Journal of Scientific & Industrial Research
Vol. 63, November 2004, pp 877-887

# An approach towards the query handling in mobile cellular network

S Mitra and S Das*(Bit)

Department of Computer Science & Technology, Bengal Engineering College (D U), Howrah 711 103

*Received 30 December 2003; rev recd 21 July 2004; accepted 28 September 2004*

A data management scheme for processing different types of queries in a mobile cellular environment is proposed. The present work is able to give answers of different types of queries such as, proximity query, personal query, and query for any broadcast type of data by exchanging a few number of messages among different components of the network such as, mobile unit, base station, and mobile switching centre. The scheme uses two databases stored at base station and mobile switching centre. Locations of all location dependent data are represented in cylindrical polar coordinates, and the database at each base station keeps the data of the fixed objects exclusively in its locality. As a result the proximity query raised by a mobile unit can be answered by its servicing base station without consulting other base stations and mobile switching centres, thus reducing message exchange and hence the delay in query answering. The suitable file organization to support the database is also proposed here. Moreover, a suitable Query Language is proposed here to process the queries in the mobile cellular environment. Exhaustive simulation work is done to study the performance of the system for getting satisfactory answer of the query by varying different parameters like, query arrival probability and time.

**Keywords:** Cellular mobile architecture, Base station, Mobile switching centre, Query processing

**IPC Code: Int. Cl.$^7$:** H 04 L 12/28

## 1 Introduction

The continuous growth of mobile users in a cellular network should provide the facility of accessing different types of information to its users. The cellular mobile environment[1,2] is a collection of geometric areas called cells. Each cell is serviced by a base station (BS) at its centre (Fig. 1). Several BSs are again linked to a mobile switching centre (MSC). The MSC acts as a gateway of the cellular network to the existing wired network like, PSTN and ISDN. The wireless communication takes place only between BS and mobile units (MU). The communication between BS and MSC or MSC and MSC are wireline.

Data management in the area of personal wireless communication is a new challenge. In view of the continuous growth of mobile users in a cellular network the facility of accessing different types of information[3,4] should be provided to its users. Handling of data and information in mobile environment raises various issues such as, query processing for the access of some common information by mobile users. Mobile users may need to access frequently and quickly some data which can be catered or broadcast on demand basis. For example, traffic condition of a place or weather condition of a district may be broadcast, whereas answer of a question like “where is the nearest hospital?” may be catered on demand basis.

The present work is able to give answers of different types of queries by exchanging a few number of messages between MU and BS and between BS and MSC. A database is maintained at each BS and MSC (Fig. 2 and 3). The database at each BS contains its own data in addition to the data about the objects in the adjacent BSs, whereas all the MSCs maintain identical information to process personal and broadcast queries. The scheme is able to

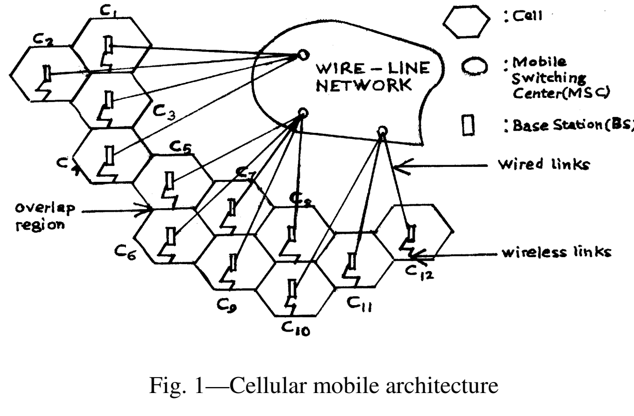


Fig. 1—Cellular mobile architecture

*Author for correspondence
e-mail: sdasbit@yahoo.co.in

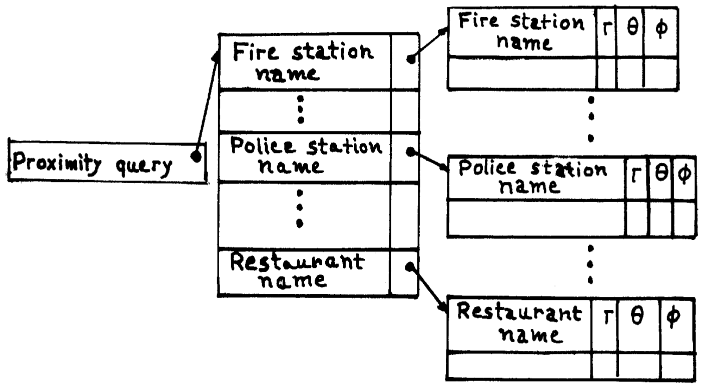


Fig. 2—Database at BS

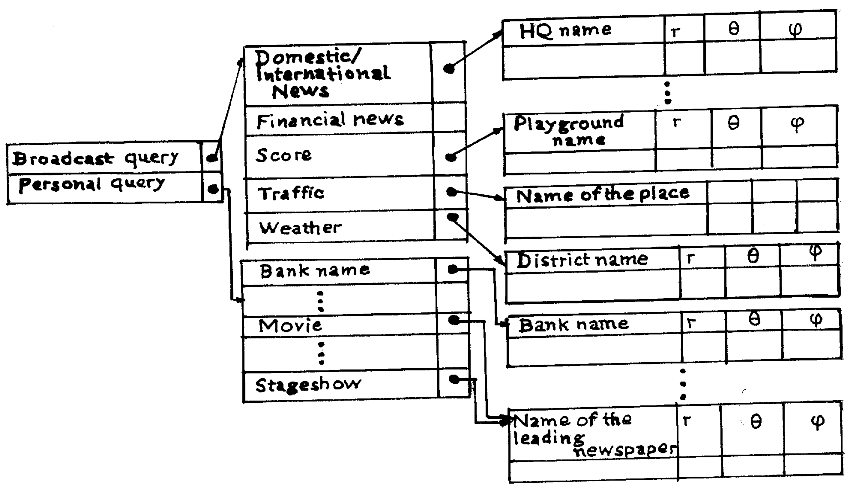


Fig. 3—Database at MSC

give answer of proximity queries. It is also able to give answer of any broadcast type query like, weather report, traffic condition, and score of a match at any playground as well as the query for any personal data like balance in one's bank account, and sales figure. Out of these the query for sales figure may be an example of movable query source and movable reply end, whereas for the other queries in this scheme query source is movable but the reply end is stationary. In the proposed work, an MU sends a query to the current BS. If the query is proximity type the BS searches its database for the desired object. If the result of search is not null, BS computes the shortest distance among the distances between the MU and each of its search result and informs the MU accordingly about its next direction of movement. Otherwise BS computes the nearest adjacent cell, sends the MU's query to that cell and informs the MU about its next direction of movement. But for the query of personal or broadcast data, BS searches its memory for the information related to the query. If the result of search is not null, BS reads the answer and sends the answer in turn to the MU. Otherwise, BS sends the query to its parent MSC. The MSC then searches its database for the location of the desired object. If the result of search is not null, MSC reads the answer of the query from the location corresponding to the search result and sends the answer to the MU through BS. Otherwise the answer of the query is not found and hence cannot be communicated.

## 2 Previous Work

Very few works on data management for query answering are so far been done. The issue of distributed location service for tracking mobile users has first been addressed by Awerbuch and Peleg[5,6]. Here the directory server supports "find" and "move" operations. A hierarchical directory structure that minimizes the cost of a sequence to "find" and "move" operations is presented. For "find" operation, nearby MUs are able to find the exact location but MUs which are far away will have to follow a chain of pointers to find the exact location. Further, on a "move", only nearby directories are updated, so that only approximate information about MUs location is maintained.

In another work[7], as an improvement of the above scheme, there is a hierarchy of location servers (LS) which are connected among themselves and to the BSs by a standard network. LSs are responsible to keep track of addresses of MUs who are currently residing in the area "below" the LS. Each MU is permanently registered under one of the LSs; additionally the MU may also register as a visitor under some other LS. BS is always be aware of mobile terminals which are active within their cells. LSs, however, do not have to know in which cell a given mobile terminal is currently located. They can always find out by means of paging to a set of BSs under the given LS. Since the LS needs to contact at BSs over the fixed network the cost of broadcasting is equal to the number of BSs to which paging requests are sent. Importantly the database almost never store the actual location of an MU. Either some outdated previous positions or a set of possible locations is stored. Therefore, it almost always stores incompletely specified location data to avoid massive and frequent location updates which, in turn, increases the paging cost due to paging in a number of probable cells. But in this scheme the method of accessing the desired information from remote resources is not considered.

To minimise the paging cost, a profile based scheme[8] is proposed. In this scheme usage of bandwidth is reduced by avoiding unnecessary updation/paging. The system cost is also reduced by selectively choosing the MU's profile for updation, based on dynamic categorization of the MU periodically. But each MU should know its profile in this case. All the above schemes are able to locate the desired MU either within a cell or within an LA consisting of a few cells. But with the advent of cellular network and increasing number of users in this environment the demand of different services are growing day by day. For many of such services, it is required to identify the exact location of a mobile user within a radius of few meters or so. In the scheme[9], a location management scheme is proposed to determine the exact location of the mobile user. The exact location is needed in the case of any emergency such as, fire and accidents. Other than the emergency the computation of exact location is very much useful for processing of proximity query.

The schemes discussed so far are deterministic in nature. The deterministic approaches assume obviously that either an MU is present in a cell or absent in a cell. Thus the deterministic approach considers two extreme cases. But in real life as an individual MU follows its own pattern of movement the said assumption may not be much realistic. Rather the probability value of an MU to be in a cell may be anything between the two extreme cases. Such distribution helps to improve the result of decision making process.

There are two different probabilistic location update schemes[10,11] where Gaussian distribution is considered for MU location. In the first scheme[10], MSC maintains a logical structure of LAs which consists of cells under it. Each MSC also maintains an indexed data to keep the frequency of traversals of all the MUs in the cells under it. As soon as a call arrives for a particular MU, MSC computes the location probability of that MU in all the cells under it and finds the best probable cell(s) where the MU may be found using best first search or AND-OR best first search. This existing probabilistic scheme is made more realistic in the second probabilistic scheme[11] considering the fact that most of the users are moving around a few cells. The second scheme is a distributed probabilistic location management scheme, where the task of keeping track of an MU is shared by BSs and MSC. The BSs always update each visiting user's information and periodically send this information to the MSC. The MSC in turn captures movement pattern in terms of location probability. The search for a called MU is performed along a hierarchically represented search space maintained by MSC. This search space is continuously updated by the current location probability computed at MSC. Here, Gaussian distribution is appropriately used to simulate the real life movement pattern of the MUs. The performances of the various deterministic and probabilistic location update schemes are compared in a survey paper[12].

A number of works[13,14] on query answering of location dependent data (LDD) in cellular network are so far been reported. A LDD query processing scheme is proposed by Xu and Lee[13] using data replication and caching in server and MU. The demerit of the scheme lies on the fact that, data replication needs to keep the replica up-to-date at any time. The replication at MU would be especially difficult as it can go into disconnected state. At the same time, although caching at MU can save power due to less data transmission, the problem is to maintain cache consistency.

In another work[14] also the researchers used semantic cache scheme for improving data reusability. They used Voronoi diagram to construct an index to facilitate searching. Whenever a query is submitted to the server, the user profile is also sent. As the profile contains velocity of the MU, the system can predict the time when the MU will cross the region where from the query is raised. Accordingly the answer of the query is calculated and sent to the MU.

To access a wide range of information from remote resources, mobile units inevitably depend on remote resources[15,16]. In the work of Imielinski and Badrinath[15], a MU communicates with a BS or server via a wireless network in order to access files/data. In addition the BS may require to access locally unavailable files/data from other remote BSs. Mobility makes certain data dynamic and it may be desirable to have the distributed directories which move from one BS to another as MUs shift their geographic location. But to access remote resources a large number of message exchange is required.

Kumar *et al.*[16] proposed a dynamic distributed directory scheme (DDS) to facilitate efficient and transparent access to information files in mobile environments. The DDS comprises a hierarchical directory that enables efficient search of requested files. The DDS keeps track of update versions of files. Once connection is established, the MU transmits a request for a file. Otherwise, it searches the address of the remote BS that contains the required file, the BS initiates transfer of the required file from the remote station to the current BS via the shortest available path on the wired LAN. Thus, in this scheme, it is possible to access locally unavailable files/data from other remote BSs at the cost of huge message exchange.

The authors[17] proposed a *n* level hierarchy (Fig. 4) of location servers (LSs) where LSs are connected

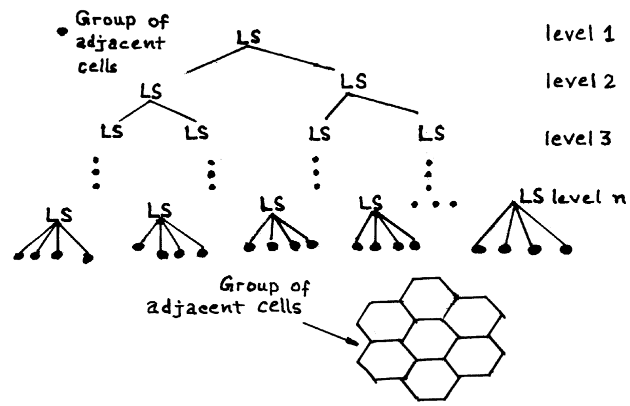


Fig. 4—Architecture of cellular network

among themselves and to the BSs by a standard network. Each LS and BS contains database having some fixed records. Root level LS (level 1) contains the information available in the LSs at level 2. The LSs at level 2 in turn contain all the information available in the LSs at level 3 and so on. LSs at leaf level (level *n*) are connected with the BSs. Each of the LS at level *n* contains the information which is available in the BSs connected with them. The scheme is able to give answer of any proximity query like “where is the nearest hospital?”. It is also able to give answer of personal query like balance in one's bank account, sales figure, etc., as well as the query for any broadcast type of data like the score of a match at a playground, etc. When an MU sends a proximity query along with its coordinate to the host BS, the BS searches its database for the desired object. If found, it computes the shortest distance among the distances between the MU and the search result. Finally the host BS sends the answer of the query to the MU. But if the result of search is null, host BS computes the shortest distance among the MU and its six adjacent cells. Accordingly it informs the MU to move towards the nearest BS to get the desired answer of the query. The new host BS performs the same operation as the previous host BS. In case of personal and broadcast query, the host BS first searches its database for the answer of the query. If the result of search is not null, BS retrieves the answer of the query from the database and sends the answer to the MU. Otherwise BS is able to reply to the user's query after consulting the database stored at its parent LS. But in this scheme as the information required to process such query remains distributed among the LSs at various levels, there is a considerable amount of delay to process a personal and broadcast query.

On the contrary, in the proposed scheme the information required to process the personal and broadcast query is maintained by MSC. So for such queries, BS has to contact with the parent MSC only resulting in a single database search. Consequently the delay to process a personal and broadcast query is reduced drastically. So the present scheme is basically an extension/modification of our previous work[17].

## 3 The Present Scheme

Not much work has been done and reported on query processing in cellular mobile environment so far. But most of them maintain an approximate information about MU's location which increases the paging cost. MUs are able to access locally unavailable files/data from other remote BSs at the cost of huge message exchange. The proposed scheme is able to give answer of different types of query by exchanging a less number of message among the various component of the network.

### 3.1 Data Management

In the proposed work, each MSC keeps some fixed information to process the personal as well as broadcast queries, whereas each BS maintains a database containing various locations of fixed objects to process proximity query.

#### 3.1.1 Database at BS

Each BS contains the name of the various fixed objects like hospital, restaurant, and police station along with their $r$-θ-ϕ coordinates to process proximity query. The database at each BS contains its own location dependent data in addition to the data in adjacent cells (Fig. 2).

#### 3.1.2 Database at MSC

Each MSC contains all the information required (Fig. 3) to process broadcast and personal query. It contains the name of the headquarters to broadcast various news such as, domestic or international news, stock exchange report, and weather report of a district along with their $r$-θ-ϕ coordinates. It also contains data such as the name of the bank along with the $r$-θ-ϕ coordinates of its headquarter to process personal query.

### 3.2 Query Processing

In this section the processing of various queries is discussed. The steps of operation to process various types of queries are as follows:

#### 3.2.1 Processing of Proximity Query

In this case the query source is movable, whereas the reply end is stationary. After receiving such a query, BS searches its database for the desired fixed object. If the result of search is not null (the desired object may be in current BS of MU or in at least one of its adjacent cells), BS computes the shortest distance among the distances between the MU and each of its search result. Moreover, BS computes the new direction of movement of the MU (Fig. 5) as $\tan^{-1}(\pm\theta)$, where $\theta = (r_r\sin\theta_r - r_s\sin\theta_s)/(r_r\cos\theta_r - r_s\cos\theta_s)$ and the sign depends on the position of the MU and the desired nearest object. Let the point A be the position of the nearest desired object having coordinate $(r_r, \theta_r)$ and the point D be the position of the MU having coordinate $(r_s, \theta_s)$. The new direction of movement of the MU is along DA and it is equal to $\varphi = \tan^{-1}(r_r\sin\theta_r - r_s\sin\theta_s) / ( r_r\cos\theta_r - r_s\cos\theta_s)$. Finally, BS guides the MU about its next direction of movement. If the result of search is null, BS computes the shortest distance among the distances between the MU and all of its adjacent cells', replies to the MU about its next direction of movement towards the nearest adjacent cell. BS sends the query to the nearest BS through the parent MSC. The new BS then repeats the same procedure.

#### 3.2.2 Processing of Query for Personal Data

First the movable query source and the stationary reply end is considered. When the query is raised for some personal information say the balance of a user's bank account, BS searches its memory to find out whether the same query is processed in recent time and hence the result may remain stored in memory. If the result of the search is not null, BS reads the answer and sends it to MU. Otherwise, BS sends the same query to its parent MSC**.** The MSC searches its database for the desired object. If the result of search is not null, MSC reads the $r$-θ-ϕ coordinates

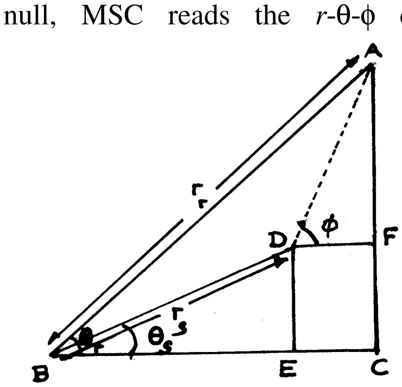


Fig. 5—Calculation of the new direction of movement of the MU

corresponding to the desired object, reads the result from that location and sends the answer to the host BS. The BS conveys the answer to MU. Otherwise the answer cannot be communicated.

Secondly the query source and the reply end, both are considered as movable. The architecture (Fig. 4), as proposed in the work[17], may be considered in the present scheme to process this particular query. This architecture helps to use a standard location management scheme using distributed searching[18] to locate the desired reply end which works better than any centralized location management scheme[19,20].

In our previous work[17], as described in Section 2, an *n* level hierarchy of LSs are considered to process the query when the query source is movable and the reply end is stationary. This processing needs a huge amount of message exchange and delay to process the query as the information required to process such query are distributed among the various levels of LSs. Whereas, in the present scheme the information required to process such query is available in the MSC which reduces the number of message exchange as well as delay to process such query.

#### 3.2.3 Processing of Broadcast Query

In this case the query source is movable, whereas the reply end is stationary. MU sends the query to BS. The BS searches its primary memory for the answer of the last few queries. If the answer of the current query is available at primary memory, BS reads the answer and sends it to MU. Otherwise, BS sends the same query to its parent MSC. The MSC searches its database for the desired object. If the result of search is not null, MSC reads the $r$-$\theta$-$\phi$ coordinate, corresponding to the desired object, reads the result from that location and sends the answer to the host BS. The BS convey the answer to MU. Otherwise the answer cannot be communicated.

Just like processing of personal query, a very few number of message exchanges and less amount of delay are required in the present work compared to our previous work[17].

### 3.3 Query Language

In this section the presentation strategy of various queries of MU has been discussed. In the present work the query is formulated, using modified QBE (Query By Example) where the MU specifies its query by filling in certain columns of the template (Fig. 6) that are displayed on the screen of MU. To send the query the mobile user enters the type of the query, name of the search key and parameters (if any) in the template. For example, in the case of personal query the parameters may be the account number of the user for the first case or MU_id of the reply end for the second case as discussed in Section 3.2.2.The system processes the query as discussed in Section 3.2 and displays the answer of the query in the template. The modified QBE has all the advantages as in QBE.

| Type of query | Search key name | Parameter | Result of query |
|---|---|---|---|
| | | | |

Fig. 6—Template for specifying the query in the modified QBE

It is to be noted that the similarity of modified QBE with QBE lies on the fact that user does not have to specify a structured query explicitly; rather the query is formulated by filling in templates. As far as modification is concerned, in QBE general purpose queries are processed and the answer of the query comes from different tables designed beforehand. As a consequence user has to choose the table names and has to give the predicates. On the other hand, in this modified QBE, user neither has to choose any tables nor has to specify any predicate. Here depending on query type, either the table at BS or the table at MSC will be searched for getting the answer. Moreover as the queries are restricted to proximity, personal and broadcast type, user does not have to give any predicate.

### 3.4 The Algorithm

The algorithm is used to process the various queries in the cellular mobile environment.

```
main()
  {
  MU sends the query to BS;
  BS_id : = current BS of MU;
  MU_location : = position of MU within BS_id;
  desired_obj : = desired search key;
  if (MU's query == proximity type)
     prox (desired_obj,MU_location)
  else
     perbro (desired_obj, MU_location);
}
prox (desired_obj, MU_location)
/*processing of proximity query*/
{
```

```
   BS_id searches its database for desired_obj;
   if found /*within the same BS_id or adj_cell_id*/
   {
      obj_locations : = desired_obj's locations;
      m : = number of successful search result;
      shortdis (MU_location, obj_locations);
      MU moves towards the shortest_location;
   }
else
   {
      obj_locations : = cell_id of the six adjacent cells;
      m : = number of adjacent cells;
      shortdis (MU_location,obj_locations);
      MU moves towards the shortest_location;
      BS sends the MU's query to shortest_location;
      BS_id : = BS_id corresponding to shortest loca-
tion;
         prox(desired_obj, MU_location);
      }
   }
shortdis (MU_location, obj_locations)
/*Shortest distance calculation*/
{
   m : = number of successful search result or number
of adj_cells;
   for search_result : = 1 to m
   {
      BS_id computes the shortest distance between
MU_location and different obj_locations;
      return (shortest_location);
      }
   }
perbro (desired_oj, MU_location)
{
   /*processing of broadcast and personal queries*/
   BS_id searches its memory to find out whether the
same query is
   processed in recent time or not and hence the result
may remain stored in memory;
   if found
   {
      BS_id reads the answer;
      BS_id sends the result to MU;
else
   {
      BS_id sends the query to the parent MSC;
      MSC searches its database for the desired search
key;
if found
   {
      location : = the coordinate of the desired_obj;
      MSC reads the result from the location;
      MSC sends the result to BS_id;
      BS_id sends the result to MU;
   }
else
      answer cannot be communicated;
   }
}
```

#### 3.5 Performance Analysis

In this section the performance of the system has been studied. The number of message exchange and time taken by a BS to give a satisfactory answer of MU's query has been found out. This section also estimates the cost of the system for processing various queries. The notations used in this section are as follows :

$P$ = cost / message in wireless network,
$Q$ = cost/message in wired network,
$n$ = number of BS involved in query processing,
$c$ = cost/byte,
$x_i$ = number of byte in the $i^{th}$ block,
$w$ = data rate (no. of bytes / sec) during transmission / reception in wireless network,
$l$ = data rate (no. of bytes / sec) during transmission / reception in wired network,
$\gamma$ = waiting time of MU in second for connection,
$y$ = length of message transmitted by MU in bytes, and
$q$ = length of message transmitted by BS and answer of any query in bytes.

Here real life data rates ($w$ = 30 kbps, $l$ = 144 kbps) for wireless and wired network are considered.

The reference websites are as follows :
http://www.computerworld.com/mobiletopics/mobile/technology/story/0,10801,87486,00.html
http://www.ee.latrobe.edu.au/~mf/ELE52PMC/Lectures/PMC-06.ppt

##### 3.5.1 Cost Calculation

This section estimates the cost of the system to process various types of queries.

##### 3.5.1.1 Cost for the Proximity Query

MU sends the query to BS (host). BS searches its database. The required number of block access is three. If the result of search is not null and if the desired object is inside the host BS, the host BS reads the answer and sends the result to MU. Hence,

$$\text{cost} = (\sum_{i=1}^{3} x_i) * c + 2P$$

If the result of search is null,

$$\text{cost} = ((\sum_{i=1}^{3} x_i) * c) * c * n + (n-1)*2*Q) + (n+1)P$$

#### 3.5.1.2 Cost for Other Queries

In this case, MU sends he query to BS (host). BS searches its memory for the desired search key. If the result of search is not null, host BS reads the answer and sends the result to MU. Hence, cost = 2*$P$

If the result of search is null, cost = (2*$P$) +(4*$Q$)+ $[(\sum_{i=1}^{3} x_i) * c]$

### 3.5.2 Delay in getting the Satisfactory Answer of MU's Query

This section computes the delay in getting answer of various query as follows :

#### 3.5.2.1 Delay for the Proximity Query

It has two parts, corresponding to the result of search at host BS.

(i) If the result of search is not null,
Connection time = $\gamma$
Time required by the MU to send the query to BS = ($y/w$)
Time taken by each BS to give answer to MU = ($q/w$)
Thus the total delay = $\gamma$ + ($y/w$) + ($q/w$)

(ii) If the result of search is null,
Connection time at each BS = $\gamma$
Time required by the MU to send the query to BS = ($y/w$)
Time for communication of messages among BSs= 2*($n$-1)*($y/l$)
Time taken by each BS to give answer to MU= ($q/w$)
Thus the total delay = ($\gamma$ + $q/w$)*$n$+($y/w$) +2*($n$-1)*($y/l$)

#### 3.5.2.2 Delay for the Other Queries

This delay has two parts which are as follows:

(i) If the answer of the query is within the memory of host BS,
Connection time = $\gamma$
Time required by the MU to send the query to BS = ($y/w$)
Time taken by BS to give answer of MU's query = ($q/w$)
Thus the total delay = $\gamma$ + ($y/w$) + ($q/w$)

(ii) If the memory does not contain the answer of the query sent by MU,
Connection time at host BS = $\gamma$
Time required by the MU to send the query to BS = ($y/w$)
Time for communication of messages among MSC and BSs = 2*($y/l$+$q$/l)
Time taken by the host BS to give answer of MU's query = ($q/w$)
Thus the total delay =$\gamma$ + ($y/w$) + ($q/w$) 2*($y/l$+$q/l$)

### 3.5.3 Message Exchange for Processing Query

#### 3.5.3.1 Message Exchange to Process a Proximity Query

It has two parts, corresponding to the result of search at host BS.

(i) If the result of search is not null,
Number of message exchange :
MU to host BS → 1 (wireless)
host BS to MU → 1 (wireless)

(ii) If the result of search is null,
Number of message exchange :
MU to host BS → 1 (wireless)
host BS to other $n$ number of BSs → *2(n-1)* (wired)
n number of BSs to MU → $n$ (wireless)

#### 3.5.3.2 Message Exchange for Other Queries

It also has two parts, corresponding to the result of search at the memory of host BS.

(i) If the answer of the query is within the memory of host BS,
Number of message exchange :
MU to host BS →1 (wireless)
host BS to MU →1 (wireless)

(ii) If the memory at the host BS does not contain the answer of the query,
Number of message exchange :
MU to host BS →1 (wireless)
BS to MSC → 1 (wired)
MSC to BS → 1 (wired)
host BS to MU → 1 (wireless)

## 4 Simulation Results

In this section the performance of the proposed work is presented. Detailed simulation experiments are carried out for the query processing problem in mobile environment. Experiments are conducted to note the variation of system cost, delay and message exchange in getting answer of query depending upon the time and query arrival probability, which follows poisson's process. Here simulation is carried out in SUN OS platform with RAM capacity 256 MB, Frequency 400 MHz and Hard disk 16 GB.

Table 1 shows the variation of cost with time for three different values of query arrival probability.

Test case 1: Query arrival probability = 0.3. The cost of the system increases with time.

Test case 2: Query arrival probability = 0.6. The cost of the system increases with time but it is lesser than that in Test case 1.

Test case 3: Query arrival probability = 0.9. The cost of the system increases with time but it is even lesser than that in Test case 1 and Test case 2.

Thus the cost of the system increases with time but reduces with the increase in query arrival probability. The graphical representation of the system cost as a function of time for three different values of query arrival probability is shown in Fig. 7.

Table 2 shows the variation of delay in getting answer of query with time for different query arrival probability.

Test case 1: Query arrival probability = 0.3. The delay in getting answer of query from the system increases with time.

Test case 2 : Query arrival probability = 0.6. In this case delay increases with time but it is lesser than that in Test case 1.

Test case 3: Query arrival probability = 0.9. Here also delay increases with time but it is even lesser than that in Test case 1 and Test case 2.

Fig. 8 shows the variation of delay with time for three different values of query arrival probability.

Table 3 shows the number of message exchanges with time for three different values of query arrival probability.

Test case 1: Query arrival probability = 0.3. Both the number of wired and wireless messages increase with time and query arrival probability.

Test case 2: Query arrival probability = 0.6. Both the number of wired and wireless messages increase with time and query arrival probability. It is higher than that in Test case 1.

Test case 3: Query arrival probability = 0.9. Both the number of wired and wireless messages increases with time and query arrival probability. It is higher than that in Test case 1 and Test case 2.

Table 1—System cost as a function of time for variousquery arrival probability

| Query arrival probability | Time (s) | System cost (s) |
|---|---|---|
| 0.3 | 3 | 66 |
| | 6 | 126 |
| | 9 | 150 |
| | 12 | 174 |
| | 15 | 198 |
| 0.6 | 3 | 60 |
| | 6 | 84 |
| | 9 | 108 |
| | 12 | 132 |
| | 15 | 164 |
| 0.9 | 3 | 24 |
| | 6 | 48 |
| | 9 | 72 |
| | 12 | 96 |
| | 15 | 126 |

Table 2—Delay as a function of time for various query arrival probability

| Query arrival probability | Time (s) | Delay (s) |
|---|---|---|
| 0.3 | 3 | 70 |
| | 6 | 125 |
| | 9 | 160 |
| | 12 | 178 |
| | 15 | 200 |
| 0.6 | 3 | 64 |
| | 6 | 88 |
| | 9 | 112 |
| | 12 | 140 |
| | 15 | 160 |
| 0.9 | 3 | 28 |
| | 6 | 52 |
| | 9 | 80 |
| | 12 | 100 |
| | 15 | 130 |

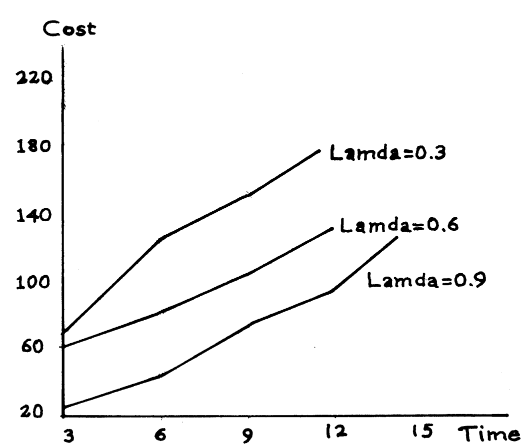


Fig. 7—System cost vs time for different query arrival probability

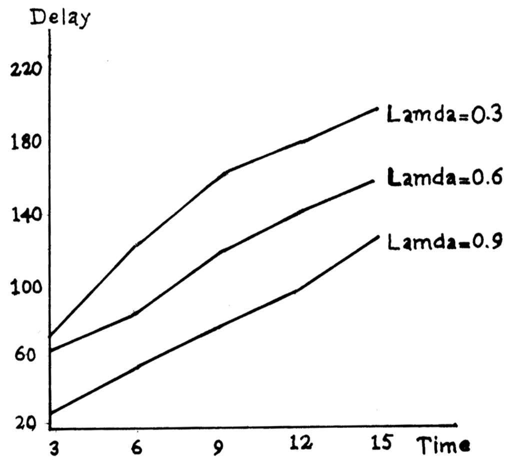


Fig. 8—Delay vs time for different query arrival probability

Table 3—Number of wired and wireless message exchanges to process query

| Query arrival probability | Time (s) | Wired message (Data rate=144kbps) | Wireless message (Data rate=30 kbps) |
|---|---|---|---|
| 0.3 | 3 | 1 | 1 |
| | 6 | 2 | 2 |
| | 9 | 3 | 3 |
| | 12 | 5 | 4 |
| | 15 | 7 | 6 |
| 0.6 | 3 | 2 | 2 |
| | 6 | 5 | 5 |
| | 9 | 9 | 8 |
| | 12 | 14 | 9 |
| | 15 | 20 | 11 |
| 0.9 | 3 | 7 | 4 |
| | 6 | 15 | 9 |
| | 9 | 25 | 15 |
| | 12 | 37 | 22 |
| | 15 | 52 | 31 |

From the results shown in Tables 1, 2 and 3, summarily it can be said that although message exchange increases with query arrival probability, delay and cost reduces. As discussed in Section 3.2.2 and 3.2.3, the reason behind this promising result is that for some of the personal and broadcast queries, the system does not have to process the query afresh; rather it can readily answered from the past answers stored in memory.

## 5 Conclusion

The present work gives answers to different types of queries with the help of database stored, both at BS and MSC. The system is able to give answers of any proximity query as well as the query for any personal data or for broadcast type of data without much delay. An indexed file organization along with a suitable query language is proposed. Exhaustive simulation work in SUN OS platform is done to study the performance of the system.

The present work may be extended in future by capturing the query arrival pattern in the various BSs of the environment, to minimize the number of message exchanges further. Moreover, the BSs where the probability of query arrival is high, may keep the updated information to process the broadcast query which may help to minimize the number of message exchanges and delay in getting answer for such query. With the advent of technological growth resulting in easy and cheap availability of computing power, answering versatile and complex queries is now a practical proposition. Further research must make the system user-friendly so that the question may be posed in a free format. A secure on-line database and an optimized query processing system needs to be developed to realize this potential as a reality.

## References


1 Black U, *Mobile and wireless network* (Prentice-Hall, Englewood cliffs, NJ), 1996.

2 Macdonald V H, The cellular concept, *The Bell Syst Tech J*, **58**(January 1979).

3 Kumar M J, Venkatesh S & Panchanathan S, A distributed directory scheme for information access in mobile computing environments, *IEEE Int Conf High Perform Compu,* (December 1996) 138-143.

4 Satyanarayanan M, Mobile information access, *IEEE Personal Communications*, (February 1996).

5 Awerbuch B & Peleg D, Sparse partitions, *Thirty first IEEE Symp FOCS*, (October 1990).

6 Awerbuch B & Peleg D, Concurrent online tracking of mobile users, *Proc ACM SIGCOMM Symp Commun, Architecht Protocols*, (October 1991) 221-233.

7 Imielinski T & Badrinath B R, Querying in highly distributed mobile environments, *Proc Eighteenth VLDB*, (August 1992) 41-52.

8 Mitra S & DasBit S, A cost-effective location update strategy in cellular mobile environment, *Proc Int Conf CIC*, (October 2001).

9 Mitra S & DasBit S, Precise location identification in a cellular mobile network, *Int Conf Personal Wireless Commun*, (December 2002), 265-269.

10 DasBit S, Raha P & Mitra S, A probabilistic location management strategy in cellular mobile environment, *Proc IEEE TENCON*, **2** (October 2002) 1008-1011.

11 DasBit S, Raha P & Mitra S, A distributed probabilistic location management strategy in cellular mobile environment, *Workshop Commun Network Media Int Conf Eurasia-ICT*, (October 2002).

12 DasBit S & Mitra S, Challenges of computing in mobile cellular environment – A Survey, *J Comput Commun, Elsevier Science*, **26** (2003) 2090-2105.

13 Xu J & Lee D L, Querying Location-dependent data in wireless cellular environment, *W3C and WAP Workshop on Position Dependent Information Services*, (February 2000).

14 Zheng B & Lee D L, Processing Location Dependent Queries in a multi-cell wireless environment, *Proc ACM Press*, (2001) 54-65.

15 Imielinski T & Badrinath B R, *Wireless Computing, Commun ACM*, **10**(37) (1994) 19-28.

16 Kumar M & Venkatesh S, Lim K Y & Santosh H, Information access and QoS issues in a mobile computing environments, *Network Comput Applications*, **22** (1999).

17 Mitra S & DasBit S, Query processing in a cellular network – a database approach, *Proc IEEE VTC Spring Conf*, (May 2001) 248-248.

18 Mitra S & DasBit S, A location management strategy in cellular mobile environment using distributed searching, *Proc IEEE 3Gwireless*, (May 2001) 350-355.

19 DasBit S & Mitra S, A varying per user profile based location update strategy for cellular network, *Proc Int Conf ICCT/ WCC*, (August 2000) 754-760.

20 Bar-Noy A & Kessler I, Tracking mobile users in wireless communication networks, *IEEE Trans Inf Theor*, **39** (November 1993) 1877-1886.