

# Query Management for Efficient Data Access in Wireless Environment

Seema Verma  
AIM & ACT  
Banasthali University (India)

Rakhee  
Mathematics Group  
BITS, Pilani (India)

Savita Kumari  
School of Computer & Information  
Science  
Galgotias University, Greater Noida  
(India)

## ABSTRACT

With the cutting edge technology advance in wireless computers data broadcast has emerged as powerful tool for information dissemination to massive number of clients equipped with portable gadgets. In such computing system server continuously broadcast data over air channel without waiting for client's response and client retrieve only the data pertaining to him. To retrieve such broadcasted data mobile client submit a query to broadcast channel and get responded in sequential manner. Hence query plays central role in wireless communication. The queries evoked in wireless environment are much more complex and varied than that in traditional tethered environment. Owing to their complex nature handling of wireless queries is a challenge to the database and networking research community. This paper studied various query management strategies for wireless environment and evaluates them for multichannel broadcast system.

**Keywords:** Mobile client, mobile query, multichannel broadcast, wireless computing environment.

## 1. INTRODUCTION

The rapid development of wireless communication technology and battery-powered portable devices has led to emergence of wireless/mobile computing as a new era in data communication and processing. It ease human life by establishing a number of information services including weather forecast services, news, stock indices information, foreign exchange rates, election results, tourist services, airlines, railways schedules etc [1]. The advent of this technology makes it possible to access information anytime, anywhere using portable size wireless computers (e.g., laptop, palmtop, PDAs, and mobile phone). These portable devices communicate with a central stationary server via a wireless channel. Wireless computing provides database applications with useful aspects of wireless technology and a subset of wireless computing that focuses on query to a central database server is referred as wireless databases [2, 3]. The architecture of wireless communication system is based on client server model (CSM) where mobile client (MC) generates a query and server has to satisfy this query. In fact, server has heap of data, it has to organize them in proper order before

providing to MC. Air channels acts as a mediator to process the query from client to server and response from server to client. The major shortcoming with broadcast data items in a wireless environment is that data are accessed sequentially. The increasing number of broadcast items causes mobile clients to wait for a large time before receiving desired data item. Consequently, dependence of mobile devices on rechargeable batteries, which has limited capacities, is also another drawback of mobile data retrieval. The efficiency of wireless communication network is characterized by two parameters viz access time and tune time as defined below:

- Access latency: time elapse from the moment a request is initiated until all data item of interest are received.
- Tuning time: the amount of time spent by the client to listen for the desired broadcast data item(s).

The data delivery process in wireless environment is similar to traditional environment except the two generic challenges of energy scarcity and queries varieties, which generally do not exist in traditional database environment. Hence the query satisfaction process in wireless environment is somewhat more bounded and complex than traditional query delivery. The CSM system architecture demands the minimum use of battery power so that it can be utilized for long while MC obviously needs answer of query in minimum possible time because tardy answer of query some time lost its significance. For instance, a person driving car on highway generate query for vegetarian restaurant available within 5 km from him alongside the road. If he got result of query after he already had passed 8-10 km has no significance for him. The pure data broadcast in its generic form have low access time but to minimize energy consumption of battery we, employ some techniques like indexing partitioning, clustering of data which increase data access time and hence give rise to new version of problem of access latency [4]. This mean dilemma of high access latency itself is originated from solution of problem of minimization of tune time. Since the access time and tune time are contradictory to each other and no one can be optimized without having adverse effect on other. In the light of above complexity of data processing, as well as the retained characteristics of a mobile environment, it is essential to have an effective query retrieval mechanism that is able to

manage all of the above issues. The effective query management can be envisaged in three phases: query selection, query classification and viable query processing in variety of environments. This paper discusses all these points in context of multichannel wireless environment and presents algorithms generally used for query management. The rest of this paper is organized as follow: section2 presents brief overview of related work which formulates the problem discussed in this paper. Section 3 presents process of query selection and discuss basis for query selection used in wireless environment. Classification of queries based on context and types is presented in section 4. Section 5 gives query processing for wireless environment and recapitulates algorithms generally used for this purpose. Section 6 finally concludes the paper.

## 2. RELATED WORK

Wireless computing allows mobile clients to enjoy unrestricted mobility and ubiquitous information access in a region called cell. With data broadcast, clients listen to a broadcast channel to retrieve data based on their queries and hence are responsible for query processing. Acharya et al (1995) has designed a broadcast disk for query processing in asymmetric environment, which is different from traditional environment [5]. An item-based solution to the problem of broadcast query processing by providing air indexing was proposed by Imielinski et al. (1994) [4]. One major limitation of such item-based schemes is their lack of semantics associated with the broadcast. Thus, it is hard for mobile clients to determine if their queries could be answered from the broadcast entirely, forcing them to contact the server for possibly additional items. To remedy this, a semantic-based broadcast approach was suggested by Lee et al. [6]. This approach attaches a semantic description to each broadcast unit, called a chunk, which is a cluster of data items. This allows clients to determine if a query can be answered based solely on the broadcast and to define precisely the remaining items in the form of a “supplementary” query. Mobile clients can move between cells while being active and the intercell movement is known as a handoff process. Query management in intercell environment is discussed by Imielinski et al. (1994) [4] Waluyo, A.B. (2004) and Pfoser D. (2000) develop analytical model for query processing in moving objects and provides a comprehensive taxonomy of wireless query [7,8]. Cai et al. (2002) have studied real time query delivery by involving the adaptability concept hierarchy. This paper develops collective model comprising query selection, classification and processing, which are essential components of wireless communication environment.

## 3. QUERY SELECTION

The order of answering the query play vital role in both energy and time saving. The broadcast scheduler on server side place data in an order in which they are to be broadcasted. By knowing client access pattern in advance server can arrange data in requisite order. The popularity of each data items can be

known by various methods so that server has a priori knowledge of the client access probabilities and, based on these, it decides the contents of the broadcast, which is usually aligned from most popular data items to least popular data item. Although, such a priori knowledge seems like a strong assumption, there are several methods for determining the data access probabilities. The work of Sakata et al. (2003) and Yu et al. (2000) describe statistical methods for the estimation of these probabilities, whereas the work of Nicopolitidis et al. (2002) and Stathatos et al. (1997) present more practical approaches [10, 11, 12, 13]. One of the methods to count access probability is to gather data access statistics at regular interval of time and draw inference from it. Another method is to synthesize data with function  $\theta$  called zipf distribution function or access skew coefficient with assumption that the probability of a specific data item being requested follows a zipf distribution [14]. The Zipf distribution considers that access frequencies for queries follow 80/20 rules i.e. 80 % of client need only 20% of data items. This function provides skewness to data access pattern by adjusting the value of  $\theta$ . The value  $\theta=0$  represent flat access pattern while higher value of  $\theta=1$  represent skewed access pattern. Breslau et al. shows that Zipf distribution which is shown to be a realistic approximation of skewed data access [15]. Such distributions are used to characterize the popularity of one item among a set of similar data, like a web page in a web site. It may not represent real behavior of client but represent an average environment. In Zipf distribution, the probability of accessing the  $i^{\text{th}}$  item is:

$$\text{Access Probability, } P_i = \frac{\left(\frac{1}{i}\right)^\theta}{\sum_{i=1}^N \left(\frac{1}{i}\right)^\theta} \quad \text{where } 1 \leq i \leq N$$

We have simulated broadcast environment with MATLAB<sup>®</sup> 7.4 simulator and have been tested on some benchmarks where item popularities follow Zipf distributions [16]. The variation of demand probability with number of data item for a sample data base of 500 items is plotted in figure 1. MATLAB<sup>®</sup> has immense capability to simulate all type of data communication with real and synthesized data set along with ability to simultaneously solve mathematical problems and generate data set synthetically from Zipf function.

When client have complete knowledge of broadcast schedule, the placement of data over broadcast channel two major concerns related to the exploitation of a multichannel broadcast system. The first consideration is related to the capability of the server to concurrently transmit in all channels; the second is related to the capability of the client to simultaneously listen to multiple channels and also perform instantaneous hopping among channels. In such scenario we make following two

assumptions which guarantee a predictable schedule: the broadcast is cyclic, that is, it has a beginning and an end and the interval between successive transmissions of an item is constant for all broadcast cycles.

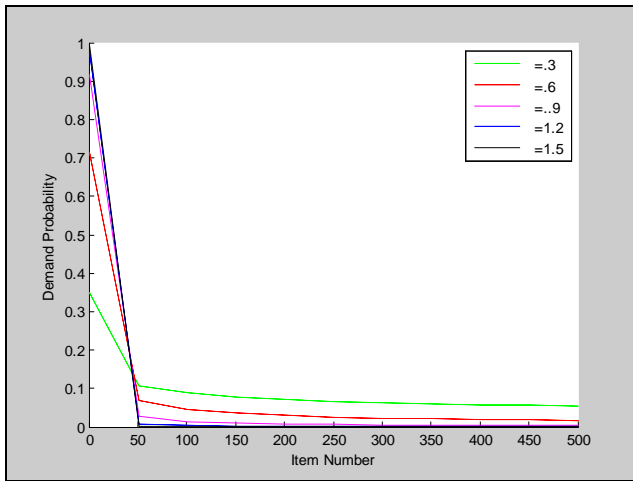


Figure 1: Plot of Variation in Demand Probability with

## 4. QUERY FRAME WORK FOR WIRELESS

There are different types of queries evoked in different communication systems. In the wireless environment the varieties of queries existence is more than the wired environment. The queries in former environment are more complex than in later. This paper is focus to query taxonomy and classification. Some queries exist only in a wireless environment and other exists in traditional environment, while some classes of quires exist in both environments. Figure 2 illustrates the query frame work in mobile environment. As can be seen from the figure, queries in a mobile environment are classified into two classes: traditional queries and context awareness queries. Context awareness queries are categorized into three types, namely; location-dependent queries, context-dependent queries, and hybrid queries.

### 4.1. Traditional queries

Traditional queries are common queries in traditional Database Management Systems (DBMS). These queries are typical queries that people are dealing with on a day-to-day basis in a wired network environment system. This type of query explicitly mentions the required information in the query statement and the query result is based only on the actual query itself. Examples of such type of queries are: (i) an investor wants to retrieve his account statement of Reliance Mutual fund or (ii) a person wants to know up gradation of his ticket he has already purchased for Common Wealth Games – 2010.

### 4.2. Context Awareness Queries

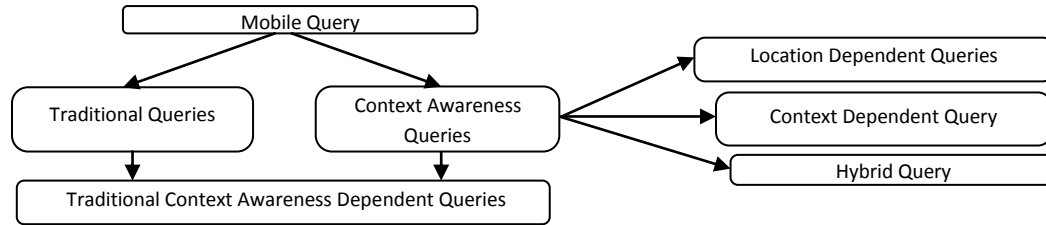
This type of queries involves computing context, user context, or physical context of information. Context awareness queries relates to who, when, where, and what and create new class of applications in mobile computing environment. With context awareness, a mobile device is expected to adapt constantly to a wide range of dynamically and continually changing situations. It is important for the device to be able to aware of the situation, environment, and tasks that the mobile client is currently performing, as well as those that will be performed in the near future. The utilization of context information in an application minimizes the amount of user involvement in a service by providing related information.

Knowledge of the context of the query enables the device to pre-fetch all data that are likely to be queried in the near future. With this query, the client initiates only a single request and all related data will be retrieved implicitly. These techniques prevent a client from making multiple requests and improve energy efficiency. For instance, let us consider how to find information about restaurants in a particular area. The query will result in retrieving information about restaurants based on the user's preferences for Italian, Chinese or continental food, for instance, as well as pre-fetching maps, traffic and weather conditions, which may be queried next. Mobile queries can be classified into location-dependent queries, context-dependent queries, and hybrid queries which is a combination first two.

### 4.3. Location-dependent Queries

Location-dependent query are queries that begin by mobile clients. In this type of queries the location of the mobile client is main parameter. The value of the location parameter is provided explicitly by the client or implicitly using a global positioning system (GPS). An example of this query is a traveler's information system that provides information to tourist about hotels, restaurant, and bars depending on their current location. Processing of this query must be based on knowledge of the user's location. For queries that require services are ahead of the current route location, direction of motion and speed must of necessity be taken into account. Current technology presents a new feature in automobiles, which provides navigational relieves as a built-in feature. In common, every location update generates two direct costs, particularly transmission cost, which refers to the cost to inform the server of the new location, and server processing cost, which corresponds to the cost of updating the system having the location of the mobile unit. The location parameter can be in any vehicles, such as buses, trucks, and airplane. Location-dependent queries can be divided into two categories. The first category is depending on user and object types, and the second is depending on query types. User and object types represent the state of the mobile user when issuing the query and the searched object. The query types category relates to the state of the queries whether continuous or

non-continuous queries. A continuous query is very much different from conventional or non-continuous queries, which



**Figure 2: Query Framework in Mobile Environment**

notification of the updated information will be sent automatically as clients move to different regions. These two categories are described as follows:

### (A) User and Object Types

There are three types of location-dependent query in this category: (i) moving user seeking static object (ii) moving user seeking moving object and (iii) static user seeking moving object.

- *Moving User seeking Static Object (MUSO)*

Moving User seeking Static Object query originates from a moving user and the searched object is static. Examples of this query are:

- Request of information by motorist on highway to know about petrol pump with in distance of 5 kms; strictly on highway side.
- Request of information by motorist on highway to know about multi cushion within the distance of 10 kms.

- *Static User seeking Moving Object (SUMO)*

This type of query originates from a stationary user and the searched object is moving. Examples of this query are:

- Request of information for a particular trains arriving on Jaipur junction with in half hour.
- Request to retrieve information about VT-BSB aircraft during flight training at Banasthali Aviation Control Room.

- *Moving User seeking Moving Object (MUMO)*

This type of query originates from moving user, and the searched object is also moving. Examples of this query are:

- Request to retrieve information about VT-CVI plane by crew members of VT-BSB aircraft when both are in flight.
- A flying helicopter requests when it intercept a friendly tank.

Location-dependent queries in a mobile environment are very much centered on the above three types of queries. The related issues will become more complex when they involve the client's disconnection. In mobile environment client get frequently disconnected from server either due to poor network or to save cost, therefore, it is common for a mobile client to become disconnected from the network after sending queries. Location-

are based on an instant of the database at some moment in time. With this query, clients need to send a query only once and

dependent query requires location-dependent data. Location-dependent data can be categorized into static location-dependent data and dynamic location-dependent data. Static location-dependent data relates to the stationary object, when the object location is not continuously updated in the database. Generally, the "moving user seeking static object" query type falls within this category. On the other hand, dynamic location-dependent data correspond to a moving object that requires constant update of the object's location in the database. The "moving and static user seeking moving object" query type normally corresponds to dynamic location-dependent data.

### (B) Query Types

Location-dependent queries can also be classified based on the query type namely; continuous query, and non-continuous query.

- *Continuous query*

The continuous query includes real-time monitoring of mobile objects. Real-time monitoring queries are continuous for monitoring purposes. In a continuous query setting even after the initial query is answered and the query is still kept by the server. Hence, in the case where the user moves into a different location or when new information becomes available, it will then need to be dispatched to the user. Example: (i) To request information about nearby tourist attractions, hotels, or shopping centre while traveling. With this class of query, clients need to send a query only once and notification of the updated information about nearby tourist attractions, hotels, or shopping centre will be sent automatically as clients move to different regions. (ii) To notify mobile clients whenever they are close to a certain situation such as dangerous zone or traffic jam by providing some form of alerts to them.

In this case, the system must be able to provide the accurate query results and update them in real time whenever mobile clients enter or exit the region defined by the query. This class of query can be referred as range-monitoring queries [9]. The range-monitoring queries are removed from the system only when the user explicitly ends the query.

- *Non-continuous query*

The non-continuous location-dependent query is different from the continuous query as the system does not manage the query,

but only the location of each mobile unit in specified regions. This class of query does not have to specify when to end the query since the server does not keep the query, but location only. Examples: retrieve nearby hospital, police station or petrol station in the area.

#### **4.4. Context-dependent Queries**

These types of query require maintenance of an interior representation of user's needs, preferences, roles, profile, etc. With these parameters, application will be able to decide the situation and task that a user is currently performing, and adapt to changes in user needs and roles. This enables a mobile device to not only retrieve the required information, but also some other information that is highly relevant [17]. Examples: (i) a tourist wants to see a list of restaurants in a town. Rather than retrieving the entire list of restaurants, the query retrieves the restaurants based on the user's preferences and needs such as cuisine preference, price, occasion, etc. (ii) a business woman wants to check the closest meeting schedule. The query retrieves the time as well as the previous history of the meeting, people involved, client accounts, meal preferences, required data etc.

#### **4.5. Hybrid Queries**

Location- and context-dependent queries require the system to maintain all parameters of context-awareness queries including location parameter. Examples: (i) A traveler wants to find restaurants in the region that suits his/her taste within a travel distance of 10 minutes from current position; (ii) a person wants to check movies in the nearby cinemas that s/he will like, and whether or not s/he will be in time for the show, given travel time and distance. These queries are very useful for nomadic users since it not only considers the users' preferences and needs, but also the location of the user. A more sophisticated application utilizes a variety of sensors to monitor the environment as well as user's actions in order to provide assistance with the tasks being performed by the user. This application requires the ability to process data streams in real-time, and to analyze and interpret it precisely. The main issue in context-awareness query is to accurately analyze the environment and the user's intention, regardless of the source of the context information. It is a difficult challenge since there is the possibility of conflicting data, and the need to have efficient processing to provide a useful application to the user.

### **5. QUERY PROCESSING FOR WIRELESS**

In mobile databases, queries can be processed mainly using three different mechanisms: (i) on-demand mechanism (or also called pull operation), (ii) data broadcasting mechanism (or also called push operation) and (iii) hybrid broadcasting mechanism (combination of on-demand and broadcast mechanism) as illustrated in Figure 3.

The mobile client generate query to obtain data from repository at the remote stationary server. Data management strategies in the on-demand mechanism refer to the optimization method used at the server side to serve an on-demand request or a request that is sent to the server for processing. Data broadcasting strategy relates to determining a method to disseminate the database item to mobile client so that the response time, tuning time and power utilization of retrieving database items are minimized.

#### **5.1. On-Demand**

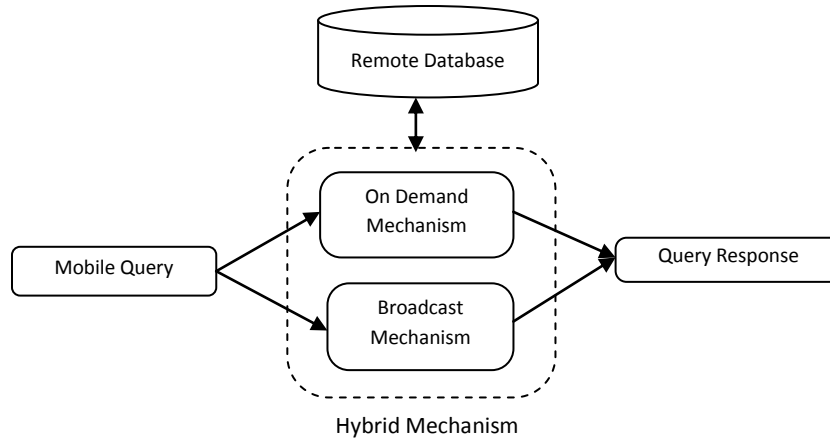
In on-demand broadcast, clients make explicit requests for data. If multiple clients request the same data at approximately the same time, the server may match these requests and only broadcast the data once. On-demand data dissemination is only user-oriented. It provides interactive capability to users for accessing the information through query. Users do not have to search in the wireless information space by tuning several channels. However, this approach has many disadvantages. First of all, it is resource intensive. Users require a separate channel to send requests to the server. The server, after receiving the request, composes the result and sends it to the user on a back channel (downstream) known to the user. Thus, every pull needs two channels for completing the process. Moreover, since incoming requests are usually not identical, the server cannot always efficiently group requests in order to exploit the advantages of broadcast. Obviously, this depends on the volume and the context of the incoming workload. To make things worse, client-server architectures are notoriously not scalable. When the number of incoming requests becomes too high, the server fails to keep up.

#### **5.2. Push-Based**

In push-based systems, the server employs point-to-multipoint communication and sends data items in the absence of explicit client requests. In order to achieve that, the server maintains a broadcast schedule, which determines the order and the frequency in which data items are broadcast. Let the scheduler handles three data items (A, B and C), out of which B and C are broadcast with the same frequency and A twice more frequently, resulting in the transmission schedule: (A, B, A, C, A, B, A, C ...). The major feature of such systems is scalability. Client population does not influence the dissemination process because clients do not issue requests. The additions of new clients do not influence the server's incoming load or the client perceived access time.

In addition to that, clients need few resources such as air indexing enabling clients to efficiently locate data in the broadcast channel. Moreover, data can be kept properly, since the server can simply broadcast any updates. The major problem of push-based systems is their lack of self organization and adaptiveness. Since the server does not receive explicit client requests, it remains unaware of possible changes in client

population or querying characteristics. This incurs several problems. Bandwidth for instance, can be unnecessarily utilized for a relatively low number of end clients. Apart from that, the push service requires more powerful hardware.



**Figure 3: Query Processing in Mobile Computing Environment**

### 5.3. Hybrid data dissemination

As the name suggests, hybrid data dissemination is a combination of on-demand and push-based approaches. Data items are classified into popular and unpopular (also called hot and cold). Hot data items are delivered via push-based channels, while cold data items are disseminated via on-demand channels. The hybrid data dissemination requires proper document classification and bandwidth division for data dissemination. These are interrelated issues, simply because a given bandwidth division determines the performance of a document classification choice and, conversely, a given document classification determines a bandwidth split that optimizes performance [18]. In turn, both document classification and bandwidth division depend on the popularity of data items because download latency is smaller when hot items are assigned to multicast push, cold items to unicast pull, and the bandwidth is divided appropriately between the two channels.

### 5.4. Broadcast Algorithms

To partition data among on-demand and broadcast channel various algorithms have been developed. We study a wide range of broadcast algorithms for query dissemination in two major categories: pull and push algorithms.

#### (A) Pull Algorithms

In a broadcast pull environment, responses to requests are transmitted on a dedicated shared broadcast channel. Following are the specific algorithms that we study:

**First Come First Served (FCFS)** [19] - It broadcasts the data items in the order they are requested. To avoid redundant

broadcasts, a request for a data item that is already in the queue is ignored.

**Longest Wait First (LWF)** [20] - This schedules the data item that has the largest total waiting time, that is, the sum of the time each pending request for the item have been waiting for. We implement LWF using three values for each unique item requested: (i) the number of outstanding request for the data item, (ii) the last recording of the total waiting time and (iii) the time of this recording. For instance, at time 0, when a data item is requested for the first time, the three fields of its service queue entry are set to (1, 0, 0). At every subsequent request at time 't', this entry gets updated. We repeat the update process for every new coming request and also at the time of scheduling decision to avoid using stale total waiting time values.

**PULL RxW** [21] - This algorithm selects the data item with the maximal RxW value where R is the number of outstanding requests for that item and W is the amount of time that the oldest outstanding request for that item has spent in the service queue. Using two threaded lists that keep the service queue entries sorted in R and W order, the scheduling decision time is significantly reduced.

**PULL RxW.a** [21] - It is modified version of RxW algorithm and reduces the scheduling decision time further. The algorithm broadcasts the first item it encounters whose RxW value is greater than or equal to  $\alpha$  times the current threshold value. The threshold is computed as the running average of the RxW value of the last item broadcast and the previous threshold. After each broadcast decision, the threshold is updated accordingly. The scheduling overhead decreases as alpha decrease. The most commonly used settings for the approximation algorithm are PULL RxW.90, ( $\alpha = 0.9$ ), and PULL RxW.0, ( $\alpha = 0$ ). RxW.0 compares only two items, the item with the most outstanding

requests and the one that has been waited for the longest time. While RxW.90 compare every item to be broadcasted.

### **(B) Push Algorithms**

The fundamental push algorithms are Broadcast Disks [5] and Teletext System [22]. In these schedules all items being accessed are partitioned into a number of disks. Items on the smaller numbered disks are broadcast more frequently than those on the larger numbered disks. We use the optimized parameters for Broadcast Disks assuming perfect knowledge of access probabilities.

**PUSH MAD** [5, 23] - It selects the data item whose access probability is highest since last broadcast. In the extreme case, MAD requires examining all items in the database before making a scheduling decision. In the experiments MAD is run twice; the first run records the schedule generated and the second run use this recorded schedule for taking measurements. This results in an optimistic off-line algorithm with the highest throughput.

**PUSH LB (Push Lower Bound)** [5, 22, 23] - The optimum schedule for push is defined by Ammar et al. as follows: (i) popular items should be broadcast more often than their not so-popular counterparts, (ii) two successive broadcasts of the same item should be equal distance apart throughout the schedule and (iii) the relative number of appearances of items on the broadcast channel should be in the square root ratio of their access probabilities. In other words the items have to be broadcasted in ratio of their access probabilities. In order to calculate the theoretical lower bound we assume that full bandwidth and an ideal allocation is possible. In practice, it is not always possible to generate such a schedule due to timing conflicts, that is, two pages need to be broadcast at the same time. It should be noted that we assume perfect knowledge of access probabilities when determining push schedules.

## **6. CONCLUSION**

Recent emergence of wireless technology enables people to conduct activities, business, or transactions anytime and anywhere without any attachment to stationary computer. Nomadic people are now able to access email, news, weather, and query to the central database server using wireless devices. Mobile database focuses on the query issue that is the dominant operation in mobile computing. Since mobile database is a new dimension of database application, the type of query, query processing strategy, and communication technology that involves in the application are different than what applies in traditional databases. In this paper, we first discussed the issues of query selection, taxonomy and query processing strategies. We have defined query taxonomy as well as query processing strategy in mobile databases. Query in mobile databases are categorized into context-awareness query and ad-hoc query. Context-awareness query is further classified into location

dependent, context dependent, and hybrid query. Query processing in mobile databases includes mobile client, on air, and server strategy. Mobile client and on air strategy corresponds to caching strategy and broadcast strategy respectively. This work presented query broadcast management schemes to optimize and minimize the query access time of mobile clients when retrieving broadcast database items. Here we have addresses to three questions: (i) what are the varieties of queries invoked in wireless system and how they differ from traditional system? (ii) What server does to answer these queries efficiently to MC? (iii) What special and temporal strategies can be applied to manage air channel for data dissemination? Section 3 has answered first question, section 4 answered to second question and section 5 answered to third question.

## **7. ACKNOWLEDGEMENTS**

Savita Kumari contributed this manuscript when she was working with Faculty of Engineering College, University of Seventh April, Zawia (Libya) – 16418. She thanks faculty and staff there for extending help to prepare this paper.

## **8. REFERENCES**

- [1] WAP Forum, Wireless Application Protocol Downloads <http://www.wapforum.org>.
- [2] Barbara D., "Mobile Computing and Databases- A Survey", IEEE Transactions on Knowledge and Data Engineering, pp. 12(1), 118-127, 1999.
- [3] Malladi R. and Davis K.C., "Applying Multiple Query Optimization in Mobile Databases", In Proceedings of the 36th Hawaii International Conference on System Sciences, pp. 294 – 303, 2002.
- [4] Imielinski T., Viswanathan S. and Badrinath B. R., "Energy Efficient Indexing on Air", In the proceeding of the ACM Sigmod Conference, pp.25-36, 1994.
- [5] Acharya S., Alonso R., Franklin M. J. and Zdonik S. B., "Broadcast Disks: Data Management for Asymmetric Communications Environments", In the proceeding of the ACM International Conference on Management of Data, pp. 199—210, 1995.
- [6] K. C. K. Lee, H. V. Leong, and A. Si, A semantic broadcast scheme for a mobile environment based on dynamic chunking, in Proceedings of the 20th IEEE International Conference on Distributed Computing Systems (ICDCS'2000), pp. 522–529, Taipei, Taiwan, April 2000.
- [7] Waluyo, A.B., Srinivasan, B., and Taniar, D., "Optimizing Query Access Time over Broadcast Channel in Mobile Databases", In Proceedings of the Embedded and Ubiquitous Computing (EUC'04), Lecture Notes in Computer Science, vol.3207, Springer-Verlag, pp. 439-449, 2004.

- [8] D. Pfoser, C.S. Jensen, and Y. Theodoridis. Novel approaches in query processing for moving objects. In Proceedings of the 26th International Conference on Very Large Databases (VLDB), Cairo, Egypt, September 2000.
- [9] Cai Y. and Hua K. A., "An Adaptive Query Management Technique for Real-Time Monitoring of Spatial Regions in Mobile Database Systems", In the proceeding of 21st IEEE International Conference on Performance, Computing and Communications, pp.259-266, 2002.
- [10] Sakata T. and Yu J. X., "Statistical Estimation of Access Frequencies: Problems, Solutions and Consistencies", ACM/Kluwer Wireless Networks, 9(6), 647-657, 2003.
- [11] Yu J. X., Sakata T. and Tan K.L., "Statistical Estimation of Access Frequencies in Data Broadcasting Environments", ACM/ Baltzer Wireless Networks, 6(2), 89-98, 2000.
- [12] Nikipolitis P., Papadimitriou G. I. and Pomportsis A. S., "Using Learning Automata for Adaptive Push-based Data Broadcasting in Asymmetric Wireless Environments", IEEE Transactions on Vehicular Technology, 51(6), 1652-1660, 2002.
- [13] Stathatos K., Roussopoulos N. and Baras J. S., "Adaptive Data Broadcast in Hybrid Networks", In the proceeding of the 23rd International Conference on Very Large Data Bases (VLDB), 326-335, 1997.
- [14] Zipf G. K., "Human Behavior and Principle of Least Effort: An Introduction to Human Ecology", Addison-Wesley, 1949.
- [15] Breslau L., Cao P., Fan L., Phillips G. and Shenker S., "Web Caching and Zipf-like Distributions: Evidence and Implications", In the Proceeding of IEEE INFOCOM, 1999.
- [16] MATLAB - The Language Of Technical Computing, <http://www.mathworks.com>.
- [17] Franklin M. J., "Challenges in Ubiquitous Data Management", Informatics, pp. 24-33, 2001.
- [18] Vaidya N. H., and Hameed S., "Scheduling Data Broadcast in Asymmetric Communication Environments", ACM Wireless Networks 5(3), pp. 171–182, 1999.
- [19] Dykeman H., Ammar M., and Wong J., "Scheduling Algorithms for Videotext Systems under Broadcast Delivery", In the proceeding IEEE International Conference on Communications, Toronto, Canada, 1986, pp. 1847–1851.
- [20] Dykeman H. and Wong J., "A Performance Study of Broadcast Information Delivery Systems", In the proceeding of IEEE INFOCOM, 1988.
- [21] Aksoy D., Franklin M., and Zdonik S., "Data Staging for on-demand Broadcast," In the proceeding of International Conference on Very Large DataBases (VLDB), 2001.
- [22] Ammar M. H. and Wong J., "The Design of Teletext Broadcast Cycles", Performance Evaluation 5(4), pp. 235–242, 1985.
- [23] Su C. and Tassiulas L., "Broadcast Scheduling for Information distribution", In the proceeding of IEEE INFOCOM, 1997.