



ENERGY EFFICIENT TARGET TRACKING WITH PARTICLE FILTERING TECHNIQUE IN WIRELESS SENSOR NETWORKS

Aysegul Alaybeyoglu
Department of Computer Engineering
Celal Bayar University, 45140
Manisa, Turkey
aysegul.alaybeyoglu@bayar.edu.tr

Abstract- In this study, a new target tracking algorithm is proposed for wireless sensor networks. The aim of the algorithm is to decrease energy consumption of the system by decreasing the ratio of target misses. Next location of the target is predicted by using Particle Filtering (PF) technique which aims to represent the posterior density function by a set of random samples with associated weights. Nodes are deployed according to the hexagon shaped network topology in which each of the hexagons represents a cluster with a predetermined leader node. In order to decrease the ratio of target misses, nodes that are closer to the target's predicted location are woken up to make them ready to detect the target. This increases the probability of detecting the target by one of the neighboring hexagons when the target makes sudden turns or unexpected movements. Tracking performance of the proposed algorithm is evaluated by comparing with K-Nearest Cluster Tracking (KNCT), Wakening Based Target Tracking Algorithm (WBTA)[10] and Generic Static Tracking Approach (GSTA) in terms of miss ratio and energy consumption metrics.

Key Words- Wireless Sensor Networks, Target Tracking, Particle Filtering

1. INTRODUCTION

Target tracking is one of the most popular sensor network application which has tremendous potential for commercial, military and environmental application areas [1]. In a classic target tracking application, when a target enters the tracking area, it is detected by a group of nodes that are located closer to the target. The nodes that detect the target communicate with each other and determine the location of the target by using one of the localization techniques. After the location of the target is determined, this information is sent to the base station [2]. If the target's location can not be determined, target miss occurs and recovery mechanisms are used to recapture the target.

Clustering strategy fits the nature of target tracking applications where neighboring nodes form a cluster in order to cooperate with each other. In a cluster, nodes are classified as leader and member nodes. While leader node is responsible for calculating the location of the target, member nodes are responsible for sending their sensed data to the leader node. When a target enters a cluster, the leader node collects all information from members and calculates the location of the target by using one of the localization techniques. Reducing the energy consumption is one of the most important benefits of the cluster based approaches. There are many cluster based target tracking algorithms in literature[3]. An important step of the tracking algorithm is to predict target's next

location. Many prediction techniques are proposed in literature [4]. PF is one of the most popular prediction technique that aims to represent the posterior density function by a set of random samples with associated weights. Main task of PF is to assign and update appropriate weights to the particles. [5,6] are some examples of the tracking algorithms that use PF technique.

In this study we propose a target tracking algorithm that aims to decrease the energy consumption by decreasing the ratio of target misses. In order to decrease ratio of target misses, we used Local Distributed Particle Filtering as a prediction technique in which samples are drawn and importance weights are calculated and normalized locally. In this technique each node sends only sufficient data (such as mean and covariance) for the estimation to the leader node, that is responsible for providing the filter estimate [11]. Then we awake the nodes that are located on the neighboring hexagons to the predicted location of the target. We also decrease the energy consumption with the cluster based approach by decreasing the number of communications. The rest of the paper is organized as follows: system is described in Section 2, the proposed algorithm is explained in Section 3, simulation results are given in Section 4 and lastly conclusions are given in Section 5.

2. SYSTEM DESCRIPTION

2.1. Network Model

We modeled the wireless sensor network as an undirected graph $G(V,E)$ where set of vertices (V) indicates the sensor nodes and set of edges (E) indicates the communication links between these nodes. We used a hexagon shaped sensor topology as shown in Fig. 1. In this topology, each hexagon represents a cluster and each node with black color located at the center of the hexagons represents the leader node. Clusters are formed statically at the time of network deployment that all the member nodes and their related leader nodes are defined before the tracking algorithm executes. According to the network topology, a member node can be a member of three clusters with different leader nodes. Member nodes have a table including their leader nodes id that provides them to know their leaders to send messages. All nodes have the same communication range r and each node is surrounded by six neighboring nodes. The overlap area of two neighboring nodes are reduced to the common edge and by this way network topology could be modelled as hexagon [7].

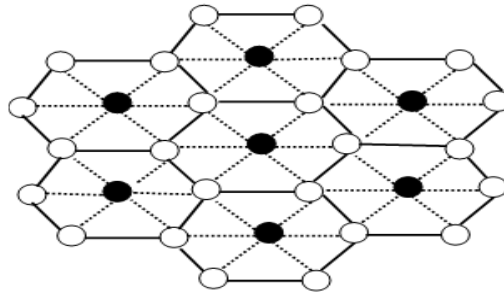


Fig.1 Hexagon Network Topology

We assumed that all nodes are stationary, time synchronized and aware of their own and neighbor's locations. Each node has sensing capability as well as computing and communication capabilities. Nodes are able to determine their distances to the target by observed *RSS*. Each node is determined with a unique node id. Links between the nodes are symmetric. Thus if there is a link from u to v , there exists a reverse link from v to u .

2.2. System Dynamics

System and the observation models are required to analyze and inference about a dynamic system [4]. A maneuvering target tracking system can be represented by these models. Under the Bayesian framework a dynamic system can be described by the following equations:

$$X_k = f_k(X_{k-1}, w_k) \quad z_k = h_k(X_k, n_k) \quad (1)$$

where the state of the system is X_k , the observation model is z_k which is composed of sensor measurements, w_k is the system noise and n_k is the measurement noise. f_k and h_k represent the deterministic process and measurement models. While the system model describes the evolution of the state, observation model updates the state with the receipt of new measurements. Prediction and update are two main stages that the Bayesian filter consist. In prediction stage, the system model is used to predict the next state of the system and the update state uses the observations to modify the predicted state. We used acoustic sensor nodes for tracking the target. When the target enters tracking area, the acoustic signal from the moving target is detected by the sensor nodes. At time k , the sensed acoustic energy Z_k^i of node i is calculated as [8]:

$$Z_k^i = \frac{S_k}{(x_k - x_k^i)^2 + (y_k - y_k^i)^2} + n_k^i \quad (2)$$

Where S_k denotes the acoustic energy emitted by the target, (x_k, y_k) and (x_k^i, y_k^i) represent the location of the target and the node i at time k , respectively.

2.3. Particle Filtering Technique

PF is one of the most popular prediction technique that addresses the problem of estimating the state of a dynamic system from the observations. PF is a technique for implementing a recursive Bayesian filter by Monte Carlo simulations. It provides an effective solution for non-Gaussian and nonlinear systems. The aim of PF is to represent the posterior density function by a set of random samples $(x_k^i, i = 1 \dots N)$ with associated weights $(w_k^i, i = 1 \dots N)$ which represent the importance of the samples on prediction. The weights of the samples should be normalized to make sum of them equal to 1. If $\{x_{0:k}^i, w_k^i\}_{i=1}^N$ denotes a random measure that characterizes the posterior pdf $p(x_{0:k}, z_{1:k})$, the posterior density at k can be approximated as:

$$p(x_{0:k} z_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(x_{0:k} - x_{0:k}^i) \quad (3)$$

Assigning weights to each sample:

$$p(x_k | z_k) = \frac{p(z_k | x_k) p(x_k | z_{k-1})}{p(z_k | z_{k-1})} \quad (4)$$

Relationship with Bayes rule:

$$\propto \frac{p(z_k | x_k) p(x_k | x_{k-1}) p(x_{k-1} | z_{k-1})}{p(z_k | z_{k-1})} \quad (5)$$

$$\propto p(z_k | x_k) p(x_k | x_{k-1}) p(x_{k-1} | z_{k-1}) \quad (6)$$

The weight update equation can be shown as:

$$w_k^i \propto \frac{p(x_k^i | z_k)}{q(x_k^i | z_k)} \quad (7)$$

$$\propto \frac{p(z_k | x_k^i) p(x_k | x_{k-1}^i) p(x_{k-1}^i | z_{k-1})}{q(x_k^i | x_{k-1}^i, z_k) q(x_{k-1}^i | z_{k-1})} \quad (8)$$

$$\propto \frac{p(z_k | x_k^i) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{k-1}^i, z_k)} \quad (9)$$

$$\propto p(z_k | x_k^i) w_{k-1}^i \quad (10)$$

Normalization:

$$w_k^i = \frac{w_{k-1}^i}{\sum_{j=1}^N w_{k-1}^j} \quad (11)$$

One of the problem encountered in PF is the degeneration problem which means that after number of iterations, some samples will have negligible weights. Therefore, using these samples brings additional computational cost. Degeneration can be identified by introducing the effective sample size N_{eff} defined as follows [8]:

$$N_{eff} = \frac{1}{\sum_{i=1}^N (w_k^i)^2} \quad (12)$$

To avoid the degeneration problem, N_{eff} value is compared with the resampling threshold N_{thr} . If the effective number of samples is less than a given threshold, then resampling is performed. In resampling procedure, new samples are drawn from the

current sample set with probabilities proportional to their weights and then current sample set is replaced with the new particle set [4].

3. ENERGY EFFICIENT TARGET TRACKING WITH PF (EETT)

When the target enters the tracking area, it will be detected by the member and the leader nodes that are located close to the target. The leader in whose cluster the target locates, collects all the sensing reports from its members and predicts the next state of the target by PF. When the distance between the target and the leader node exceeds a predefined threshold, the leader node sends a *WARNING* message including last estimation results (estimation of the state and the covariance) to the member node closest to the target's predicted location. As the member receives *WARNING* message, it forwards this message to the both leaders of the neighboring hexagons. As soon as the leader nodes receive this message, they broadcast *WAKE* messages to their member nodes to make them ready to detect the target. The new awake leader nodes make new estimations based on the estimation results from the previous leader node as well as new measurements from the sensor nodes in its own cluster [9]. By this way, when the target makes sharp turns or unexpected movements, it will be detected by the nodes in one of the neighboring hexagons. The pseudo code of the algorithms for sensors and the leader nodes are given in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1 Tracking Algorithm at sensor nodes

```

1:   Node detects the target as soon as it enters the radio range of a node
2:   while target is inside the radio range do
3:       sense the target and send sensed data to the leader node
4:   end while
5:   if Warning_Msg is received then
6:       send Warning_Msg to the leader nodes whose ids are in the list
7:   end if
8:   if Wake_Msg is received then
9:       become awake and be ready to detect the target
10:  end if

```

Algorithm 2 Tracking Algorithm at leader nodes

```

1:   while target is inside the cluster do
2:       collect the sensed data from member nodes
3:       predict target's next state
4:       if (the distance between the target and me > threshold) then
5:           send Warning_Msg to the member node that locates
               closest to the predicted location
6:   end while
7:   if Warning_Msg is received then
8:       broadcast Wake_Msg to the member nodes
9:   end if

```

An example operation of the proposed tracking algorithm is given in Fig. 2. As shown in the Fig.2.a, the target exists in the cluster of leader node 1. Node 1 collects the sensed data from its member nodes {2, 3, 4, 5, 6, 7}. With PF technique, node 1 calculates the posterior density of the target state from these measurements. When the distance between the node 1 and the target exceeds the predetermined threshold, node 1 sends *WARNING* message to its member node 2 that is the closest node to the target's next predicted location. Up on receiving this message, node 2 forwards it to the neighboring leader nodes 13 and 8. When these nodes receive *WARNING* message, they broadcast a *WAKE* message to their member nodes {2,7,12,14,15,16} {2,3,9,10,11,12} respectively and make them ready to detect the target. The target can go towards and backwards, can turn right and left. By waking up the nodes on neighboring hexagons, we increase the probability of detecting the target as shown in Fig.2.b

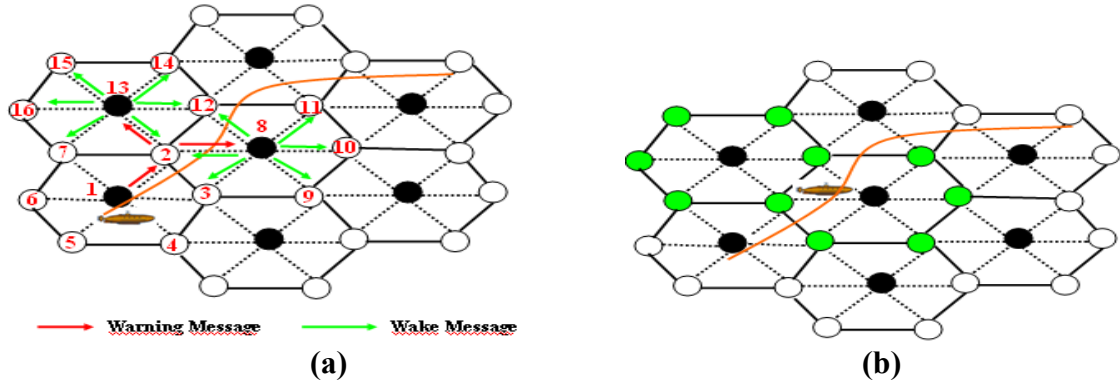


Fig.2 An Example Operation

Although we decrease the probability of target misses by waking up the nodes that are located neighbor to the predicted location, we can not guarantee the prediction to achieve 100% accuracy. In these cases, target missing occurs which causes a requirement for recovery mechanism in order to recapture the target. As the ratio of target missings increases, the frequency of using recovery mechanism also increases which causes excessive energy consumptions. The recovery mechanism that we used is shown in Fig. 3.

When the pre-woken up clusters with the leader nodes 8 and 13 detect and calculate the location of the target, they send a *REPLY* message to the previous active leader node 1 once. If leader node 1 does not receive any *REPLY* message during a certain time, it decides that a target missing occurs and it wakes up all nodes that are located on neighboring hexagons. Fig.3 shows an example of how the recovery mechanism we used executes. As it can be seen from the figure, leader node 1 notices that target missing occurs and it broadcasts *RECOVERY* message to its member nodes {2,3,4,5,6,7}. When these nodes receive this message, they forward it to their leader nodes whose ids exist in their leader lists. Leader nodes that receive *RECOVERY* messages, broadcast *WAKE* messages to their member nodes. For example in Fig.3, when node 3 receives *RECOVERY* message from node 1, it forwards this message to its leader nodes 8 and 20. As the nodes 8 and 20 receive *RECOVERY* message, they broadcast *WAKE* messages to their member nodes. The same operations are executed in

member nodes 2,4,5,6 and 7. By this way, all nodes that locate on neighboring hexagons are waken up in order to recapture the target.

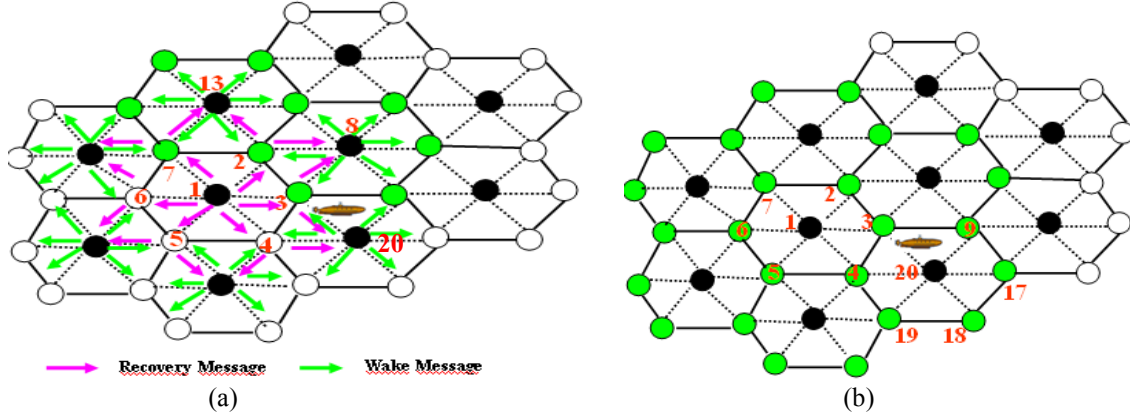


Fig.3 Recovery Mechanism

It can be seen from Fig.3.b that target is recaptured in the cluster of leader node 20 and node 20 informs previous active leader node 1 by sending *REPLY* message.

As the target miss ratio increases, in order to recapture the missing target, the frequency of recovery mechanism usage also increases. This increment leads high energy consumptions. With this study, we aimed to decrease energy consumptions by decreasing the frequency of recovery mechanism usage. In order to decrease the frequency of recovery mechanism usage, by awakeing the nodes that locate neighbor to the target's predicted location, we decreased the ratio of target misses.

4. SIMULATIONS

We implemented the proposed tracking algorithm (EETT) in *ns2* simulator version 2.31. We programmed the tracking algorithm with *C* and modelled the network environment with *TCL*. We used 100 nodes and run time of a simulation is set to 200 sn. IEEE 802.11 radio and MAC standards are chosen as lower layer protocols. We used 200 particles and *S* is set to 5000. The system noise is a zero mean Gaussian white noise with $N(0, 1)$, and the measurement noise is zero mean Gaussian white noise with $N(0, 0.01)$. We compared EETT with K-Nearest Cluster Tracking (KNCT), Wakening Based Target Tracking Algorithm (WBTA)[10] and Generic Static Tracking Approach (GSTA) againts target miss ratio and energy consumption. Both in KNCT, WBTA and GSTA, clusters are formed randomly at the time of network deployment. In GSTA active leader node wakes up only one cluster that locates closest to the target's predicted location. Hence, when the target moves faster that the expected, the probability of target misses increases. In order to overcome this, WBTA which awakes number of clusters along the predicted trajectory of the target. Although WBTA performs better than GSTA and is good at in tracking very fast moving targets, this approach considers only the target's linear movements. Hence, in WBTA, when the target makes sudden or unexpected turns, the probability of target miss increases. In KNCT, two clusters that are closest to the target's predicted location are waken up. The general idea of GSTA, WBTA and KNCT are given in Figure 4.

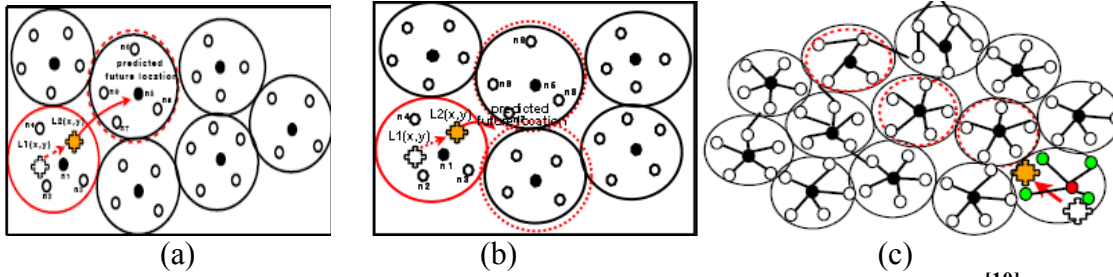


Fig.4 General Idea Proposed in (a) GSTA (b) KNCT (c) WBTA^[10]

With the proposed algorithm EETT, by awakening the nodes locating on the hexagons that are neighbour to the target's predicted location, we aimed to decrease target miss ratio when it makes sudden turns or unexpected movements. We tested and compared the performances of GSTA, WBTA, KNCT and EETT for both linear and nonlinear movements. For linear movements we used Random Waypoint mobility model. The speed of the target is varied to measure the detection capability of our algorithm under different mobility conditions. The speeds are chosen from 5 m/s to 30 m/s. In Fig.5, we plotted the target's linear movements to illustrate the tracking accuracy of EETT for different speeds. As it can be seen from Fig.5, calculated coordinates are very approximate to the real coordinates in each range of speed.

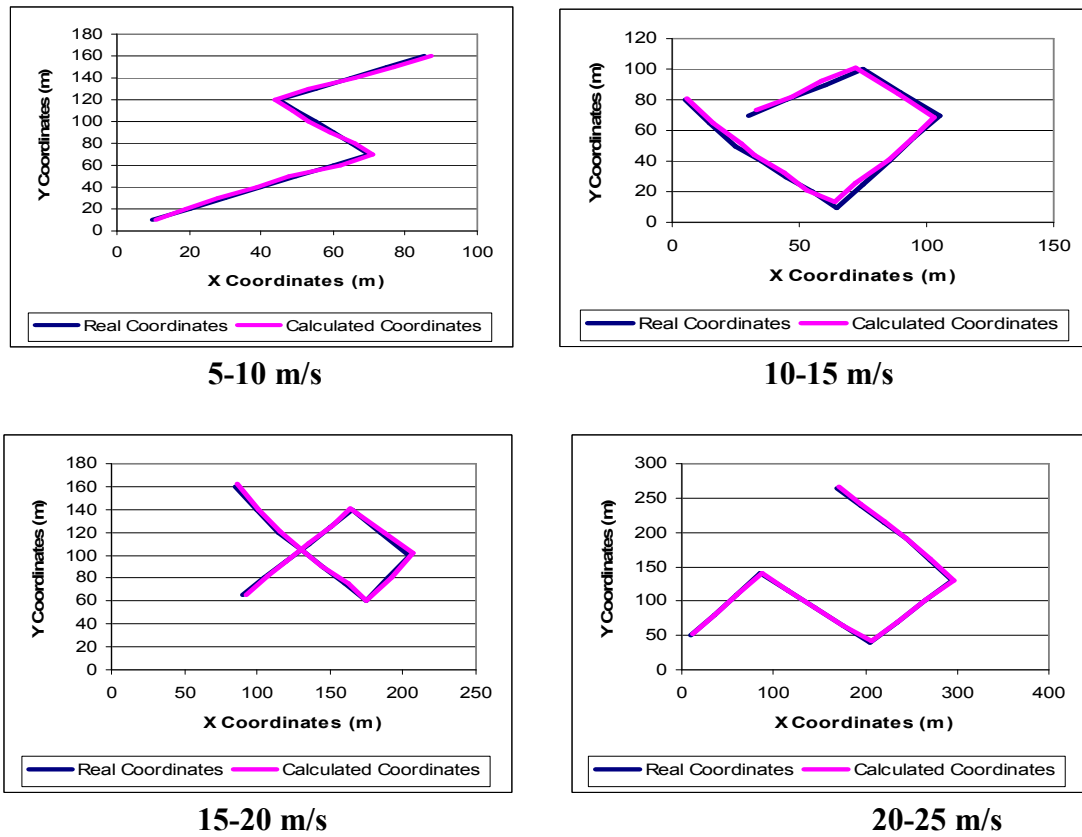


Fig.5 Real and Calculated Linear Trajectories of the Target for Different Speeds

For nonlinear motions, we implemented a software that generates random non linear movement scenarios. Simulation results are obtained from many nonlinear trajectories and one of them is plotted in Fig. 6 to illustrate the tracking accuracy of the EETT.

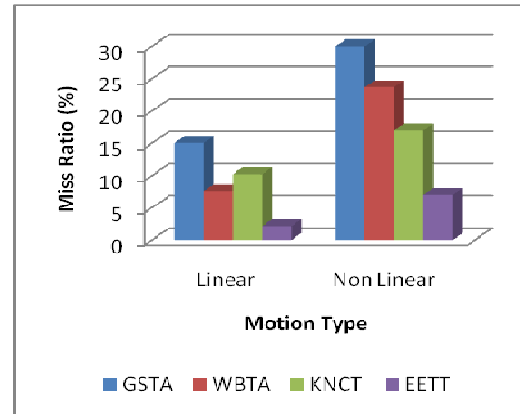
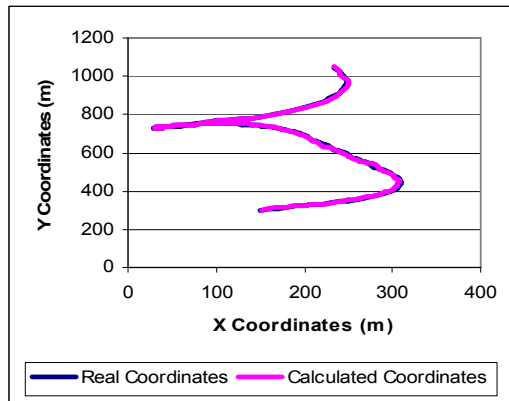


Fig.6 Non Linear Trajectory of the Target **Fig.7 Miss Ratio for Different Motion Types**

As it can be seen from Fig.6, EETT performs well in tracking the maneuvering target. Fig.7 compares the miss ratio of GSTA, WBTA, KNCT and EETT against linear and nonlinear motion types. For both of the motion types, miss ratio of EETT is less than the other algorithms. WBTA performs better than KNCT for linear motions with the help of number of clusters that are pre-waken linearly. For nonlinear motions, when the target makes sudden turns, the probability of detecting the target is higher in KNCT than WBTA. Miss ratio in non linear movements is higher than linear movements because of the sudden maneuvers of the target.

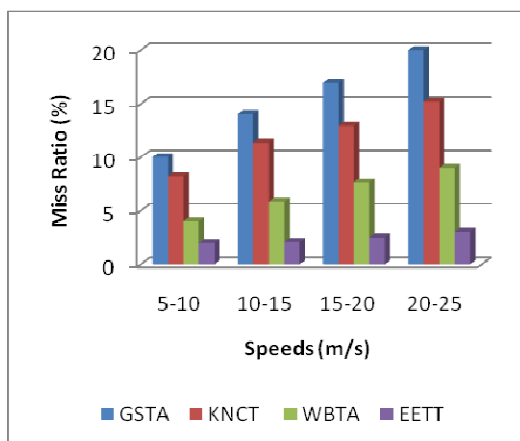


Fig.8 Miss Ratio for Different Speeds

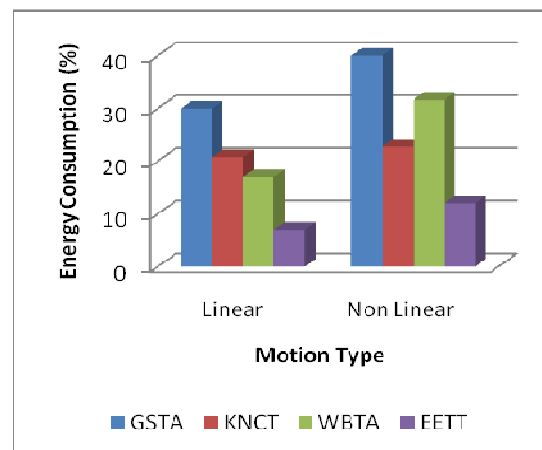


Fig.9 Energy Consumption for Different Motion Types

Fig.8 compares the miss ratio of the algorithms against different speeds. As it can be seen from the figure, miss ratio of EETT is less than the other algorithms for each range of node speeds. Lastly, Fig.9 shows the energy consumption comparison of the algorithms against different motion types. For each of the motion type, energy consumption in EETT is less than the other algorithms because of the less miss ratios of

EETT. As the miss ratio decreases, the frequency of recovery mechanism usage also decreases. This leads decrement in energy consumption and hence increment in network lifetime.

5. CONCLUSIONS

In this study, we propose a new algorithm for tracking linear and non linear moving target. The aim of our algorithm is to decrease miss ratios and energy consumptions by awakening the neighboring nodes to the target's predicted location. We compared our algorithm with GSTA, WBTA and KNCT for miss ratio and energy consumption metrics. We also plotted the real and calculated trajectories of the target. Simulation results show that EETT performs well in tracking linear and nonlinear moving targets. When we compared EETT with GSTA, WBTA and KNCT, for different motion types and the speeds, EETT performs better than the other algorithms in target miss ratio. When the energy consumptions of the algorithms are compared, the percentage of energy consumption in EETT is less than the other algorithms because of the less miss ratios. Our work is ongoing and we plan to extend our algorithm for tracking multi targets.

6. REFERENCES

1. B. Thorstensen, T. Syversen, T. Walseth and T. Bjornvold, Electronic Shepherd: A Low-Cost, Low-Bandwidth, Wireless Network System, Proc. Second Intl. Conference on Mobile Conference on Mobile Systems, Applications and Services, 245-255, 2004.
2. A. Alaybeyoglu, K. Erciyes, A. Kantarci, O. Dagdeviren, Tracking Fast Moving Targets in Wireless Sensor Networks, IETE Technical Review, 2010.
3. S. Bhatti, J. Xu, Survey of Target Tracking Protocols Using Wireless Sensor Network, In Proc. of ICWMC, pp.110-115, 2009
4. M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking, IEEE Trans. on Signal Processing, 174188, 2002
5. N. Shrivastava, R. M. U. Madhow, and S. Suri, Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms, In Proc. of SenSys, 251264, 2006.
6. J. Singh, U. Madhow, R. Kumar, S. Suri, and R. Cagley, Tracking multiple targets using binary proximity sensors, In Proc. of IPSN, 529-538, 2007.
7. Y. Xu, W. Lee, On Localized Prediction for Power Efficient Object Tracking in Sensor Networks, Proc. of ICDCS, 434, 2003.
8. X. Sheng and H. Y. Hu, Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor Networks, IEEE Trans. Signal Process. 53, 44-53.
9. Y. Huang, W. Liang, H. Yu, Y. Xiao, Target tracking based on a distributed particle filter in underwater sensor Networks, Wireless Communications and Mobile Computing, 1023-1033, 2008.
10. A. Alaybeyoglu, O. Dagdeviren, A. Kantarci, K. Erciyes, A Distributed Wakening Based Target Tracking Protocol for Wireless Sensor Networks, ISPDC 2010
11. A. Bashi, V. Jilkoy, X. R. Li, H. Chen, Distributed Implementations of Particle Filters, In Proc. Sixth International Conference of Information Fusion, 2003.