. For clarity, we only show the performance of the hybrid method when the number of channels allocated to on-demand mode and broadcast mode ratio is 4/1. Obviously, the performance will be different when the channel assignment is different. The optimal assignment for various system parameters is presented in the table. In the following comparisons, we assume that a cell has 50 channels for 500 mobile users. The MSS has maintained a database of 2000 data items of 1000 bytes each.

Parameters	**Values**
Number of Mobile Computers	500
Number of Data Items	2000
Number of Channels	50
Request Arrival Rate	500/sec
Size of Data Item	1000 bytes
Size of Access Info.	10 bytes
Channel Bandwidth	1000 bytes/sec.
Simulation Time	4000 sec.

Table 1. System Parameter Settings.

We assume that the exclusive broadcast method periodically broadcasts 500 data items. For the hybrid channel allocation method, the top 50 data items are available through the broadcast channels while the other data items are available through the on-demand channels. The above parameter settings are summarized in Table 1.

We increase the size of data items from 50 bytes to 400 bytes, while fixing the other parameters. As shown in Figure 4, ’on-demand’, ’broadcast’ and ’hybrid’ represent the access time of the exclusive on-demand, exclusive broadcast, and hybrid channel allocation, respectively. The leap on the access time of exclusive on-demand method is due to the increase of waiting time for mobile computers to connect to the MSS. On the other hand, the access time of the exclusive broadcast method is proportional to the size of data items. Compared to the on-demand method, the broadcast method is worse in access time when the data access load is light (data item size is less than 90 bytes) but better when the load is heavy. As shown in the figures, the hybrid method does improve overall system performance significantly. It is interesting to observe, when the load becomes very heavy (i.e., data items size is above 190 bytes), exclusive broadcast method is the best choice.

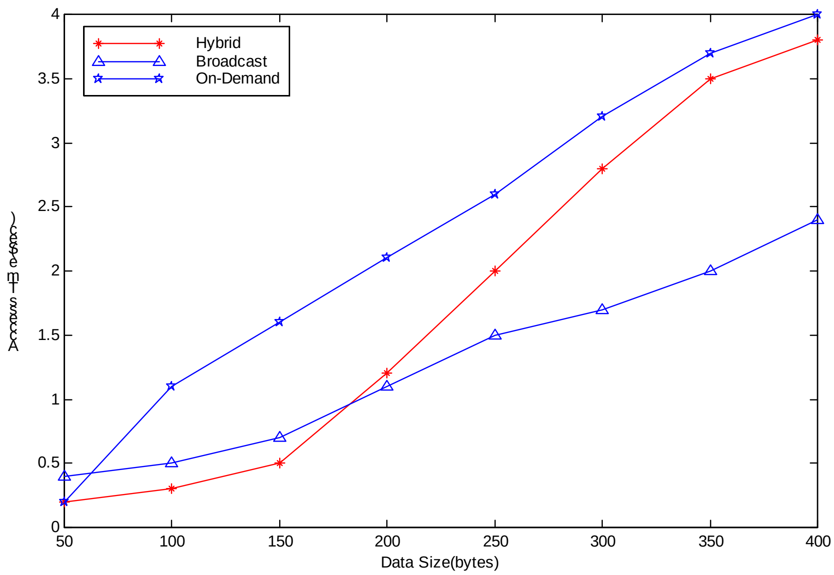

Figure 4. Access Time Vs Data Size.

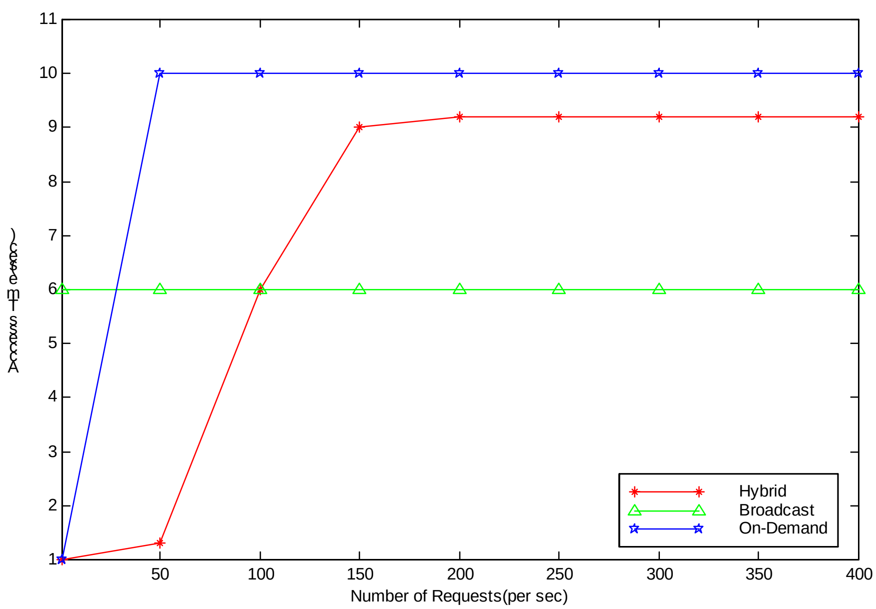

Figure 5. Access time Vs Request Arrival Rate.

The request arrival rate, which may be transformed into data request frequency, is another factor affecting the system load and thus the access time of the on-demand channels. Figure 5 shows that, for exclusive on-demand, when the request arrival rate is over 50, the system is overloaded. Otherwise, the access time for on-demand method is close to the transmission time for a data item. The figure also shows that the access time of exclusive on-demand method remains constant instead of rising for an overloaded system. That is because we assume that the mobile computers do not issue new requests until their previous requests are completed. We observe that the broadcast channels are not affected by the request arrival rate. No matter how frequent a data item is being requested, the access time for exclusive broadcast remains constant. Again the hybrid allocation gives a better access performance when request arrival rate is between 50 and 100. We varied the number of users and obtained similar results, since number of users may be transformed into data request frequency.

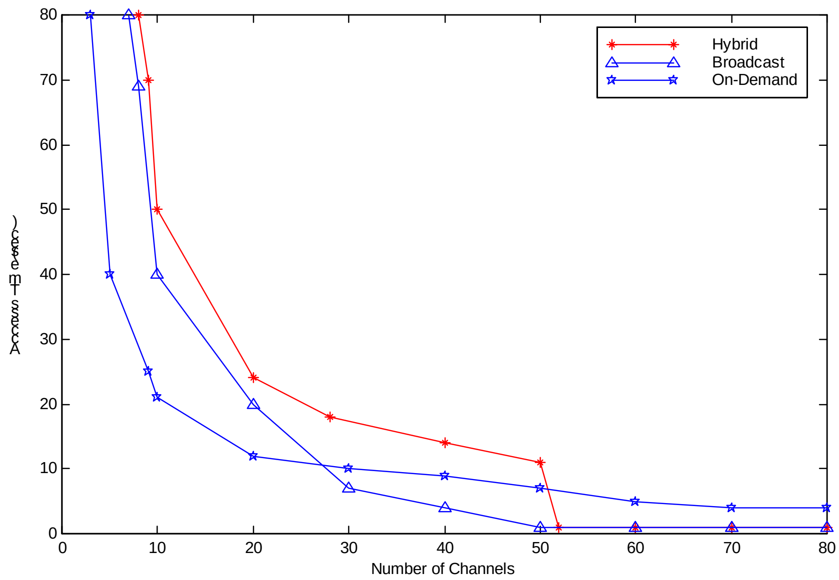

Figure 6. Access time vs Number of Channels.

We vary the number of channels in a cell to observe the change in access time. Figure 6 shows that the access time of the exclusive broadcast method drops rapidly when the number of broadcast channels increases from 5 to 20. From then on, however, increasing the number of broadcast channels has diminishing impact on the access time. This observation suggests that adding channels is not an effective way for lowering the access time for the exclusive broadcast method. For the exclusive on-demand method, the number of channels used has a dramatic impact on the access time. Under some circumstances, the system access time may improve significantly with a small increase of the number of channels.

We also compared the access time of the methods by varying the bandwidth of channels and obtained similar results. In fact adding more channels has the same effect as increasing channel bandwidth while maintaining the same number of channels, and vise versa. As shown in the figure, the performance of the hybrid approach is the best when the number of channels is between 28 and 52 and is generally pretty good under all circumstances. While the request frequency and the number of mobile computers in the system have no impact on the access time of broadcast channels, the number of data items available in the MSS has no impact on the access time of on-demand channels. Therefore, on-demand channels are good for providing user access to a large database.

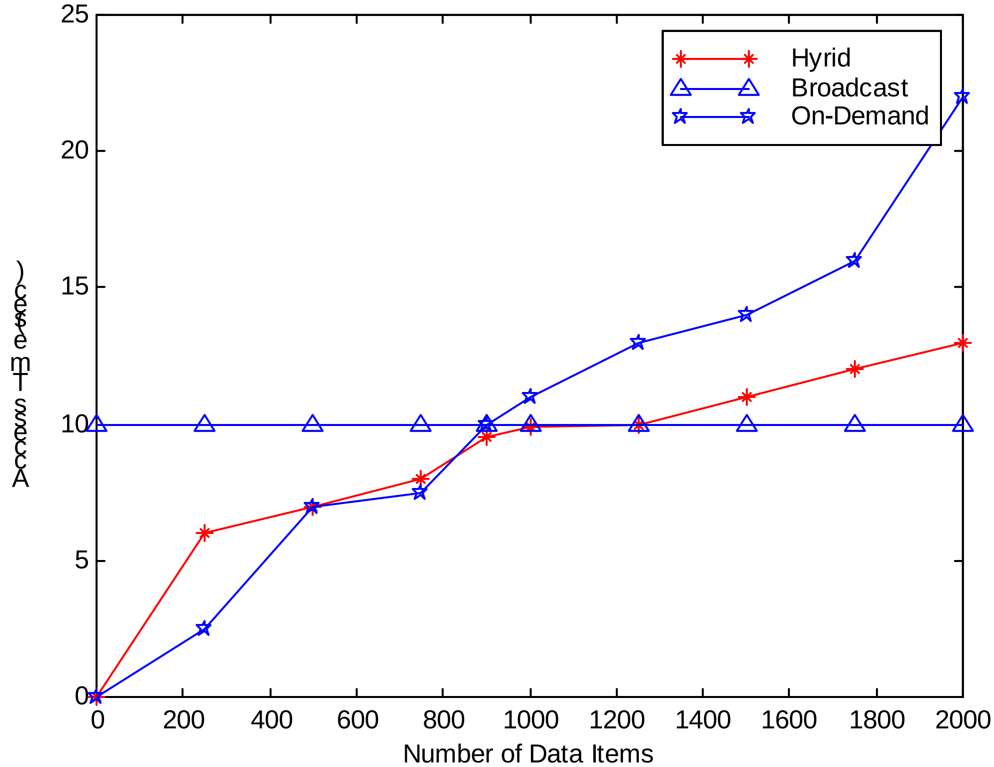

Figure 7. Access Time vs Number of Data Items.

Figure 7 shows the access time for the channel allocation methods corresponding to the number of data items available in the MSS. We observe that, with less than 900 data items, the exclusive on-demand method does better than the exclusive broadcast method. With more than 900 data items, the exclusive broadcast method has lower access time performance than that of the exclusive on-demand method, which always gives the same performance independent of the number of data items. If we offload a significant portion of the broadcast data items to on-demand channels, the overall system performance will improve. This is demonstrated by the performance of the hybrid method. When the number of data items is between 800 and 1200, hybrid allocation out perform both exclusive broadcast and exclusive on demand methods.

In the previous comparisons, we have demonstrated that, when the number of data items is small, broadcast channels are in general better under various system parameters. The comparisons can be scaled to systems with much larger number of data items to be disseminated— as long as user accesses are concentrated to a small number of data items, the broadcast channels may be used to alleviate communication workload at the MSS. However, when accesses are distributed to a large number of data items, shifting on-demand channels to broadcast channels are not a good solution.

We have shown that the hybrid channel allocation method with certain ratio of channels for broadcasting and on-demand services has better data access efficiency than the exclusive methods in general. Thus, an optimal approach for improving the data access efficiency of wireless channels is to dynamically decide the ratio of broadcast and on-demand channels based on system workload. Analysis of optimal channel assignments in a hybrid environment is quite complex and will be presented elsewhere [3]. In this paper, we use simulations to demonstrate the improvement that a dynamic channel allocation method can obtain over the exclusive on-demand method. In our experiments, we successively change the number of data items moved from the on-demand channels onto the broadcast channels. For each number, we exhaustively test all combinations of broadcast and on-demand channels in order to find out the optimal channel allocation with the best average access time. In order to demonstrate the performance of the dynamic channel allocation method, we select different parameter values for the number of mobile computers and the request arrival rate to represent scenarios with different workloads at the MSS.

5. Previous Work

Several organizations for indexing and filtering broadcasted data have been proposed in the literature [5,6,9]. These papers use index, hashing, and signature techniques to provide auxiliary information, which is interleaved with the data items in the broadcast, for data accessing on the broadcast channels. The main criteria of the performance used in the papers is the tune-in time. The main issue is to see, with certain access time delay, how much tune-in time may be saved. In order to efficiently use the network bandwidth, some of the researchers proposed to use the broadcast channels as broadcast file systems [1,2] or broadcast channels [4]. [2] focused on scheduling of the data broadcast. Two heuristics for broadcast scheduling were proposed to minimize the system access time. In addition to the scheduling problem, [1] discussed the issues for caching broadcasted data items. New cache management policies which took into consideration
access and broadcast frequencies of data items were developed and presented with simulation studies. [4] provided an architecture for wireless information services and discussed the broadcast and on-demand channels. However, they chose a different scheduling strategy and assumed one logical on-demand channel and one logical broadcast channel of different bandwidths. Thus, their model and analysis is different from ours. There is no simulation results and comparisons in the paper. All of the three papers propose to schedule the data items for broadcast based on their access frequencies. The strategy results in unbounded access time of the broadcasted data items, because some data items may be pushed back by frequently broadcasted data items. The users do not know how long they should monitor the broadcast channels to decide if the data items are available on the broadcast channels.

6. Conclusion

Wireless broadcast is an important technique for disseminating data to mobile users, because it scales up to an arbitrary number of users. In this paper, we have used two types of data organization to efficiently use the channels bandwidth and to reduce the response time and we study the data access performance of a hybrid system which combines the broadcast and on-demand services. In this system, Vertical Data organization is used in broadcast, whereas Horizontal data organization is served on on-demand channels. We studied and compared the performance of the hybrid method with systems where channels are used exclusively for broadcast or on-demand services.

References

[1] S. Acharya, R. Alonso, M. Franklin & S. Zdonik, “Broadcast Disks: Data Management for Asymmetric Communication Environments,” Department of Computer Science, Brown University, Technical Report No. CS-94-43, December 1994.

[2] T.-C. Chiueh, “Scheduling for Broadcast-based File Systems,” NSF MOBIDATA workshop at Rutgers University, New Brunswick, NJ, November 1994.

[3] Qinglong Hu, D.L. Lee &W.-C. Lee, “Optimal Channel Management for Mobile Computing Systems,” submitted for publication.

[4] T. Imielinski & S. Viswanathan, “Adaptive Wireless Information Systems,” Proceedings of SIGDBS (Special Interest Group in DataBase Systems) Conference, Tokyo, Japan, October 1994.

[5] 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.

[6] T. Imielinski, S. Viswanathan & B. R. Badrinath, “Energy Efficiency Indexing on Air,” Proceedings of the International Conference on SIGMOD, May 1994, 25–36.

[7] J. Ioannisdis, D. Duchamp & G. Q. Maguire Jr., “Ipbased Protocols for Mobile Interworking,” Proceedings of the International Conference on SIGCOMM, 1991, 235–245.

[8] R. Jain, The Art of Computer Systems Performance Analysis. JohnWiley & Sons, Inc., 1991.

[9] W.-C. Lee&D.L. Lee, “Using Signature Techniques for Information Filtering in Wireless and Mobile Environments,” Special Issue on Databases and Mobile Computing, Journal on Distributed and Parallel Databases, Vol. 4, No. 3, July 1996, 205–227.

[10] Mequite Software, Inc., CSIM17 Users’ Guide.

[11] Chitra Manikandan and Dr.R. Varadharajan, “Multi-Version Data Broadcast for Ordered Time Constrained Mobile Computing Systems” Proc. Int’l Conf. on Emerging Technology, pp. 156-160, 2004.

[12] S. Acharya, R. Alonso, M.J. Franklin, and S. Zdonik, “Broadcast Disks: Data Management for Asymmetric Communications Environments,” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 199-210, 1995.

[13] D. Barbara´, “Certification Reports: Supporting Transactions in Wireless Systems,” Proc. IEEE Int’l Conf. Distributed Computing Systems, 1997.

[14] D. Barbara´ and T. Imielinski, “Sleepers and Workaholics: Caching Strategies in Mobile Environments,” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 1-12, 1994.