

Performance Evaluation of Cluster-based Target Tracking Protocols for Wireless Sensor Networks

Aysegul Alaybeyoglu*, Orhan Dagdeviren[†], Kayhan Erciyes[‡], Aylin Kantarci*

*Computer Engineering Department, Ege University, Izmir, Turkey

e-mail: (aysegul.alaybeyoglu, aylin.kantarci)@ege.edu.tr

[†]Computer Engineering Department, Izmir Institute of Technology, Izmir, Turkey

e-mail: orhandagdeviren@iyte.edu.tr

[‡]International Computer Institute, Ege University, Izmir, Turkey

e-mail: kayhan.erciyes@ege.edu.tr

Abstract—Target tracking is an important application type for wireless sensor networks (WSN). Recently, various approaches [1–11] are proposed to maintain the accurate tracking of the targets as well as low energy consumption. Clustering is a fundamental technique to manage the scarce network resources [12–19]. The message complexity of an application can be significantly decreased when it is redesigned on top of a clustered network. Clustering has provided an efficient infrastructure in many existing studies [1–8]. The clusters can be constructed before the target enters the region which is called the static method [1–4] or clusters are created by using received signal strength (RSS) from target which is called the dynamic method [5–8]. In this paper we provide simulations of static and dynamic clustering algorithms against various mobility models and target speeds. The mobility models that we applied are Random Waypoint Model, Random Direct Model and Gauss Markov Model. We provide metrics to measure the tracking performance of both approaches. We show that the dynamic clustering is favorable in terms of tracking accuracy whereas the energy consumption of static clustering is significantly smaller. We also show that the target moving with Gauss Markov Model can be tracked more accurately than the other models.

I. INTRODUCTION

Recent technological advancements made sensor nodes cheap and readily available for academic and industrial usage. WSNs may consist of thousands of nodes deployed in a large area. Sensor nodes are suitable for various application types due to their sensing and wireless communication capability. Military surveillance, habitat monitoring and target tracking are some of the important types of applications for sensor networks.

In target tracking applications when a mobile target is sensed by some of the nodes, its position is calculated by cooperation of these nodes using localization techniques and aggregated data is sent to the sink node. Target tracking applications can be cluster based [1]–[8] spanning tree based [9], and prediction based [10], [11]. In spanning tree based algorithms, nodes which detect the target select a root and construct a spanning tree. The tree is configured while the target moves away. Prediction based tracking algorithms aim to estimate the next position of the target based on the current moving speed and direction of the target.

Clustering is a widely used technique to ease the routing

operation and to manage the scarce resources in WSNs [12]–[19]. In clustered networks, nodes are either classified as cluster members or cluster heads. Cluster heads are responsible for managing the intra-cluster and cooperating in inter-cluster operations. In cluster based target tracking algorithms, member nodes detect the target and send the information to their cluster head. Cluster heads collect all information from members and calculate the position of the target by using localization techniques. After position of the target is calculated, cluster head sends the position information to the sink. Reducing the energy consumption is one of the most important benefits of the cluster based approaches. Cluster based target tracking algorithms can be further divided into two groups: static [1]–[4] and dynamic [5]–[8] approaches. In static approaches, the cluster and backbone infrastructures are built before the target tracking application starts. On the other hand, the clusters are dynamically constructed while nodes are sensing the target in dynamic approaches.

In this study, we investigate and evaluate the tracking performance of dynamic and static cluster based target tracking approaches against various mobility models. To compare the tracking accuracy of two approaches, we measure the miss and error ratio. We also evaluate the energy consumptions of dynamic and static approaches and give a general performance overview. The rest of this paper is organized as follows: In Section 2, the static and dynamic target tracking algorithms implemented are explained. The selected mobility models are described in Section 3. The performance evaluations obtained from simulation results are presented in Section 4. Finally, conclusions and future works are given in Section 5.

II. ALGORITHMS

A. Static Cluster Based Target Tracking Algorithm

In this approach, clusters are formed statically at the time of network deployment so all the member nodes and their related leader nodes are defined before the tracking algorithm comes into play. This cluster ready infrastructure brings simplicity into target tracking and decreases the energy consumption. Although these are desirable features of this approach, the restrictions on memberships can cause some problems for fault tolerance.

Fig.1 illustrates the general idea in the static cluster based target tracking algorithm. As the target enters the network area, it will be detected by a cluster of nodes in which the target currently presents. The leader node, collects all the sensing reports from its members and by using one of the localization techniques it calculates the location, speed and the trajectory of the target. After that, it predicts the future location of the target and informs the cluster head, closest to the future location of the target, about the oncoming target. When the cluster head receives this information, it wakes up its members and makes them ready to detect the moving target. Member nodes send received signal strength values to the new active leader node as long as they sense the target. By using the sensing reports received from the member nodes, the new active leader node calculates the location, speed and the trajectory of the target. This process continues as long as the target moves and the nodes sense.

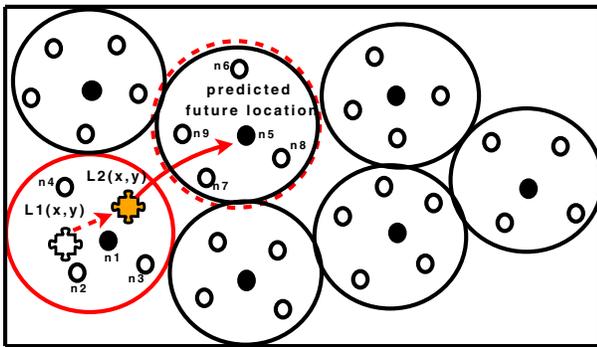


Fig. 1. Static Cluster Based Target Tracking Algorithm.

As shown in Fig.1, the target is firstly detected by the leader node n_1 and by its members n_2 , n_3 and n_4 . n_1 collects the sensing reports from its members and calculates the locations $L1(x,y)$ and $L2(x,y)$ intermittently. By using location information, it predicts the future position after a given period of time. Having this information, n_1 sends a warning message to the leader node n_5 that is closest to the predicted future location of the target. This warning message means that the target is approaching. After receiving this message n_5 , wakes up its member nodes n_6 , n_7 , n_8 , n_9 and makes them ready to detect the target.

Fig.2 shows the flow diagram of the static cluster based target tracking algorithm.

B. Dynamic Cluster Based Target Tracking Algorithm

In this approach, clusters are formed dynamically as the events occur in the network area. This approach does not impose any restriction on memberships. For example, a node can be a member of different clusters at different times which makes this approach more advantageous for minimization of localization errors. Although these are desirable features of this approach, the requirement for leader election mechanism increases the energy consumption for tracking the target.

Fig.3 illustrates the general idea of the dynamic cluster based target tracking algorithm. As the target enters the

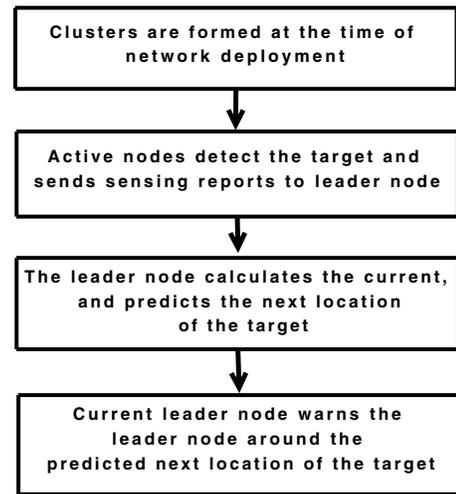


Fig. 2. Flow Diagram of the Static Cluster Based Target Tracking Algorithm.

network area, it will be detected by some nodes that are closer to the target. By using one of the leader election algorithm [20], a node that is closest to the target is selected as the leader node and the cluster is dynamically formed with the leader node's one hop neighbors. After forming the initial cluster, the leader node calculates the location, speed and the trajectory of the target by processing the sensing reports sent by its members. By using this information, it predicts the future location of the target and sends a warning message to the node closest to the target's predicted future location. The node receiving this message, becomes the new leader node and forms its cluster with one hop neighbors.

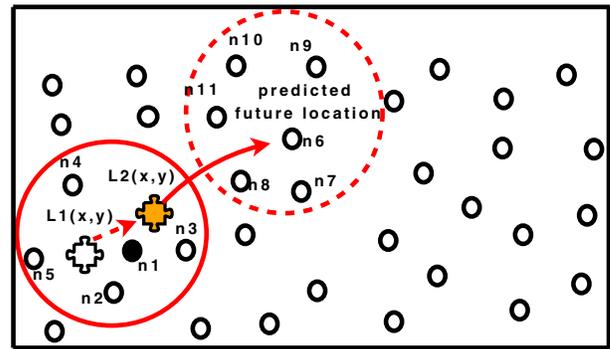


Fig. 3. Dynamic Cluster Based Target Tracking Algorithm.

As shown in Fig.3, the target is firstly detected by the nodes n_1 , n_2 , n_3 , n_4 and n_5 . The closest node to the target, n_1 , is selected as the leader node and it forms the initial cluster with its neighbors n_2 , n_3 , n_4 and n_5 which send sensing reports as long as they sense the target. By processing these reports, the leader node n_1 , predicts the future location and sends a warning message to the node n_6 that is closest to the target's predicted future location. After receiving this message, the new leader node n_6 , forms its cluster with its one hop neighbors n_7 , n_8 , n_9 , n_{10} , n_{11} .

Fig.4 shows the flow diagram of the dynamic cluster based target tracking algorithm.

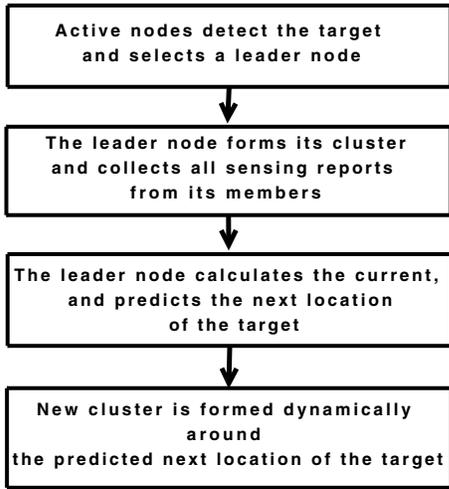


Fig. 4. Flow Diagram of the Dynamic Cluster Based Target Tracking Algorithm.

III. MOBILITY MODELS

Various mobility models were proposed to mimic the nature of mobile nodes in real applications. A hierarchical classification of the mobility models is depicted in Fig. 5 [21]. In this section, we briefly review the random waypoint, random direction and Gauss-Markov mobility models. These models are chosen because they are commonly used by researchers and their simulations can be handled with ease using *ANSim* simulator [22]. Random waypoint and random direction models are belong to the group of random-based mobility models. In random based mobility models, the nodes choose their speed, direction and destination without any restrictions. On the other hand, in mobility models with temporal dependency, the velocity of a node at different time slots are correlated. Gauss-Markov mobility model is an example of the mobility models with temporal dependency [21]. For each described mobility model we show the plot of an example trajectory generated with the *ANSim* simulator.

A. Random Waypoint Model

Random waypoint model (RWM) [23], [24] is one of the widely used mobility model in WSN simulations. CMU Monarch group implemented RWM for ns2 as the setdest tool. The idea of the RWM is simple. Each node travels from a starting coordinate to a random ending coordinate with a randomly generated constant velocity. The velocity is randomly picked from $[0, V_{max}]$ interval. For greater V_{max} values, the nodes move faster since $V_{avg} = (0 + V_{max})/2$, becomes greater. When a node reaches the destination point, it waits for a T_{pause} time before arriving in the next destination. As the T_{pause} becomes greater, the total time of the nodes in the stationary state increases [21]. In this model, nodes move along a zigzag line. An example of RWM trajectory is shown in Fig. 6.

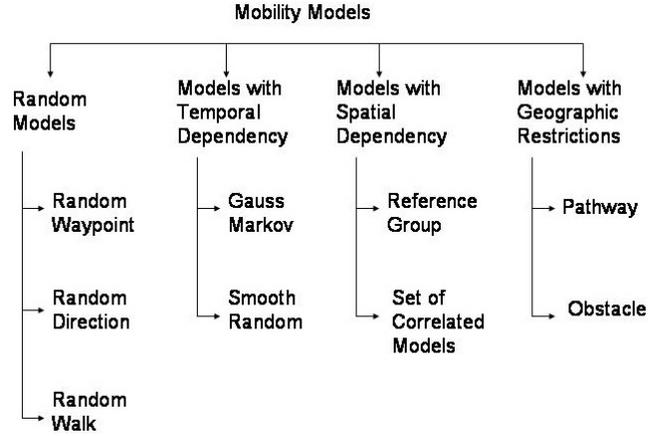


Fig. 5. Mobility Models.

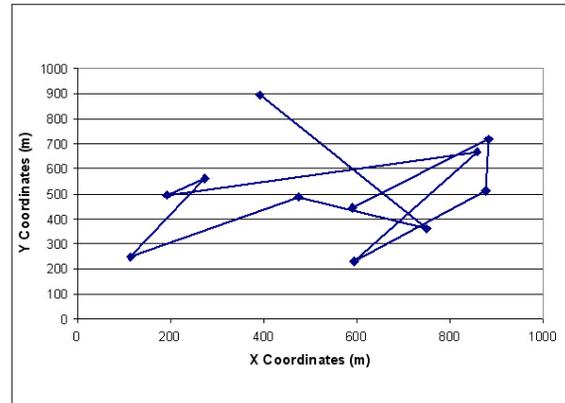


Fig. 6. An Example Trajectory of RWM.

B. Random Direction Model

One of the most important problems of the RWM is the non-uniform distribution of the nodes. The nodes cluster at the center of the simulation area as the simulation time elapses. As the nodes move to the center, the node density at the border of the simulation area becomes closer to zero causing to a non-uniform node distribution. To overcome this problem, random direction model (RDM) is proposed [25]. In RDM, instead of choosing a random destination, nodes choose a random direction which reaches the boundary of the simulation area. When a node reaches the boundary of the simulation area, it waits for T_{pause} time and it chooses a new direction to travel [21]. Fig. 7 shows an example of RDM trajectory.

C. Gauss-Markov Model

Random mobility models are simple in nature. They lack capturing realistic behavior of the mobile nodes. In RWM and RDM, unrealistic behaviors such as sharp turns, sudden stops, and sudden accelerations may frequently occur [21]. Gauss-Markov Model(GMM) is proposed to prevent these problems [26]. In this model, V_{t+1} is correlated with V_t where V_t is the velocity at time t . An example GMM trajectory is shown in Fig. 8.

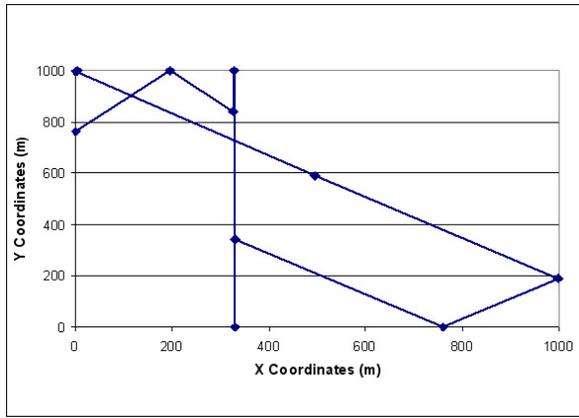


Fig. 7. An Example Trajectory of RDM.

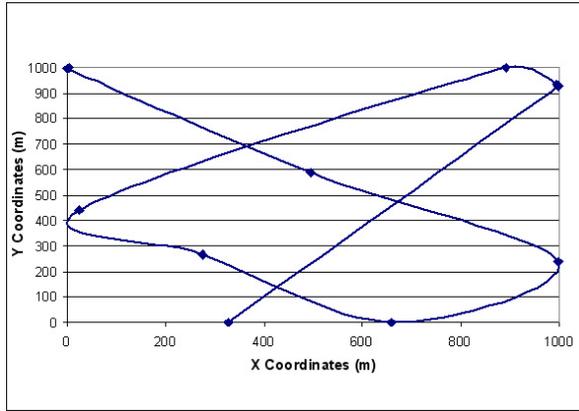


Fig. 8. An Example Trajectory of GMM.

IV. RESULTS

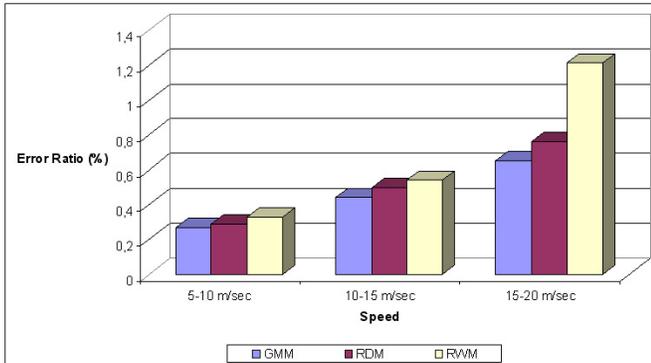


Fig. 9. Error Ratio of STA.

The static cluster based tracking algorithm (STA) and dynamic cluster based tracking algorithm (DTA) are implemented in the *ns2* simulator version 2.31. The clusters and backbone in STA are created by the algorithm proposed in [18], however other clustering algorithms can also be applied. The backbone of DTA is a simple spanning tree rooted at the sink. We generated randomly connected networks with 50,

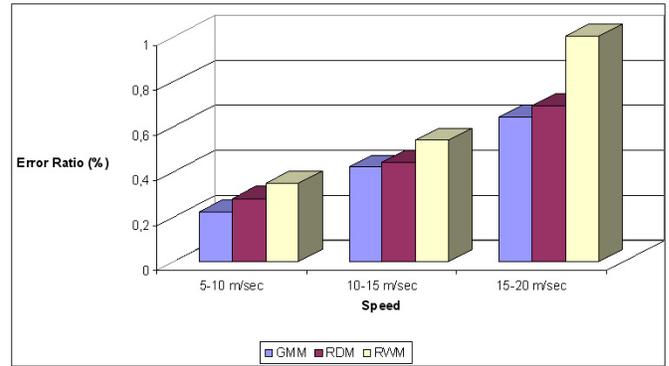


Fig. 10. Error Ratio of DTA.

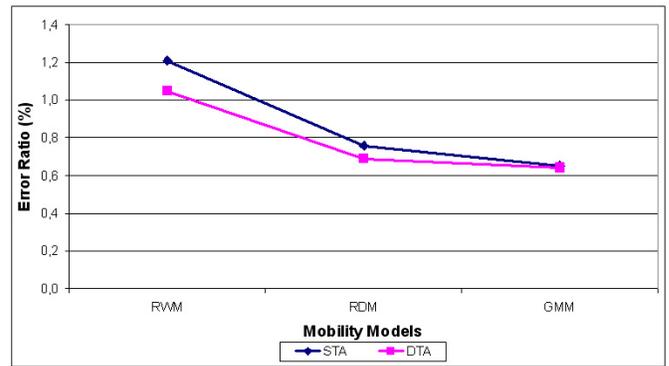


Fig. 11. Comparison of Error Ratios.

100, and 150 uniformly distributed nodes. The mobility model of target is selected either as RWM, RDM or GMM to measure the tracking accuracy. The scenario files of mobility models are generated with *ANSim*. Also the speed of the target is varied to measure the detection performance of the algorithms. For each scenario, a lower and an upper bound speed is determined. The speeds are respectively chosen from 5 m/s to 10 m/s, 10 m/s to 15 m/s, 15 m/s to 20 m/s. IEEE 802.11 radio and MAC standards readily available in *ns2* simulator are chosen for lower layer protocols and trilateration is used as the localization method. In this technique, at least three of the nodes in a cluster must detect the object in order to calculate the position of the object and the simulation time is 200 s.

In our simulations, we used the error and miss ratios as the performance metrics. If the distance of the calculated position of a node to the origin at time t is c_t and the distance of the real position to the origin at time t is r_t , the error ratio (e_t) at time t can be calculated as: $e_t = |c_t - r_t| / r_t$. To find the error ratio, we averaged the measured values during the simulation time. If the target's signal at t is not collected by at least 3 nodes belonging to the same cluster, the target's location can not be calculated, thus the target is missed at time t . The miss ratio of the target is the percentage of miss to the total signal dissipated by the target.

Firstly, we measured the tracking accuracy of STA and DTA against mobility models to compare the effects of RWM, RDM

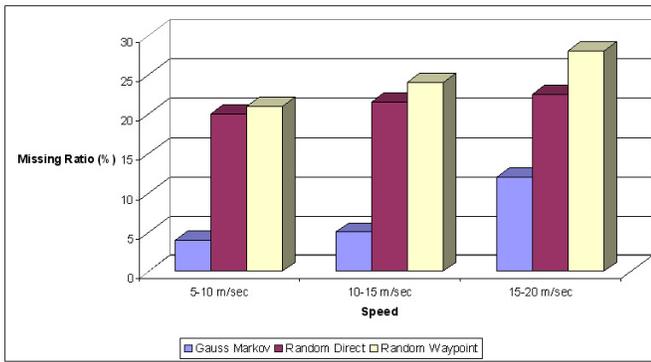


Fig. 12. Miss Ratio of STA.

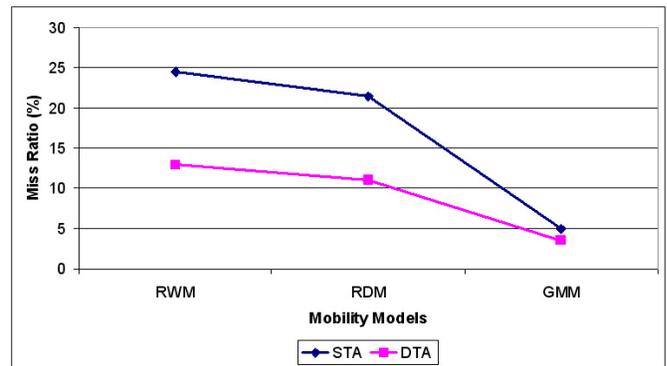


Fig. 14. Comparison of Missing Ratios.

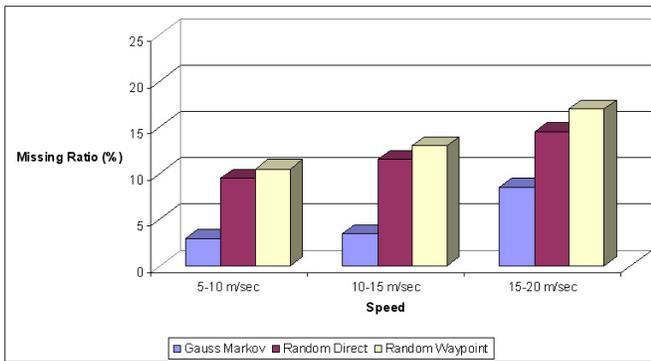


Fig. 13. Miss Ratio of DTA.

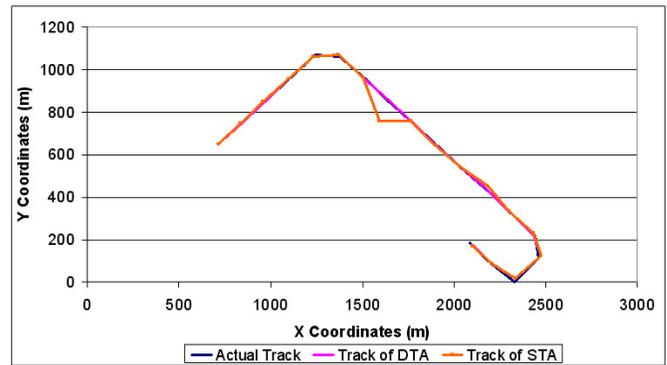


Fig. 15. Comparison of Trajectories.

and GMM. The error ratio of the STA and DTA against various mobility models and speeds are shown in Fig. 9 and Fig. 10. Regardless of the implemented tracking algorithm, the error ratio measured when tracking the target with GMM is smallest and RWM is the highest as seen in Fig. 9, Fig. 10 and Fig. 11. Generally, the measured error ratios are very small, at the worst case it is equal to 1.2 %. On the other hand difference of miss ratios between mobility models have significant values and reach approximately 20 % as shown in Fig. 12, Fig. 13, and 14. The main reason for this difference between mobility models is the sharp turns and sudden accelerations. The miss ratio difference between models obviously shows that accurate tracking of the target with a random mobility becomes very hard as the target moves faster.

Fig. 11 and Fig. 14 depicts the comparison of the tracking accuracy of STA and DTA more clearly. The measurements are collected for speeds 15 - 20 m/s. Both of error and miss ratios of STA are higher than those of DTA as seen in Fig. 11 and Fig. 14. In random models, the difference between error and miss ratios of algorithms are higher, on the other hand, algorithms calculate approximate trajectories in GMM. For the target with RWM, approximately 1/4 of the signals dissipated by the target is missed by STA. In STA, clusters are constructed before the target enters the region whereas the clusters are dynamically created in DTA by using RSS values obtained from the target. For this reason, the chance of locating the nodes which belongs to the same cluster

which are geographically close to the target is higher in DTA. As mentioned before, in the trilateration technique, at least 3 nodes of the same cluster must detect the target for localization. To illustrate the tracking performance of DTA and STA, a sample scenario with RDM is applied to both of the algorithms. Fig. 15 shows a sample trajectory of the target and the trajectories produced by STA and DTA for speeds 15-20 m/s. The trajectory generated by DTA is very close to the real trajectory, on the other hand, trajectory of the STA has some deviations as seen in Fig. 15.

Lastly, we compare the energy consumptions of STA and DTA as shown in Fig. 16. In STA, clusters are constructed during network deployment thus there is a cost of initial energy consumption of clusters. However DTA has no initial cluster formation but clusters are dynamically constructed when target is sensed. As the simulation time elapses from 100 s to 500 s, the energy consumption of DTA increases linearly with a higher slope than STA as shown in Fig. 16. The total energy consumption of DTA reaches to approximately 3 times of the energy consumption of STA. From the energy consumption values, we can state that the initial energy cost of STA can be seen at 100.s, but dynamic construction of the clusters applied in DTA is a very expensive operation compared to the energy consumption of STA can be seen at 500.s.

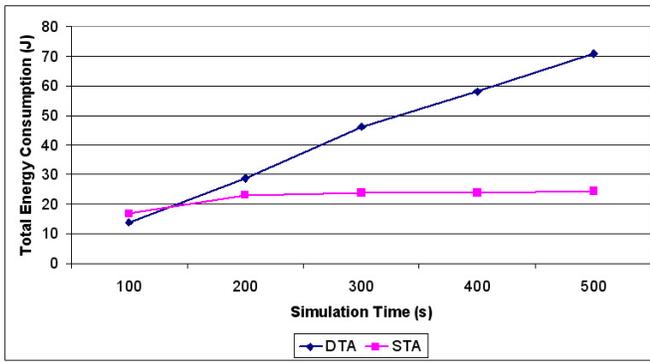


Fig. 16. Comparison of Energy Consumptions.

V. CONCLUSIONS

We evaluated cluster based target tracking algorithms against various mobility models and target speeds. A static cluster based and a dynamic cluster based algorithms are illustrated and implemented on *ns2*. We used RWM, RDM and GMM as the mobility models and described them by plotting a sample trajectory from each of them generated by *ANSim*.

The error ratios of both algorithms are 1.2 % at the worst case. When the speed of target is 15-20 m/s and the mobility model is RWM the miss ratio of STA becomes 20 %. However, when same conditions are applied, the miss ratio of the GMM is the half of RWM's miss ratio. Independent of the mobility model and speed of the target, DTA always produces smaller error and miss ratios than STA. We also illustrated the tracking accuracy of DTA by plotting a sample RDM trajectory, the trajectory generated by DTA and STA at the same graph.

A comparison of the two algorithms terms of their tracking accuracies and energy consumptions is shown in Fig. 17. Based on the extensive simulations, our general conclusion is that the dynamic clustering based target tracking is favorable for accurate tracking but consumes more energy than static tracking based clustering. Secondly, tracking of the targets moving with random models is harder than the targets moving with a correlated pattern. Our work is ongoing and we are planning to design static cluster based algorithms with improved tracking accuracy and dynamic cluster based algorithms with lower energy consumptions.

Algorithms	Tracking Accuracy			Energy Consumption
	RWM	RDM	GMM	
STA	LOW	LOW	HIGH	LOW
DTA	MEDIUM	MEDIUM	HIGH	HIGH

Fig. 17. Comparison of Algorithms.

REFERENCES

[1] W. Yang, Z. Fu, J. Kim, and M.-S. Park, "An Adaptive Dynamic Cluster-Based Protocol for Target Tracking in Wireless Sensor Networks", in *Proc. of WAIM 07*, 2007, pp. 157-167.

[2] W. Chen, and J. Hou, "Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks", *IEEE Transactions on Mobile Computing*, vol. 20, 2004, pp. 258-271.

[3] R. R. Brooks, P. Ramanathan, and A. M. Sayeed, "Distributed Target Classification and Tracking in Sensor Networks", *IEEE Signal Processing Magazine*, vol. 91, 2002, pp. 1163-1171.

[4] S. Suganya, "A Cluster-Based Approach for Collaborative Target Tracking in Wireless Sensor Networks", in *Proc. of ICETET 08*, 2008, pp. 276-281.

[5] H. Yang, and B. Sikdar, "A Protocol for Tracking Mobile Targets Using Sensor Networks", in *Proc. of SNPA 03*, 2003, pp. 7181.

[6] E. Olule, G. Wang, Minyi Guo, and Mianxiong Dong, "RARE: An Energy-Efficient Target Tracking Protocol for Wireless Sensor Networks", in *Proc. of ICPPW 07*, 2007, pp. 76.

[7] Y. Xu, and W. Lee, "Compressing Moving Object Trajectory in Wireless Sensor Networks", *International Journal of Distributed Sensor Networks*, vol. 3(2), 2007, pp. 151 - 174.

[8] R. Goshorn, J. Goshorn, D. Goshorn, and H. Aghajan, "Architecture for Cluster Based Automated Surveillance Network for Detecting and Tracking Multiple Persons", in *Proc. of ICDCS 07*, 2007, pp. 219-226.

[9] W. Zhang, and G. Cao, "DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks", *IEEE Transactions on Wireless Communications*, pp. 1689-1701, 2004.

[10] Z. Wang, H. Li, X. Shen, X. Sun, and Z. Wang, "Tracking and Predicting Moving Targets in Hierarchical Sensor Networks", in *Proc. of ICNSC 08*, pp. 1169-1173, 2008.

[11] K. Mechitov, S. Sundresh, Y. Kwon, and G. Agha, "Cooperative Tracking with Binary-Detection Sensor Networks", in *Proc. of ACM Sensys 03*, pp. 332-333, 2003.

[12] S. Bandyopadhyay, E. J. Coyle, "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks", in *Prof of INFOCOM 03*, vol. 3, 2003, pp. 1713-1723.

[13] S. Ghiasi, A. Srivastava, X. Yang, M. Sarrafzadeh, M., "Optimal Energy Aware Clustering in Sensor Networks", in *Proc. of MDPI 02*, 2002.

[14] H. Huang, J. Wu, "A Probabilistic Clustering Algorithm in Wireless Sensor Networks", in *Proc. of VTC 2005*, vol. 3, 2005, pp. 1796-1798.

[15] Y. Jin, L. Wang, Y. Kim, X. Yang, "EEMC: An Energy-Efficient Multi-Level Clustering Algorithm for Large-Scale Wireless Sensor Networks", *Elsevier Computer Networks*, 2007.

[16] F. Kuhn, T. Moscibroda, R. Wattenhofer, "Fault-Tolerant Clustering in Ad Hoc and Sensor Networks", in *Proc. of ICDCS 06*, 2006, pp. 68-78.

[17] C. Li, M. Ye, G. Chen, J. Wu, "An Energy-Efficient Unequal Clustering Mechanism for Wireless Sensor Networks", vol. 16(1), 2005, pp. 3-17.

[18] Erciyas, K., Ozsoyeller, D., and Dagdeviren, O, "Distributed Algorithms to Form Cluster based Spanning Trees in Wireless Sensor Networks", in *Proc. of ICCS 08*, 2008.

[19] O. Younis, and S. Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach", in *Proc. of INFOCOM 04*, 2004, pp. 629-640.

[20] G. Tel, "Introduction to distributed algorithm", in *Press. of Cambridge University*, 2000.

[21] F. Bai, and A. Helmy, "A Survey of Mobility Models in Wireless Adhoc Networks", *Wireless Ad Hoc and Sensor Networks*, Kluwer Academic Publishers, 2004.

[22] ANSim: Ad-Hoc network simulation, Available: <http://www.ansim.info/>.

[23] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols", in *Proc. of ACM/IEEE Mobicom 98*, 1998.

[24] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-Hoc Networks", in *Proc. of MobiCom 99*, 1999, pp. 195-206.

[25] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser, "An Analysis of the Optimum Node Density for Ad hoc Mobile Networks", in *Proc. of ICC 01*, 2001.

[26] B. Liang, and Z. J. Haas, "Predictive Distance-Based Mobility Management for PCS Networks", in *Proc. of INFOCOM 99*, 1999.