Asian Journal of Information Technology 15 (8): 1349-1358, 2016
ISSN: 1682-3915

Object Movement Prediction Technique for Cluster-Based Location Monitoring Technique in Mobile Computing Environment

[1]G. Kalaimani and [2]B.G. Geetha
[1]Department of Computer Science, Anna University, Chennai, India
[2]Department of CSE, K.S. Rangasamy College of Technology, KSR Kalvi Nagar,
Thiruchengode, Tamil Nadu, India

Abstract: Most of the location updation technique often causes delay in order to update the information and hence, results in congestion in the network. As the objects are mobile in the mobile computing environment, answer to the spatial queries loses their significance if there is any delay in query processing or in communication. In order to overcome these issues we design an object movement prediction technique in order to predict object movement proactively and hence the delay for location updation can be reduced. For this, we implement a prediction algorithm along with the updating approach that precompute the movement of object in order to avoid any kind of delay for location updation. Once the query result is received from cell, it is evaluated in the query result computation phase in order to find whether the result is precomputed or it is to be calculated by scratch. A Location Update Table (LUT) is implemented that helps in determining all the possible movement of query and issue a location update only if, it may affects any query result and enables a smart server probing technique in order to avoid any kind of overhead in the network.

Key words: Movement prediction technique, cluster-based location, monitoring technique, mobile computing environment, (LUT)

INTRODUCTION

Mobile computing environment: In a mobile computing environment a mobile user with limited palm computer (or mobile computer) access various information through wireless communication (Peng and Chen, 2005).The main aim of mobile computing is to deal with the mobility of hardware, data and software in computer applications. It is a specialized class of distributed systems where some nodes gets separate from joint distributed operation and move freely in the physical space and reconnect to a possibly different segment of a computer network at a later stage to resume the suspended activities. The mobile computing environment must determine numerous data management issues like query processing, other than the packet transmission needed in conventional mobile communication protocols (Andamuthu and Balasubramanie, 2008). The main issues in mobile computing environment are introduced by the following characteristics:

Mobility: The location of mobile units is an important parameter when locating a mobile station that may hold the required data and when selecting information especially for location dependent information services.

Wireless medium: Wireless networks offer a smaller bandwidth than wired networks. There is an asymmetry in the communication because the bandwidth for the downlink communication (from servers to clients) is much greater than the bandwidth for the uplink communication (from clients to servers). Mobile environment is subjected to frequent disconnections.

Portability of mobile elements: The design of portable computers implies that they must be small, light, consume little energy, etc. This causes these computers to have generally more limited functionalities than fixed hosts, mainly in aspects such as computation power, storage capacity, screen size and graphic resolution, autonomy, etc.

Location monitoring technique: Due to advances in sensor devices and wireless communication technologies many new applications for military and civilian purposes has evolved in which aggregate location monitoring is one of the key applications. The aggregate location monitoring system provides many services that are as follows:

- Density queries: To determine the number of objects within a specified query region

Corresponding Author: G. Kalaimani, Department of Computer Science, Anna University, Chennai, India

- Safety Control: To send an alert if the number of persons in a certain area exceeds a predefined threshold
- Resource Management: In case if the number of people in a specified area is below specified threshold, it turns off some building facilities for efficient resource management

Query processing in mobile environment: In sensor networks, the main aim of continuous query is to collect periodical data from the objects under monitoring. For this type of queries special care is required to design it, in order to minimize the power consumption and maximize the lifetime of the sensor node.

The query processing in a mobile environment involves join processing among different sites that include static servers and mobile computers. The two most important metrics that need to be considered while evaluating query processing schemes are:

Efficiency: It is defined as the number of messages required to satisfy the query. It is important factor as sensor nodes have very limited power and message transmission dominates a node's power consumption.

Robustness: It is important metrics to evaluate query processing as queries may fail due to mobility and unreliable communication. (Peng and Chen, 2005; Sun, 2008; Huang *et al*., 2007).

Issues of query processing in mobile environment: To design a unique query processing technique the following challenges must be handled: Mobile query semantics: The result of the query depends on the location of its requester. Caching and sharing of query results must take into consideration the location of the query issuer. High workload: The database resides in a centralized server that typically operates a large mobile user community through wireless communication. Query promptness and accuracy: Due to user mobility, answers to Location Based Spatial Queries (LBSQs) will lose their relevancy if there is a long delay in query processing or in communication. (Ku *et al*., 2008). The query processing algorithm results in flooding and random walk:

Flooding: It floods the network with the query to find the answer. It returns the answer very quickly and is therefore highly-tolerant of changing network dynamics but it requires an excessive number of messages and can congest the network.

Random walk: It is very slow and may not complete at all if the network topology changes such that the source or destination cannot be visited. For this reason it has a much lower success rate than flooding. (Huang *et al*., 2007)

Impact of object mobility in query processing: The main impact of object mobility in query processing can be stated as follows:

- In the continuous query processing it is very difficult to handle the frequent location updates at the server and to handle the communication channel between the moving object and the server
- The limited bandwidth and dynamic topology in wireless network creates frequent mobile-server message exchanges that contain location information for the database engine to maintain up-to-date view of the world
- Usually whenever an object moves it sends its new location to the server that is wasteful as moving object can be located in an area where it does not affect the query result

Need for object movement prediction: Some of the point listed below can be use to define the necessity for the prediction of object movement:

- The prediction helps in traffic management applications, e.g., it predict congested areas before it takes place
- It avoids any kind of delay in location updation and prevents the communication channel from any kind of extra messages accumulated at the server

Literature review: Chow *et al.* (2009) in a study have proposed an aggregate location monitoring system in a wireless sensor network (Andamuthu *et al*., 2008). The underlying environment consists of counting sensors that are only capable of reporting aggregate locations, i.e., their sensing areas along with the number of detected objects residing there in, to a query processor. At the core of the query processor, they have proposed an adaptive spatio-temporal histogram that models the distribution of moving objects and answers aggregate monitoring queries based on aggregate locations. Furthermore, they have proposed three techniques, memorization, locality awareness and packing, that are combined together to enhance the histogram accuracy and efficiency by exploiting both the spatial and temporal features in aggregate locations. It takes care of the limitations of wireless sensor networks that include limited power and communication bandwidth. The drawback of this study is that in the proposed scheme they have not considered the delay metrics.

Ku *et al*. (2008) have presented a novel query processing technique that, though maintaining high scalability and accuracy, manages to reduce the latency considerably in answering LBSQs. Their approach is based on peer-to-peer sharing, which enables us to process queries without delay at a mobile host by using query results cached in its neighboring mobile peers. We demonstrate the feasibility of our approach through a probabilistic analysis, and we illustrate the appeal of our technique through extensive simulation results. The drawback of the proposed protocol they have not considered about the throughput and efficient utilization of bandwidth utilization in the network.

Hsueh *et al*. (2010) have design a framework that supports multiple query types (e.g., range and c-kNN queries). In this framework, their query re-evaluation algorithms take advantage of ASRs and issue location probes only to the affected data objects, without flooding the system with many unnecessary location update requests. The second proposed strategy for an arbitrary movement environment is the Partition-based Lazy Update (PLU, for short) algorithm that elevates this idea further by adopting Location Information Tables (LITs) which allow each moving object to estimate possible query movements and issue a location update only when it may affect any query results and enable smart server probing that results in fewer messages. They have first define the data structure of an LIT which is essentially packed with a set of surrounding query locations across the terrain and discuss the mobile-side and server-side processes in correspondence to the utilization of LITs. Simulation results confirm that both the ASR and PLU concepts improve scalability and efficiency over existing methods. The drawback of this study is that the proposed scheme they have not considered the delay metrics in the network.

Hendawi and Mokbel (2012) have presented the Panda (A Predictive Spatio-Temporal Query Processor) system for efficient support of a wide variety of predictive spatio-temporal queries that are widely used in several applications including traffic management, location-based advertising, and ride sharing. Panda targets long-term query prediction as it relies on adapting a well-designed long-term prediction function to: scale up to large number of moving objects and support large number of predictive queries. As a means of scalability, Panda smartly precomputes parts of the most frequent incoming predictive queries which significantly reduces the query response time. Panda employs a tunable threshold that achieves a tradeoff between query response time and the maintenance cost of precomptued answers. Experimental results, based on large data sets, show that Panda is scalable, efficient and as accurate as its underlying prediction function. The drawback of this study is that in the proposed system they have not considered about the energy efficiency and efficient utilization of bandwidth in the network.

Mouratidis *et al*. (2009) have proposed an on-air indexing method that uses a regular grid to store and transmit the data objects. They have designed algorithms for snapshot and continuous queries, over static or dynamic data. This is the first study on air indexing that addresses continuous queries and considers moving data objects. They have demonstrated the efficiency of their algorithm through an extensive experimental comparison with the current state-of-the-art frameworks for snapshot queries and with the naive constant re-computation technique for continuous queries. The future work is to study the different types of spatial queries such as reverse nearest neighbors and to extend their framework to process their snapshot and continuous versions.

Jeung *et al*. (2008) have presented a novel prediction approach, namely the Hybrid Prediction model that estimates an object's future locations based on its pattern information as well as existing motion functions using the object's recent movements. Specifically, an object's trajectory patterns which have ad-hoc forms for prediction are discovered and then indexed by a novel access method for efficient query processing. In addition, two query processing techniques that can provide accurate results for both near and distant time predictive queries are presented. Their extensive experiments demonstrated that proposed techniques are more accurate and efficient than existing forecasting schemes. The drawback of this study is that in the proposed scheme they have not considered about the throughput and delay metrics to determine the efficiency of proposed scheme.

MATERIALS AND METHODS

Proposed solution: In our previous study (Kalaimani and Geetha, 2013), we have proposed an adaptive cluster based location monitoring technique for query processing in mobile computing environment. In this technique, the object with minimum weight is chosen as Cluster Head (CH) which is estimated based on the parameters such as object distance, speed and mobility factor. When any object moves out of its current cluster, its location is

Node ID	Sequence No.	Object Distance	Speed	Mobility Factor

Fig. 1: Format of beacon message

updated using location update technique based on the location constraints. An adaptive location monitoring and updation technique is used to monitor and update objects locations dynamically.

Overview: As an extension research, we propose to design an object movement prediction technique, so that the movement can be predicted proactively and hence, the delay of location updation can be reduced. Here, we implement the prediction algorithm along with updating approach to design an efficient method to precompute the movement of object to avoid any kind of delay for location updation.

On getting the query result from cell, in result computing phase is first evaluated to find whether the result is precomputed or it is to be computed by scratch. In this phase Travel Time Grid (TTG) data structure is used to find the number of cells that contains object that is reachable within a certain time limit. The prediction function is used to determine the answer for the query result received from the object that lies within the cell (Hendawi and Mokbel, 2012). Also, we implement the Partition-based Lazy Update algorithm (Hsueh *et al*., 2010) that makes use of Location Information Tables (LITs) to help object in determining all the possible movement of query and it issue a location update only in case when it may affects any query results and also it enable a smart server probing technique to avoid overhead in the network.

Clustering algorithm: The applications involving location based services contain the mobile objects that are not evenly distributed. For example, the shopping malls, office buildings and bus terminals are densely populated than other areas such as park or leaf. Hence, we use clustering technique for modeling these mobile objects. The steps involved in the cluster formation are as follows.

- Step 1: Each object N_i broadcasts a beacon message to indicate its existence to the neighbors. The beacon message contains the object status. The format of beacon message is as shown in Fig. 1
- Step 2: Upon receiving the beacon messages, each neighbor object constructs its neighbor list.
- Step 3: The weight of each object (W) is computed based on the parameters such as object distance, speed and mobility factor (Estimated in previous study) shown in Eq. 4:

$$W=(\alpha \times D_t)+(\beta\times S_{av})+(\gamma\times\sigma)$$

Where, α, β and γ are coefficients used in weight calculation

- Step 4: The object with smallest W is chosen as cluster head. Then the complete objects that are neighboring to the cluster-head are not permitted to take part in the cluster head election process
- Step 5: The above steps are repeated for other objects pending to be chosen as a cluster-head or assigned to a cluster

Figure 2 demonstrates the formation of clusters. The $C_{1\text{-}5}$ represent the clusters and N_5, N_7, N_{13}, N_{18} and N_{22} represent the Cluster Heads (CH), respectively.

Estimation of metrics

Prediction function: This metric is used to predict the future location of large number of objects after a certain interval of time t (Hendawi and Mokbel, 2012):

$$\hat{R} = \frac{P(C_i / S_o)}{\sum_{d\in D_t} P(C_d / S_o)} \tag{1}$$

In the above equation, numerator represents the final destination of single object (R) and the denominator is the sum of the probabilities of all grid cells in D_t which is calculated with help of R where:

$$R = P\ (C_i\ S_o)$$

Here, R takes two inputs a cell $C_i \in C$ and a sequence of cells. Hence, (C_1, C_2, C_k) represents the current trip of an object O and after that R returns result with the probability that C_i will be the final destination of O. D_t set of all possible destination cells of object after time t which is calculated based on the travel time distance.

The recomputation of the prediction function is initiated only if object O changes its location from one cell to another rather than from point to another point within the same cell.

Object List (OL): It is a type of data structure which contains a list of all moving objects in the system. For each object O which belongs to OL, track of an object identifier and sequence of cells traversed by O in its

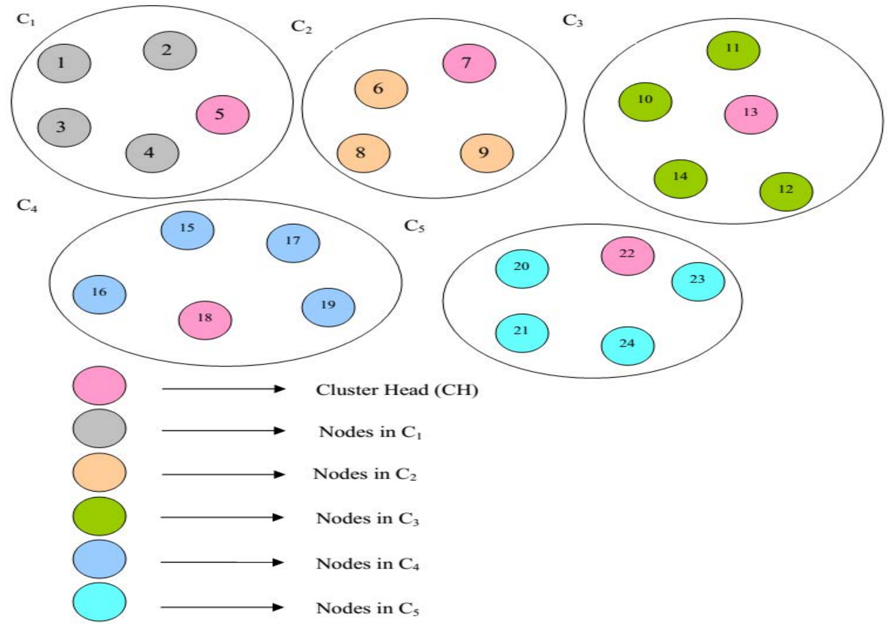

Fig. 2: Cluster formation

current trip. The example is shown in Fig. 2. The sequence of cells {C_{13}, C7, C2, C3} is represented for object O_2. Here, O_2 started at C_{13} and is currently moving inside C_3.

Space Grid (SG): In our proposed spatio based query processing, whole space is divided into N×N equal size grid cells. Here, for each cell $C_i \in SG$ four pieces of information is maintained:

- Cell Id which is cell identifier
- Current objects which represents the list of moving objects which is currently located inside C_i and presented as pointers to the object List (OL)
- Query List represents list of predictive queries which is recently issued on C_i

Here, each query Q in list is represented by triple (Time, Counter and Answer). Time represents future time included in Q, Counter represents number of times that Q is recently issued. Answer represents precomputed answer for query Q which have different format based on the type of Q. Frequent cells update the list of cells with the information that one of their pre calculated answer need to be updated with movement of an object in the cell C_i

Travel Time Grid (TTG): It is a two dimensional array of N2×N2 cells where each cell TTG (i, j) contains average travel time between space cells C_i and C_j, where C_i and C_j SG. This grid is preloaded in the generic framework of predictive query processing structure which is a read-only data structure.

Query result computation phase: This phase consists of two inputs:

- A predictive query Q which is received either as range, aggregate or k-nearest neighbor and asks about future location after particular time t and
- A cell C_i which overlaps with query area of interest S. The output consists of partial answer of Q which is computed from cell C_i

Purpose of this phase:

- This phase starts by checking whether the query answer at the input cell C_i is already computed or not
- If it is already computed, this phase is instantly concluded by updating the query result Q with precomputed answer of C_i
- If the answer is not precomputed then it proceeds by calculating the answer of C_i from scratch

Here, we apply a smart time filter in order to limit the search only to those objects which has the probability to reach cell C_i within the future time t. It utilizes Travel Time Grid (TTG) data structure in order to find a set of cells C_R that consist of objects which is reachable to C_i within the time limit t. After that prediction function is calculated for

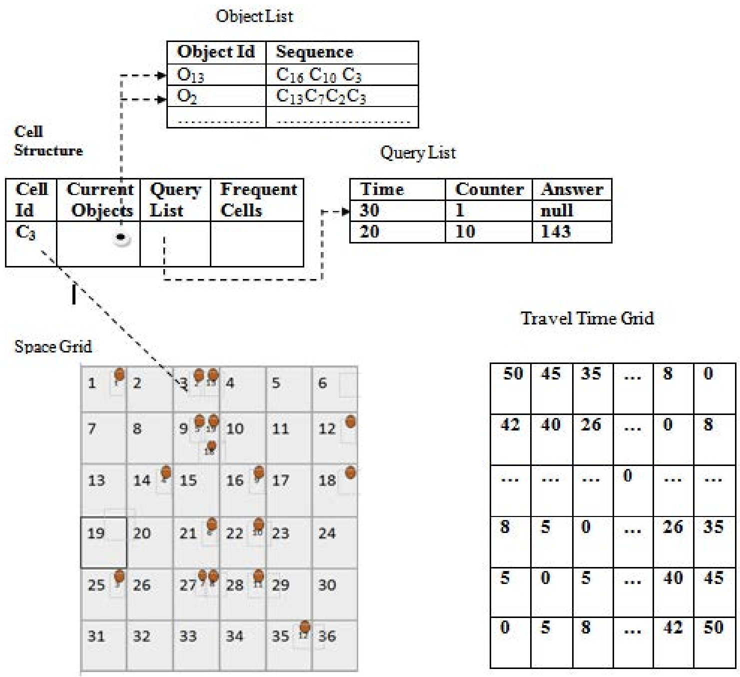

Fig. 3: Architecture of spatio DATA structure

the object that lie within any of the cells in C_R and all the object is collected to build the answer result from the cell C_i.

Algorithm:
Input: Q, S, t
Query result← null, cell result ← null1.
if answer exists in cell C_i at time t then
cell result ← read answer from cell C_i
sgsdg
else
C_R find set of cells with objects reachable to C_i within time limit t
for each cell $Cj \in C_R$ do
for object $O \in$ current object in C_j do
Object prediction ← using prediction function
Update Result (Cell Result, Object Prediction)
end for
end for

RESULTS

Location Update Table (LUT): In this study, we will describe the updation method after the result computation phase. Here, we will extend the location updation technique of our previous work by implementing Location Update Table which is generated in case of following two events (Fig. 3):

- If any of the existing query in the cell C_i changes its location or
- If a new query Q is found in the network

Here, we use a LUT value for LUTserv (i, j) which stores an integer number and is use to represent safe distance. The safe distance for LUTserver is the minimal linear distance in cells from the LUTserv (i, j) to the nearest query boundary. While assigning value to the LUTserv (i, j) two cases need to be considered:

- If LUTserv (i, j) doesn't overlap a query boundary then LUT, value$\geq$0
- if LUTserv (i, j) is covered by a query boundary then LUT.value=-1

In Fig. 4a object framework is shown with a set of registered queries and moving objects in the network at time t_o and the corresponding server-side LUT is created at t_0 in Fig. 5.

In this example it is assumed that the size of server side LUT and the size of object grid is same. There are two types of cells zones based on the LUT value:

- Border Zone (LUT value =-1): It consists of cells that overlap with boundaries of some queries Q
- Zero Zone (LUT value =0): This is the prediction zone that might be covered by neighboring moving queries as time proceeds. It has a safe distance which is equal to Zero and hence it is more likely to be covered by moving query Q_1 soon

Both of the above mentioned zones are important indicators for moving object to decide location update. In order to predict the location of moving query, each moving objects updates its local LUT and hence the new prediction cells is considered as zero zones.

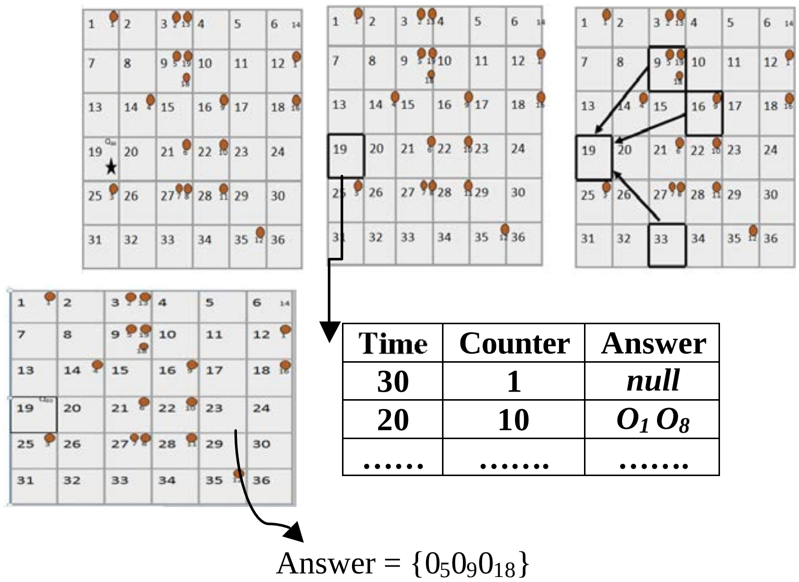

Fig. 4: a) Object framework with registered candidate at time t0; b) Server side LUT at time t0

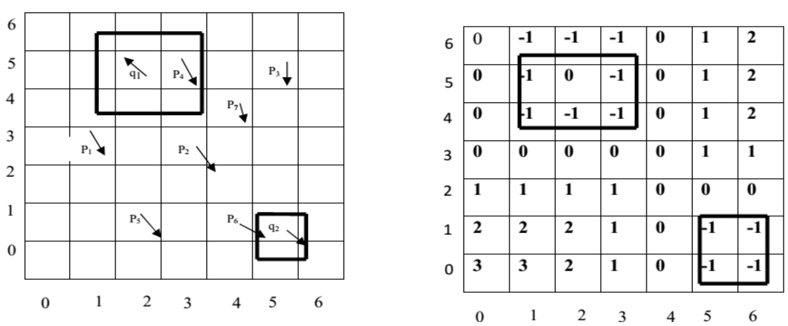

Fig. 5: Speed vs delay

Advantages:

- The proposed technique precomputes the movement of object to avoid any kind of delay in the location updation
- It also avoids overhead in the network

Simulation model and parameters: The Network Simulator (NS2) (Network Simulator), is used to simulate the proposed architecture. In the simulation, 100 mobile nodes move in a 500×500 m region for 50 sec of simulation time. All nodes have the transmission ranges from 250-450 m. The simulated traffic is Constant Bit Rate (CBR). The simulation settings and parameters are summarized in Table 1.

Performance metrics: The proposed Object Movement Prediction technique for Cluster-based Location Monitoring Technique (OMPCLM) is compared with the KNN technique. The performance is evaluated mainly, according to the following metrics:

Table 1: The simulation setting

Parameters	Values
No. of nodes	100
Area Size	500 X 500
Mac	IEEE 802.11
Transmission Range	250,300,350,400 and 450m
Simulation Time	50 sec
Traffic Source	CBR
Packet Size	512
Protocol	DSDV
Speed	5,10,15,20 and 25m/s
No. of object nodes	5

Packet delivery ratio; it is the ratio between the number of packets received and the number of packets sent. Packet drop; it refers the average number of packets dropped during the transmission. Delay; it is the amount of time taken by the nodes to transmit the data packets.

DISCUSSION

Based on speed: In our first experiment, we vary the mobile speed as 5, 10, 15, 20 and 25 ms-1.

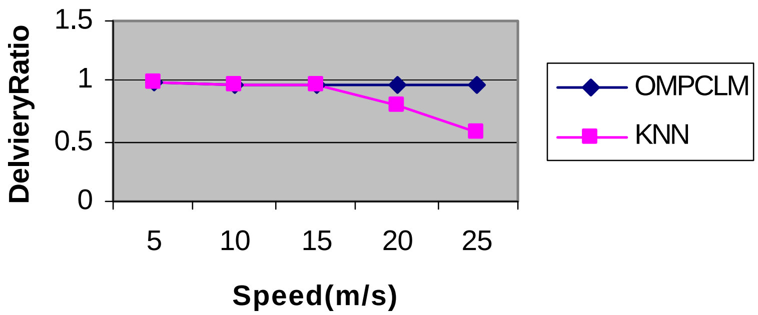

Fig. 6: Speed vs delivery ratio

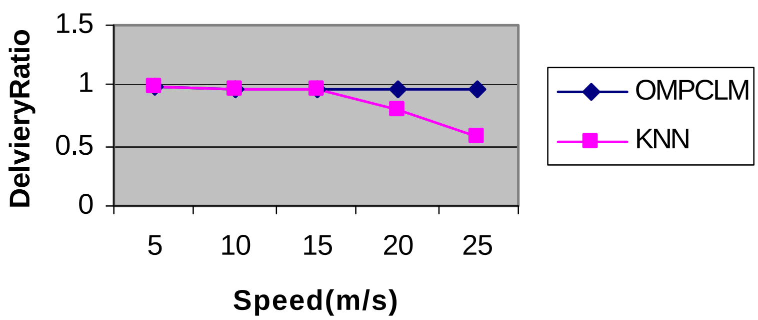

Fig. 7: Speed vs drop

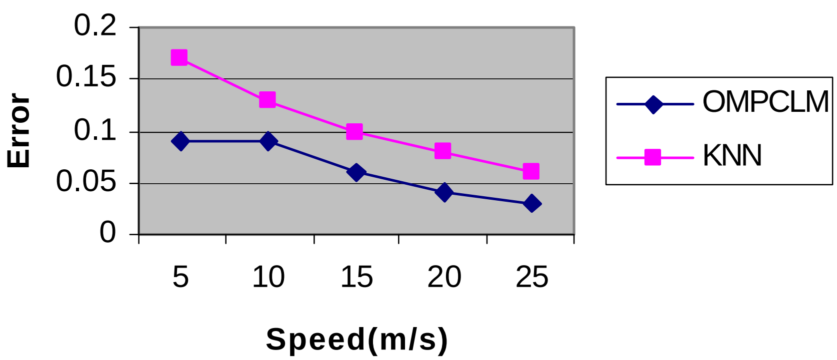

Fig 8: Speed vs error estimation

Figure 6 shows the delay of OMPCLM and KNN techniques for different mobile speed scenario. We can conclude that the delay of our proposed OMPCLM approach has 34% of less than KNN approach.

Figure 7 shows the delivery ratio of OMPCLM and KNN techniques for different mobile speed scenario. We can conclude that the delivery ratio of our proposed OMPCLM approach has 12% of higher than KNN approach.

Figure 8 shows the drop of OMPCLM and KNN techniques for different mobile speed scenario. We can conclude that the drop of our proposed OMPCLM approach has 70% of less than KNN approach.

Figure 9 shows the error estimation of OMPCLM and KNN techniques for different mobile speed scenario. We can conclude that the estimated error of our proposed OMPCLM approach has 44% of less than KNN approach.

Based on range: In our second experiment we vary the transmission range as 250, 300, 350, 400 and 450m

Figure 10 shows the delay of OMPCLM and KNN techniques for different Range scenario. We can conclude that the delay of our proposed OMPCLM approach has 20% of less than KNN approach.

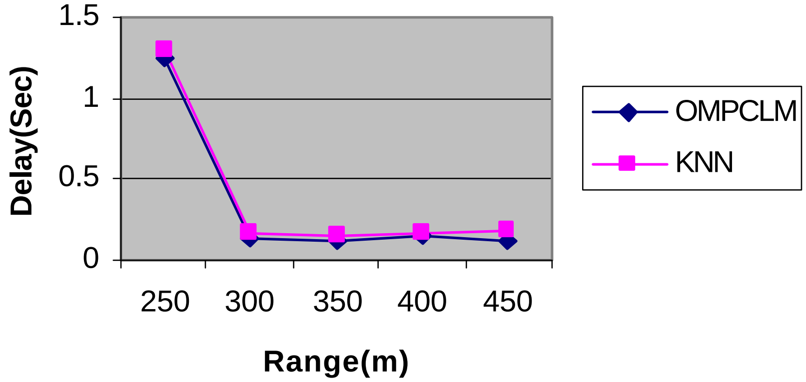

Fig 10: Range vs delivery ratio

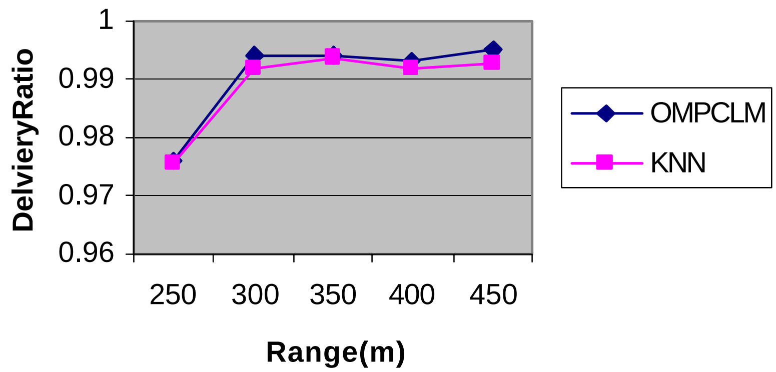

Fig 11: Range vs drop

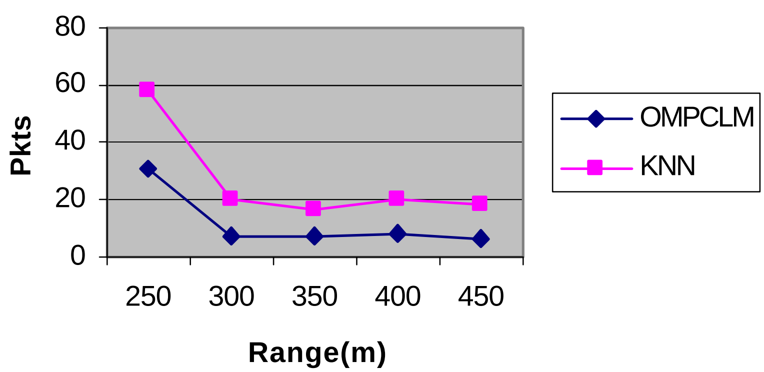

Fig 12: Range vs error estimation

Figure 10 shows the delivery ratio of OMPCLM and KNN techniques for different range scenario. We can conclude that the delivery ratio of our proposed OMPCLM approach has 0.2% of higher than KNN approach.

Figure 11 shows the drop of OMPCLM and KNN techniques for different range scenario. We can conclude that the drop of our proposed OMPCLM approach has 59% of less than KNN approach.

Figure 12and shows the error estimation of OMPCLM and KNN techniques for different range scenario. We can conclude that the estimated error of our proposed OMPCLM approach has 22% of less than KNN approach.

CONCLUSION

In this study, we have proposed an object movement prediction technique in order to predict the object movement proactively which avoids any kind of delay for location updation. Here, we have implemented prediction algorithm along with the updating approach that precompute the movement of object in order to avoid any kind of delay for location updation. In the query result computation phase, query result received from the cell is first evaluated to find weather it is precomputed or it is to be computed by scratch. Travel Time Grid (TTG) data structure is used to find the number of cells object that is reachable within a certain time limit. Prediction function is used to determine the answer for the query result received from the object that lies within the cell. Location Update Technique (LUT) is implemented in order to determine all possible movement of query that allows location update only if it affects any result and enable a smart server probing technique in order to avoid any kind of overhead in the network.

REFERENCES

Andamuthu, T.P. and P. Balasubramanie, 2008. An efficient architecture for query processing in mobile environment. Int. J. Comput. Sci. Network Sec., 8: 6-12.

Chow, C.Y., M.F. Mokbel and T. He, 2009. May Aggregate location monitoring for wireless sensor networks: A histogram-based approach. Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware, May 18-20, 2009, IEEE, Taipei, Taiwan, ISBN: 978-1-4244-4153-2, pp: 82-91.

Hendawi, A.M. and M.F. Mokbel, 2012. Panda: A predictive spatio-temporal query processor. Proceedings of the 20th International Conference on Advances in Geographic Information Systems, November 07-09, 2012, ACM, Redondo Beach, CA, USA., ISBN: 978-1-4503-1691-0, pp: 13-22.

Hsueh, Y.L., R. Zimmermann and W.S. Ku, 2010. Efficient location updates for continuous queries over moving objects. J. Comput. Sci. Technol., 25: 415-430.

Huang, H., J.H. Hartman and T.N. Hurst, 2007. Efficient and robust query processing for mobile wireless sensor networks. Intl. J. Sensor. Networks, 2: 99-107.

Jeung, H., Q. Liu, H.T. Shen and X. Zhou, 2008. A hybrid prediction model for moving objects. Proceedings of the 24th IEEE International Conference on Data Engineering ICDE, April 7-12, 2008, IEEE, Cancun, Mexico, ISBN: 978-1-4244-1836-7, pp: 70-79.

Kalaimani, G. and B.G. Geetha, 2013. Adaptive cluster-based location monitoring technique for query processing in mobile computing environment. Intl. Rev. Comput. Software, 8: 1760-1768.

Ku, W.S., R. Zimmermann and H. Wang, 2008. Location-based spatial query processing with data sharing in wireless broadcast environments. Mobile Comput. IEEE. Trans., 7: 778-791.

Mouratidis, K., S. Bakiras and D. Papadias, 2009. Continuous monitoring of spatial queries in wireless broadcast environments. Mobile Comput. IEEE. Trans., 8: 1297-1311.

Peng, W.C. and M.S. Chen, 2005. Query processing in a mobile computing environment: Exploiting the features of asymmetry. Knowl. Data Eng. IEEE. Trans., 17: 982-996.

Sun, J.Z., 2008. QoS aware query processing algorithm for wireless sensor networks. J. Comput., 3: 32-41.