

Target Tracking using Personalized Data Management

Naina Gupta
M.TECH (NIT, KKR)
2797, N.H.B.C
PANIPAT

Tanu Jindal
M.TECH (NIT, KKR)
1107, N.H.B.C
PANIPAT

ABSTRACT

In the present paper, personalized data management which is a scheme used to answer queries based on user preferences is suggested to be used for target tracking application in the wireless sensor network. This scheme enables faster recovery by reducing search results according to the type of target and environment. At the same time, personalized data management helps in deciding various parameters based on the different areas where this application is used.

General Terms

Wireless sensor network, personalized data management, target tracking

Keywords

Preference Database Management, PrefDB, PrefDB application, WSN, Target Tracking

1. INTRODUCTION

The problem being tackled here relates to the problem of target tracking in wireless sensor networks. Target tracking is one of the well studied applications in wireless sensor networks [1][2][3]. Since sensor networks are typically used to monitor the environment, one fundamental issue is the location-tracking problem. In location tracking problem, we need to trace the roaming paths of moving objects/individuals in the area in which sensors are deployed. It poses challenge in mainly two ways: (1) there is lack of central control mechanisms and backbone network in such an environment and (2) there is limited wireless communication. Since the data that needs to be communicated should be small, the sensors can make use of personalized data management to derive the data to be passed to the required sensor node. For this approach to be used in target tracking, profile corresponding to each target is created. These target profiles have been widely used for retrieving information and filtering text-based data items. In these profiles, user preferences are represented in terms of a single or multiple keyword vectors. Target preferences are stored in the form of atomic query constructs in profiles and are thus used for personalising answers. User preferences provided either as query condition or stored in profiles are of various types: unconditional, conditional, multi-value etc. They are expressed as (a) *hard* constraints, that are either satisfied or not satisfied at all as in case of where clause in traditional SQL[4]; or (b) *soft* constraints, that should be fulfilled as closely as possible as in case of preferring in [5]. Personalization is based on maintaining, for every target type, a target profile whose structure is intimately related to the features of the data and query models. In particular, here it is assumed that target preferences are stored in the form of atomic elements of queries, i.e., atomic selection or join conditions and are thus known as *atomic target preferences*.

These atomic preferences in the application of target tracking are based on the target types and their speed. These profiles

allow only the most relevant data to be communicated to the application.

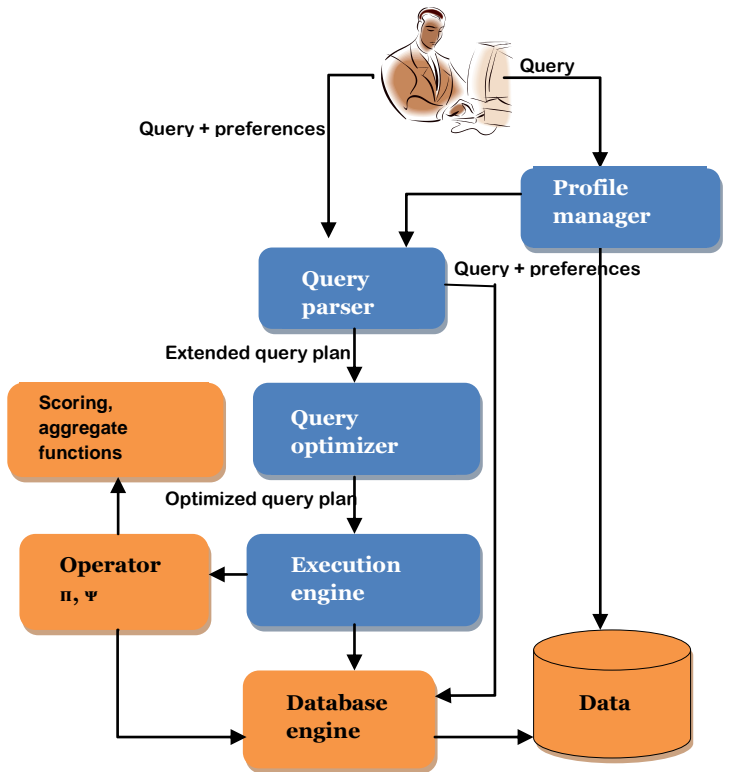


Figure 1

2. PROPOSED ARCHITECTURE

2.1 PrefDB Architecture

The key idea of *PrefDB* is the use of an extended relational data model and modified algebra that allow different types of preferential queries to be expressed easily. A *preferential query* combines preference relations, extended relational and prefers operators. Using their combination it returns a set of tuples satisfying Boolean query conditions. Each output tuple is assigned a score based on the estimated relevance [6] using logical formulas [7]. These scores have been calculated after evaluating all prefer operators on the corresponding relations [8]. The better a tuple matches preferences and the more (or more confident) preferences it satisfies, the higher final score and confidence values it will have respectively. *PrefDB* offers two alternative query options. One is that preferences can be provided along with the input query. Other one is that the system can enrich a non preferential query with related preferences. The architecture (as shown in figure 1) and the associated description is taken from [8]. Here input is the user query and the related preferences and is fed to the query parser which generates a preferential query expressed in the extended relational algebra. Along with this, it adds

projections for all attributes that will be used as part of a prefer operator and for all join attributes as well. Finally, the query parser generates an *extended query plan* keeping the order of the operators same as given in the query.

The main goal of query optimizer is to improve the input query plan by applying suitable query transformation rules that make use of the algebraic properties of the preferred and the extended relational operators. Query optimizer applies the following set of heuristic transformation rules as in [8]:

1. Selections are pushed down the query plan as far as possible. If the selection condition is such that it involves several relations, the condition is split and each piece is pushed down separately.
2. Projections are pushed down the query plan as far as possible.

The query optimizer fed the optimized query plan to the *execution engine* which in turn realizes the execution of the plan. The execution engine is mainly responsible for the execution of an extended query plan and also supports different query evaluation strategies.

Figure 2 represents the proposed architecture. The sensor nodes are usually scattered in a sensor field as shown in Fig.2. Each of these scattered sensor nodes has the capabilities to collect data and route data back to the sink and the end users. Data are routed back to the end user by a multihop infrastructure less architecture through the sink as shown in Fig. The sink may communicate with the task manager node via Internet or Satellite. Here task manager node can make use of the PrefDB to analyse the collected information based on the confidence scores computed by the PrefDB. This will help the task manager node to effectively compute the data deciding the action to be taken.

2.2 Network Architecture using PrefDB

In this scheme, the sensor network architecture is the static cluster-based architecture [9]. The entire field is divided into a number of clusters, where each cluster has a cluster head that performs computationally intensive tasks in the cluster. Thus a good cluster head selection algorithm with low convergence time and high energy efficiency can be used [10]. In clusters, we have two types of nodes. (1) Normal nodes that is capable of sensing seismic, acoustic, magnetic and other activity. (2) Periphery nodes which are present at the periphery of the network. Some of the non-periphery nodes could possess GPS (or other similar system) to act as beacon nodes. The sensor nodes have two modes (1) Active mode in which all parts of the sensor node are fully powered. (2) Sleep mode in which the sensor listens to an interrupt in order to transit to the active state. The periphery nodes in this architecture are always in active state and keep looking for the target. However, the number of periphery nodes may be increased to increase the lifetime of the network with suitable sleep-awake cycle [11].

When a periphery node or a normal node detects a target, it notices its cluster head of target presence. The periphery nodes are assumed to have adequate sensor types to detect all intended targets. To fasten this process PrefDB can be used, in which based on the observations made using sensors like target characteristics, periphery nodes can query the PrefDB to retrieve the information about the target like mobility pattern of the target, type of the target, its expected speed and direction information. This information can help the cluster head in determining the downstream cluster where the

target can move in future. The cluster heads on receiving interrupt from the periphery node or a normal node transits to active state. The normal nodes change to active state and send the sensed parameter values and the information along with the timestamp to the cluster head. The target location is determined by the cluster heads at successive time instants. During this step, the cluster head selects three normal nodes based on the highest value of sensed parameter. This has been chosen as a trade-off between increasing accuracy and minimizing the consumed energy [11]. This also decides the density of the network [12]. Target localization may be done using trilateration or (weighted) centroid localization.

The target's velocity and direction and the other patterns obtained using PrefDB from the sensor nodes can be used by the cluster heads to determine more accurately the most likely downstream cluster heads that the target is approaching and then inform them in advance about the incoming target. Additional details of the architecture are available in [11].

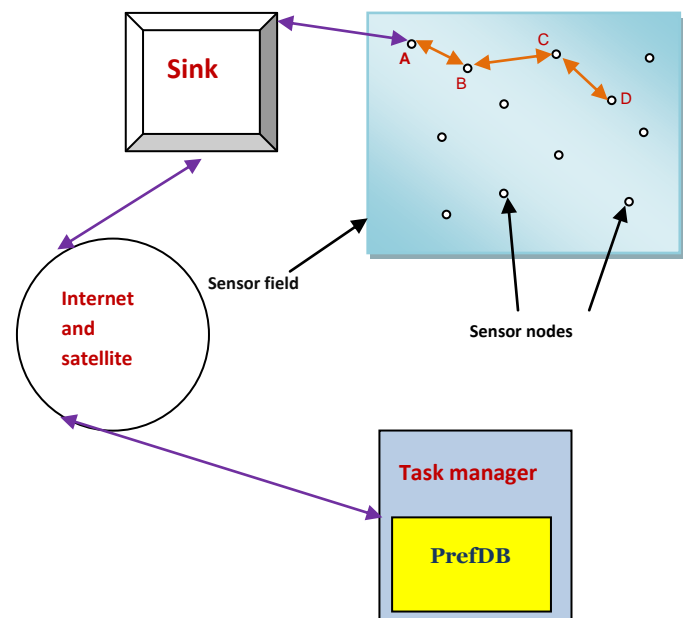


Figure 2

3. PROBLEM DESCRIPTION

A potential problem that can occur when WSNs are used for target tracking is that the network may lose track of the target even after using PrefDB as some reasons are unavoidable. This reason includes the following as given in [11]:

1. Node Failures: As WSNs have limited infrastructure and nodes are battery operated that have limited battery capacity, node failure may occur due to hardware failure, battery discharge, enemy action, etc.

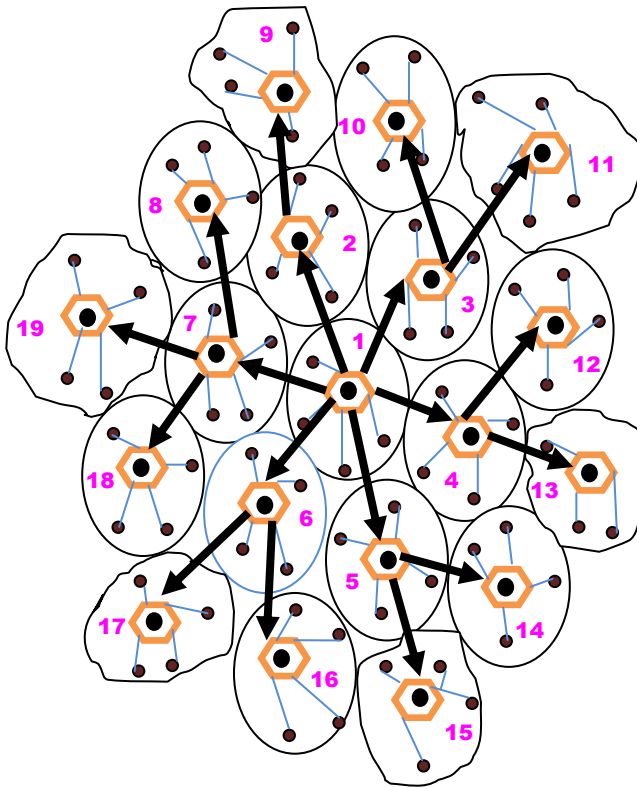


FIGURE 3

2. Network Failure: The network may fail due to communication breakdown, congestion, environmental factors, enemy action etc.

3. Localization Errors: Localization technique always have some errors and thus estimated location of the target is never perfect. These errors may have a cumulative effect on target tracking.

4. Inaccurate Prediction: As mentioned earlier, not all clusters are awake due to energy considerations. The cluster heads are informed in advance about the target arrival so that they can wake up other nodes in the cluster. For this, an upstream cluster has to predict the future location of the target which may have errors. Errors in this prediction can lead to target loss, due to inaccurate prediction about the downstream cluster.

5. Sudden change in target's velocity/direction: The target may change the trajectory or speed abruptly. If this happens, then network may not be able to keep track of the target efficiently.

Due to these factors, it may happen that target may get lost in WSN. However, mission critical (as in military) situations demand credible and reliable target tracking. In such situations target loss cannot be tolerated. So, to overcome this, a recovery mechanism is needed that quickly recover the target trajectory and at the same time minimize the time and energy spent for recovery. Some recovery mechanisms are already built but these recovery mechanisms can make use of PrefDB to improve their performance as explained next.

4. MODIFIED ALGORITHM FOR RECOVERY

The recovery mechanism can be divided into distinct phases as described in [11]. However this can be made better using PrefDB as explained below:

Loss of Target: The recovery mechanism is initiated when a cluster reports loss of target. The most important part in initiating target recovery mechanism is to find out that the target has been lost, i.e. the target's current position is not known. In the prediction strategy used, a cluster head currently tracking the target is responsible for waking up the downstream cluster based on the predicted information from the peripheral nodes about the mobility pattern and the type of the target. If the downstream cluster head does not sense the target in some stipulated time, target loss is confirmed. (Care has to be taken to reduce false positives. For example, recovery mechanism may be started without the target being lost if the target becomes stationary in the present cluster and downstream cluster has already been notified of target's arrival.) However if we use PrefDB in the previously defined approach, it can be predicted whether the target can become stationary based on the target type thus reducing false positives.

Assume the target was being tracked by cluster head 1 as shown in Figure 3. Based on the knowledge from the PrefDB, this node predicts that the target is likely to move to cluster 2 or cluster 4 so it will have less chances of target loss. But in case it happens, following approach is used. A cluster may predict two or more clusters as next possible location of the target due to target changing course within the cluster or prediction errors. Assume that the target is not detected by either cluster 2. Cluster head 2 will trigger a timeout and declare target loss to cluster head 1. Target loss can also be declared when a cluster head after being warned by a periphery node does not detect target in some stipulated amount of time.

Search: The search phase has been added to reduce cases of false recovery initiation. In this phase, a cluster head queries the downstream clusters for target presence. Continuing the above example, the cluster head 1 first checks of target presence within its own cluster for target presence. On failing to find the target, it queries cluster head 4 for target presence. If cluster head 4 confirms target presence, then cluster head 1 notifies cluster head 2 and no recovery is initiated. If cluster head 4 does not confirm target presence in stipulated time, cluster head 1 confirms target loss and initiates active recovery. However this approach can be made efficient using PrefDB. For example, if some target has specific mobility pattern in the field, then target loss can be declared using PrefDB in those expected clusters also, thus increasing chances of recovery.

Active recovery: The upstream cluster is the focal point of recovery. The cluster head sends a target loss message to all one hop clusters. On receiving message, the neighboring clusters wake up the normal/periphery nodes in the cluster. In

the above example, cluster head 1 sends target loss message to one hop clusters 2 to 9. These eight clusters are thus woken up and activated. If the target is not found in stipulated time, cluster heads 2 to 9 timeout and wake up their one hop clusters in a similar manner. Thus, clusters 10 to 21 are sent active recovery messages. This ensures that the area of active recovery is increased gradually.

However by the application of the PrefDB, we may have the additional information about the target like its type , its mobility pattern etc .This information can be used to modify the procedure. For ex, if we have a target type say A and based on the target type , we can have its maximum and the minimum attainable speed so that we can modify our recovery procedure based on its type. Like for a target that can attain very high speed, for that target type, recovery procedure can be made fast if the target is critical and its loss cannot be afforded .This can be done by searching for the target two hops simultaneously.

Similarly mobility patterns from its previous history collected by the PrefDB can also be used. Like if know that the target is such that it abruptly changes its speed and direction, or follows a pattern during its journey, then this information can be used to modify the recovery procedure by activating only the sensors in the most likely clusters.

On sensing the target, a normal node switches its radio on, sends the sensed value and keeps its radio to listen mode.

Here in the recovery procedure, the recovery algorithm can be modified using the information received by the cluster head from the associated peripheral nodes which in turn are collected by them by querying the preferential database.

5. CONCLUSION AND FUTURE WORK

This paper studied the problem of loss of target trajectory in a target tracking wireless sensor network. For WSNs to be used effectively and reliably for target tracking, efficient recovery mechanisms have to be formulated. A basic prediction based recovery mechanism has been modified in this paper by taking advantage of the PrefDB using which performance of the previous algorithm can be improved in terms of message size , network load and efficiency. PrefDB is a system that answers the queries taking into account the preferences. Here it is proposed that we can maintain history corresponding to the targets in the WSN by using the sensors .This history information is then queried using PrefDB by the periphery nodes whenever a new target arrives. This PrefDB provides a limited result set which is based on the query that can be easily analyzed. In the PrefDB, while querying, preferences are specified based on the observations about the target made by the sensors. This can lead to better prediction of the downstream cluster in recovery procedure and low signaling overhead in some cases.

6. REFERENCES

- [1] A. Arora et al, "A Line in the Sand: A wireless sensor network for target detection, classification and tracking," *Computer Networks*, vol. 46, no. 5, pp. 605–634, 2004.
- [2] "Exscal: Elements of an extreme scale wireless sensor network", in *Proc. International Conference on Embedded and Real-Time Computing Systems and Applications*, vol. 11, pp.1533–2306, 2005.
- [3] Z. Wang, W. Lou, Z. Wang, J. Ma, and H. Chen, "A novel mobility management scheme for target tracking in clusterbased sensor networks," in *IEEE DCOSS*, Jun. 2010, pp. 172–186.
- [4] G. Koutrika and Y. E. Ioannidis., "Personalization of queries in database systems", In *Proceedings of 20th International Conference on Data Engineering (ICDE)*, pp no 597–608, 2004.
- [5] Werner Kießling, Markus Endres, Florian Wenzel., "The Preference SQL System – An Overview" In *IEEE Database engineering bulletin* ,volume 34, number 2, pp. 11-18,June 2011.
- [6] R. Agrawal, E. Wimmers,"A Framework for Expressing and Combining Preferences", In *Proc. of ACM SIGMOD*, pp no. 297-306,2000.
- [7] Chomicki J., "Querying with intrinsic preferences", In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, Vol51 No.1, pp no. 34–51,Aug 2002.
- [8] G. Koutrika and Y. E. Ioannidis PrefDB ,"Bringing preferences closer to the DBMS", In *Proc. of ACM SIGMOD*, pp no. 665-668,2012.
- [9] I.F. Akyildiz., W. Su., Y. Sankarasubramaniam., and E. Cayirci., "Wireless Sensor Networks: A Survey", *Elsevier Computer Networks*, volume 38, Issue 4, pp. 393-422, March 2002.
- [10] K.Ramesh and Dr. K.Somasundaram,"a comparative study of cluster head selection algorithms in wireless sensor networks", *International Journal of Computer Science & Engineering Survey (IJCSSES)*, Vol.2, No.4, November 2011.
- [11] Anshu Khare and Krishna M. Sivalingam, "On Recovery of Lost Targets in a Cluster-based Wireless Sensor Network", *PerCom Workshops IEEE (2011)*, pp no. 208-213, May 2011.
- [12] Yang H, Sikdar B , "A protocol for tracking mobile targets using sensor network", *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, Anchorage, Alaska, pp. 71–81, May 2003.
- [13]http://en.wikipedia.org/wiki/Wireless_Sensor_Networks