Energy Efficient Target Tracking in Heterogeneous WSNs Using a Combination of Activation Mechanism and Prediction Method

Fatemeh Aghaeipour[1], Vahid Sattari-Naeini[2], Majid Mohammadi[3]
1- Shahid Bahonar University of Kerman / Department of Computer Engineering, Kerman, Iran
Email: ff.aghaei@yahoo.com
2- Shahid Bahonar University of Kerman / Department of Computer Engineering, Kerman, Iran
3- Shahid Bahonar University of Kerman / Department of Computer Engineering, Kerman, Iran

ABSTRACT:
Tracking a mobile target is one of the primary applications in Heterogeneous Wireless Sensor Networks (heterogeneous WSNs). On the one hand, tracking without complex processing as well as achieving a high degree of accuracy and energy efficient consumption are critical requirements for these applications in a network area. On the other hand, artificial intelligence method provides adaptive mechanisms that present intelligent behavior in complex and dynamic environments like WSNs. In this paper, artificial neural networks are deployed to estimate target location. For this purpose, suggested beacon signals are able to facilitate distances estimation by which the network area is learned. Moreover, by analyzing the most probable region predicted for the next target location, a tracking window is created and sensors are dynamically clustered. Afterwards, sensors turn to the required mode to both preventing energy consumption and performing appropriate actions. The simulation results demonstrate the effectiveness of the proposed method.

KEYWORDS: wireless sensor networks, target tracking, artificial neural networks, tracking window.

1. INTRODUCTION

A Wireless Sensor Network (WSN) is an interconnected system of a large number of tiny sensor nodes randomly distributed in an interested area to support different types of sensing functionalities. Sensing nodes are equipped with a microcontroller, a sensing module, a small battery, and a radio transceiver. These sensors can collaborate with each other and form a sensor network; they collect data about an event and report them to a data collection node called sink or base station. In homogeneous WSN, all sensor nodes are identical in terms of battery energy and hardware complexity; but against it, a typical heterogeneous WSN is made of a large number of low-end nodes (normal nodes) and a few number of high-end nodes. The low-end nodes are both inexpensive and source-constrained whose main tasks are sensing and reporting data, while the high-end nodes provide data filtering, data fusion, and communication tasks; they are more expensive and more capable compared with the normal nodes, they may be equipped with a more powerful microprocessor or a memory with more capacity. These nodes can communicate with the sink node via a higher process throughput and a longer communication range [1]. The results of ref. [2] show that the presence of the high-end nodes in WSNs can increase network lifetime, reliability, and cost. The motivation is that more complex hardware and an extra source of energy can be embedded in a few number of nodes instead of all nodes. This results in both minimizing the network hardware cost and reducing the communication cost of the network; so a mixed deployment of these two kinds of nodes leads to achieve a tradeoff between cost and energy consumption. In target tracking scenario, when a mobile target is sensed by the sensors in the vicinity of it, its location is calculated using localization techniques as well as the cooperation of nodes; then after the required process, the aggregated data are sent to the sink node. In these applications due to the limited communication range of normal sensors, the sensors that are far from the sink, transmit their data through multi-hop communications using intermediate high-end nodes (relays) [3].

Dynamic, uncertain, and complex nature of WSNs bring about a lot of noise sources which impact on data measurements and calculations [4]. Moreover, tracking of mobile targets has lots of other challenges which should be handled including the accuracy of target localization, the method of data gathering, the capability of predicting next target location, and optimizing consumed energy of nodes. In a localization problem and from the energy saving viewpoint, if all the nodes are awake to detect a mobile target, there is a

تاریخ دریافت: ۱۳۹۶/۱۰/۲۹   تاریخ پذیرش: ۱۳۹۷/۱۱/۲۰

lot of waste of resources such as battery power as well as channel utilization, so sensor's operations should be properly managed [5]. Each sensor does different operations to detect an object, including basic sensing operation, memory or CPU process, and communication with other sensors and sink. Each of these tasks has a significant energy dissipation depending on the sensor type. On the other hand, all the sensors don't perform all the tasks at the same time. So, if each sensor can manage different modes of operation (considering demanded functionality), it will use its energy in a timely manner, and the network lifetime will be extended generally [6].

In this paper, we try to minimize energy consumption by both predicting the next target region and putting the sensors in the required modes. Tracking window and communicating window are introduced to determine sensors modes and saving energy. While in our previous work [31], we focused on estimating target location, in this work, the energy consumption in tracking applications is elaborated.

The remainder of this paper is organized as follows; Section 2 reviews some previous works on target tracking and saving energy methods in WSNs. Section 3 introduces the main parts of the model, including the setup phase tasks and the tracking phase operations. In section 4 we present some performance metrics and evaluate our method by simulation. Conclusion of this paper and addressing future works are provided in section 5.

2. RELATED WORKS

In target tracking application, a lot of sensor nodes are distributed in a network area to localize and track the trajectory of a single or multiple mobile targets. In general, the sensor nodes are randomly deployed by the vehicle robots or aircrafts. If the targets move into the sensing range of a sensor node, that node will sample the sensed signals to collect data about the target; then it communicates with other nodes within its communication range to deliver data to the sink node. In the central method, the sink performs the required process to estimate target location. Usually, there is no assumption about mobility models of the targets and it is desirable to calculate distance, speed, and direction of the target by tracking method [7].

Ref. [8] studied classification of the target tracking methods. Tracking methods can be classified as the following: tree-based target tracking methods, prediction-based methods, cluster-based methods, mobicast message-based tracking methods, and hybrid methods.

In the spanning tree-based algorithms, the nodes which detect the target, select a root and construct a spanning tree. The tree is dynamically configured during the target moves through the network area. This scheme reduces the overhead in terms of energy and information flow [9-10].

Prediction-based tracking algorithms aim for estimating the next position of the target based on the current target speed and direction of it; then using the predicted position, sensors activation mechanisms are applied to save energy [11-12].

In clustered networks, nodes are classified as cluster members and cluster heads. The cluster heads are responsible for managing intra-cluster operations and cooperating in inter-cluster tasks. In cluster-based target tracking algorithms, the member nodes detect the target and send the gathered information to their corresponding cluster head. Cluster heads collect all the information from the members and start to estimate the position of the target by using localization techniques. When the 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. This architecture of WSNs can also be classified as Static and Dynamic clustering methods [7].

In the static clustering, the cluster heads are assigned to the specific sensor nodes at the time of network formation; afterwards they cannot be changed. During the lifetime of WSN, all parameters of a cluster are fixed and preassigned [13].

Compared to the static clustering approaches, in the dynamic clustering, sensors do not statically belong to a cluster and may be included in different clusters at different times. Moreover, due to activating a limited number of clusters in the vicinity of the target, redundant data are suppressed and potential interference and competition at the MAC level are mitigated.

Mobicast methods provide a mechanism by which a multicast message is sent to a forwarding zone [14]. Hybrid methods are the tracking algorithms that fulfill the requirements of more than one type of target tracking [8].

In addition to the above-mentioned methods, Computational Intelligence (CI) methods provide adaptive mechanisms in complex and dynamic environments like WSNs. CI brings about flexibility, autonomous behavior, and robustness against topology changes, communication failures, and scenario changes [15]. ANN is one of the most useful tools in CI terminology which can be successfully applied to learn the properties of the sensor nodes. The capability of the ANNs to predict data helps to avoid unnecessary data communication and saving energy in WSNs [16].

In Ref. [17] an ANN was trained with LVQ (learning vector quantization) to address the problem of indoor wireless sensor location detection. Received Signal Strength (RSS) was used as the input of ANN and the location of each wireless node was the output of the neural network. This method results in more accurate estimations in areas far from the base stations. In practice, the RSS measurements are highly varied and unstable under the environmental noise as well as the mobility of sensor nodes. Therefore, using a multilayer perceptron (MLP) neural network is a good candidate to moderate noise effects in the localization process of the mobile node. A major benefit of using an ANN is that prior knowledge of the environment and noise distribution is not required. Also, neural networks generally provide more accuracy than other techniques such as Kalman filter. An MLP neural network brings the best tradeoff between accuracy and memory requirements among other types of neural networks [18].

In our previous study, a target tracking method, based on the artificial neural networks, was proposed considering noisy sources. It modeled a tracking scenario to two phases: offline and online phases. Beacon signals helped to initialize the neural network in an offline mode; then the neural network was used to learn a moving target regarding the noisy distances in an adaptive and online manner [31]. Since that model was suitable for practical use in large-scale WSNs, we deploy its mechanism for estimation purpose in this paper.

In recent years, there was a strong interest to combine ANN with energy efficient methods applied to WSNs. In [17], the authors proposed an intelligent method based on Self Organizing Map neural networks that optimize the routing in terms of both energy conservation and computation power of each node. ANNs have been used in [19] for dynamic power management of WSNs. The authors used ANNs to schedule duty cycle of the sensor nodes by event prediction. They proposed a neural method to decide which nodes and when they have to be woken up through prediction of the next event occurrence time. They considered that the occurrence of the next event is a non-stationary series that can be predicted using Wavelet Neural Networks. Sensor scheduling concerns with turning on the right sensors at the right time to achieve the best performance with the minimal energy consumption. Some previous works have proposed different sensor scheduling algorithms for tracking scenario in WSNs [20–24]. In this paper, two new concepts, namely Tracking window and sensor modes will be introduced to both managing the sensors operations and estimating the target location. The next region of the target is predicted and consequently, the sensors are changed to the required modes. It helps to minimize the total energy consumption of the overall network.

3. TRACKING SYSTEM DESCRIPTIONS

The proposed method consists of two phases: a setup phase and a tracking one. In this section, we describe these two phases in details. The setup phase refers to preparing the network area before the target entrance. When the target enters into the network area, the tracking process and online operations are commenced. In the following five sub-sections, the requirements of the model are elaborated and then in the three remaining ones, the mechanisms of the target location estimation is explained.

3.1. Network model

A heterogeneous WSN is considered in this model. In these networks, two kinds of nodes are used, including high-end nodes and normal ones; the high-end nodes are a few number of nodes with powerful energy sources, stronger computation, and communication power; while the normal nodes are in a large number but with low capabilities. We divide the main network area into fixed grids. The formation of each grid can be square, rectangular, or both according to the network dimensions as well as nodes capabilities. A lot of normal nodes are randomly distributed in each grid and a high-end node, which is called Grid Head (GH), is located in the center of that grid, as shown in "Fig 1". The number of high-end nodes is completely optional; the only important thing is that the presence of at least three high-end nodes in the network area is necessary. Although it is better to have a balanced distribution and each normal node be in the coverage of at least one GH, but it is not mandatory, because if the GHs can't directly communicate with the normal nodes (or vice versa if the normal nodes can't directly communicate with the GHs), a multi-hop communicating path is constructed and the connection is established indirectly. The base idea of this topology is like static clustering, but since all members of a grid don't simultaneously perform an action, we called it grid-based topology.

GHs are responsible for communicating with the sink node. Each normal sensor node transmits the sensing data to their respective GH. The GHs collect the data from the member nodes; then after the required process, the aggregated data are sent to the sink using a single-hop communication.

The sensing range (R_S) of a normal sensor node is a circle with a radius R_S centered at the location of it.

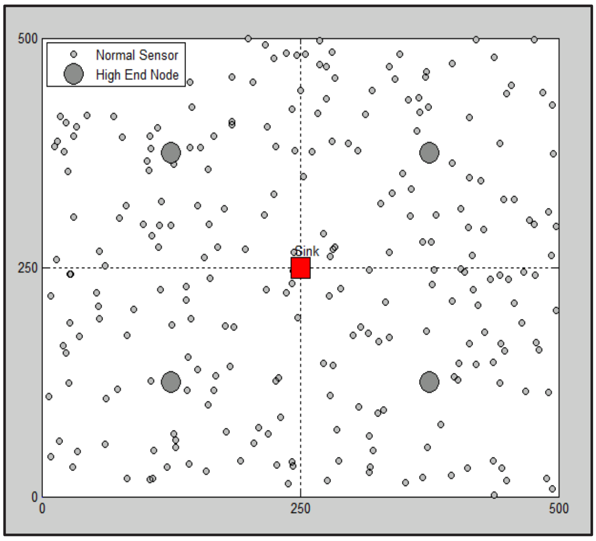

Fig 1. A grid-based heterogeneous wireless sensor network

In order to guarantee the direct communication between two normal nodes, we define communication range (R_c) twice R_s [7].

In this approach, we assume that the physical parameters of the geographical area are known. The other assumptions are listed as the following :

- Sink is located at the center of the main region;
- All sensor nodes assumed to be stationary;
- The target moves with a random waypoint model;
- Each normal sensor node is synchronized with its GH;
- GH nodes are equipped with GPS devices;
- There is no energy limitation in GHs;
- Each high-end node can directly communicate with other nodes.

3.2. Operational modes of normal sensor nodes

The main components of a sensor node are a micro controller, a transceiver, an external memory, and a sensing module. A power source also provides the required energy for these components. Turning off the micro controller and sensing module, in non-essential times, help to save more energy. Data communication consumes more energy than any other processes. The transceiver can be put into various operational states, including: Transmit, Receive, Idle, and Sleep; so it has different energy consumption levels. In the idle state, some functions, in transceiver hardware, are switched off in which the transceiver is ready to receive information; while in the sleep state all parts of the transceiver, except a radio trigger and a timer, are turned off and the sensor is able to immediately receive anything.

To reduce the energy consumption, it is not needed to run a sensor node at full operation mode all the times, so when one or more of the components are not in use, sensors working mode is defined [7, 25, 26]. There is five working mode in our approach, including: Sensing, Transmit, Receive, Idle, and Sleep modes. Typical features of the five modes are listed in "Table 1". In each mode, only necessary components are On and the unwanted ones are Off.

1) **Sense Mode**: The normal sensors can work in the Sensing mode to detect the moving target. Nodes can sense, transmit, and receive data packets. The micro controller is activated to perform the calculation. In fact, the sensors in this mode collect tracked data and then deliver them to GHs to determine the target location.

2) **Transmit Mode:** In this mode, sensors act as an intermediate device to deliver data to other nodes; so just transmit circuit is activated. The micro controller is in standby mode and it can be activated only when a control command reaches.

3) **Receive Mode:** It is like the previous mode unless the receiving part is ON in this mode.

4) **Idle Mode:** This occurs when a node is listening to the channel, but no transmission and receiption occurs. This process typically consumes more energy than Sleep mode.

5) Sleep Mode: All parts of a sensor are turned off except the necessary circuits required for waking the node up. The Sleep sensors would be awoken when either the timer sends an interrupt or the radio-trigger receives a control message [26].

Note that WSN is a dynamic system and the mode of sensors is continually changing. The energy consumption of sensor j under different working modes can be formulated as [25]:

Table 1. Sensor mode features

Sensor Modes	Sense	Transmit	Receive	Idle	Sleep
CPU	On	On	On	Off	Off
Sensing	On	Off	Off	Off	Off
Receiving	On	Off	On	On	Off
Transmitting	On	On	Off	Off	Off
Radio-trigger	Off	Off	Off	On	On
Timer	Off	Off	Off	Off	On
Power consumption (mw)	4	9	7	0.6	0.01

$$E_j = w_{se} \times t_{se} + w_{tr} \times t_{tr} \times B_{tr} + w_{\mathrm{re}} \times t_{\mathrm{re}} \times B_{\mathrm{re}} + w_{sl} \times t_{sl} + w_i \times t_i \tag{1}$$

where w_{se}, w_{tr}, w_{re}, w_{sl} and w_i represent the power consumption per time unit for Sense, Transmit, Receive, Sleep and Idle modes, respectively; t_i, t_{sl}, t_{se}, t_{tr} and t_{re} represent the unit times of being in the modes of: Idle, Sleep, Sense, Transmit (to send a unit packet) and Receive (to get a unit packet), respectively; B_{re} and B_{tr} represent the size of the packets received and transmitted by the sensor, respectively. It should be noted that a certain amount of energy is required to change the sensor modes due to activation or deactivation of electronic components; it can be considered as an average constant value to add to equation 1, but since the compared methods [7, 25, 26] have overlooked it in their equations, with no loss of generality, we also ignored it to get the same conditions for comparing the results.

The total consumed energy by sensors is calculated by:

$$E_{total} = \sum_{t_{start}}^{t_{end}} \sum_{j=1}^{n} E_j \qquad (2)$$

where E_i j = 1, 2, ..., n, denotes energy consumption of each single node, t_{start} is the time when the target enters the sensing field and t_{end} is the end time when the target leaves the network area [25]. Notice that the power consumption highly depends on the hardware platform of the sensor nodes. The sensor power consumption modes based on Berkeley MICA motes [27] are summarized in "Table 1".

3.3. Data acquisition

It is assumed that each normal sensor node is synchronized with the corresponding high-end node (GH). High-end nodes act as reference nodes in this stage. Distances of the mobile target from high-end nodes will be used as inputs of the neural network. These distances are calculated through the three following steps:

1) Each GH regularly transmits beacon signals to broadcast its position. Sensor nodes continuously listen to get beacons; once a sensor node receives at least three beacons from three different GHs, it stops listening. Sensor nodes receive beacons from GHs directly or via multi-hop communication. After receiving beacons, they synchronize themselves with the GHs and start to extract the location information of the GHs. Finally, they can calculate their distance from three GHs that their beacons have arrived sooner. Generally, each normal node knows its distance from at least three high-end nodes. Synchronization in this step certainly incurs processing and energy costs, but these costs are related to the network establishment at the set-up phase.
2) While a mobile target comes into the network area, each sensor node which has detected the target within its sensing range turns to the Sense mode and the distance of it and the target is calculated using RSS values. RSS Indicator is implemented in almost every sensor node and does not require extra hardware implementation [28].
3) After determining the distances of the sensors from the target and the GHs, the distance between the target and the GHs (D_i) will be computed by adding two vectors with regard to "Fig 2".

The distances between target and GHs are the basis measures of the proposed target location estimation algorithm. This is motivated by the trilateration technique. It basically uses the intersection of three circles, with radius distances, D_i, centered at the location of the target. In a real world scenario, the distances may be noisy [31] or the three circles never intersect at a common point. On the other hand, if a normal sensor gets beacons from three aligned GHs, the trilateration technique does not work [29]. Hence, to overcome these problems and to minimize the estimation error, a Feed Forward Artificial Neural Network (FF-ANN) technique is proposed in the next section.

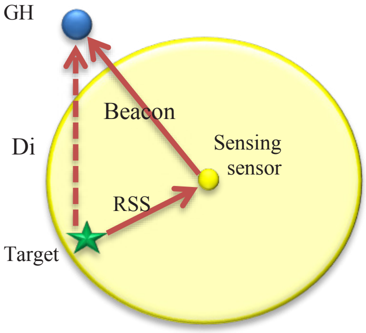

Fig. 2. Distance calculation between target and GH

3.4. Feed forward neural network

Deploying abstract sensor nodes, generating abstract target to learn network area, initializing and training FF-ANN are the main steps of the setup phase.

A grid sensor network based on real network dimensions and real positions of four high-end nodes is created. Abstract sensor nodes are placed on the intersection points of each grid. The target distances from the GHs are the neural network input values. The training data is collected by placing an abstract target at a lot of random positions and calculating its distances from high-end nodes as mentioned earlier. "Fig 3" shows the prepared network to train.

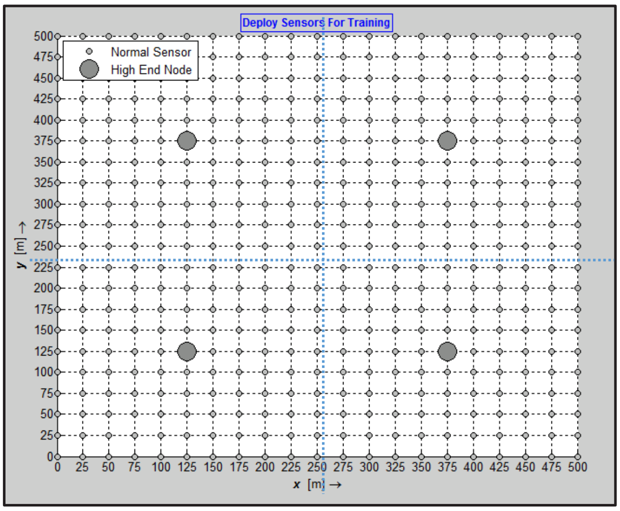

Fig 3. Abstract network to learn neural network area

The FF-ANN which is used in this approach is a three layer network composed of three neurons in the input layer, eight neurons in the first and second hidden layer, and two neurons in the output layer, as shown in "Fig 4". The exact coordination of target is as the output values. We achieved the best performance by this number of neurons. A supervised training method was performed using the Levenberg-Marquardt error back propagation algorithm. The research works related to ANN have used different configuration for their models. We found this network architecture by trial-and-error and by inspiration of the similar related works.

This trained network will be used in the following online phase of target tracking. One of the major benefits of using neural networks for target tracking is that prior knowledge of the target movement like velocity, acceleration, and movement direction is not required. More details of using FF-ANN has been described in our previous study [31].

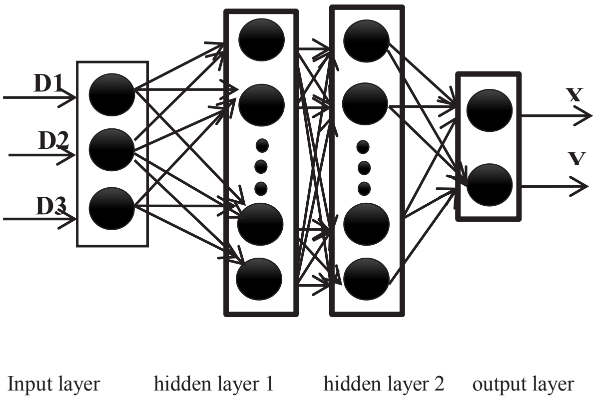

Fig 4. The architecture of the neural network

3.5. Target motion model

We consider that there is only one mobile target with the relative position (X, Y) in the network area. All normal nodes, which are located in the circle centered at the point (X, Y) with a radius R_S, can detect a target in their range. Different types of targets may have different kinds of motion characteristics. For example, a tank has a much larger maximum velocity than a human soldier. The novelty of this paper is supporting different moving velocities. We assume that the target moves with a constant velocity; however, in varying velocities, the algorithm can be run with the maximum speed in the history of target trajectory. When the target starts to go through the sensor network, it moves in a random path with random angle and random direction, so the relative position of the target (X, Y) is calculated as:

$$X = \mathrm{V} \times \mathrm{t} \times \cos\varphi + X_0 \qquad (3)$$

$$Y = \mathrm{V} \times \mathrm{t} \times \sin\varphi + Y_0 \qquad (4)$$

where V is the target velocity, t is the time interval between two target consecutive steps, φ is the momentum angle of the target and (X_0, Y_0) is the previous target location.

3.6. Tracking window and updating it

To ensure that all the required sensors are in Active mode, in order to detect the target, a tracking window concept is proposed. Actually, tracking window is the region of the network area performing tracking and sensing operation. It is created by predicting the most probable region to locate the target in the next step. It should be emphasized that the next region is predicted, not the precise position. In this way, all sensors which may detect the target in the next step are identified and included in the tracking window; they are activated and change their mode to the Idle one.

When a target comes into the tracking window, each of the Idle sensors that detect the target in the range of itself is triggered by the target's signals and changes its mode to the Sense one. Then it starts to collect target's data and send them to GHs. GHs perform required operations and transmit their results towards the sink node. The sink node estimates the current target location based on the received data; it also updates the tracking window and returns back its result. If some of the activated nodes are not included in the new tracking window, they will turn back to the Sleep mode.

Tracking window practically is a circle with radius R_{tw} centered at the latest target estimated location. As shown in "Fig 5", the tracking window for time slot $\mathrm{t}+1$ is centered at (X_t, Y_t) and R_{tw} is calculated as:

$$R_{tw} = \mathrm{V} \times \mathrm{t} + R_S \qquad (5)$$

where V is the velocity of the target; t is the time interval between two target consecutive steps; R_S is the sensing range of a normal sensor node. Dashed gray circle in "Fig 5" is the most probable region to locate the target in time slot $\mathrm{t}+1$.

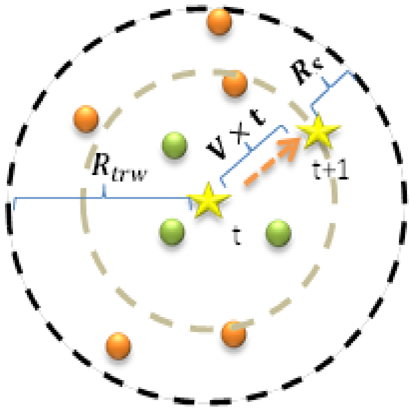

Fig 5. Tracking window for time slot t+1

If the target, in a period of time, is located in some places which have not been covered by any sensors, the target signals are not detected and it will be lost during that period. But the tracking window should be kept up-to-date to detect the target after passing the lost time when it appears again. In this situation, the radius of tracking window is calculated as:

$$R_{tw-lost} = \mathrm{V} \times \mathrm{t} \times t_{lost} + R_S \qquad (6)$$

where t_{lost} is the number of lost time slots. By increasing the lost time, more sensors turn to the Active mode, compensating weak coverage. "Fig 6" shows a situation that the target is lost for three sequential time interval.

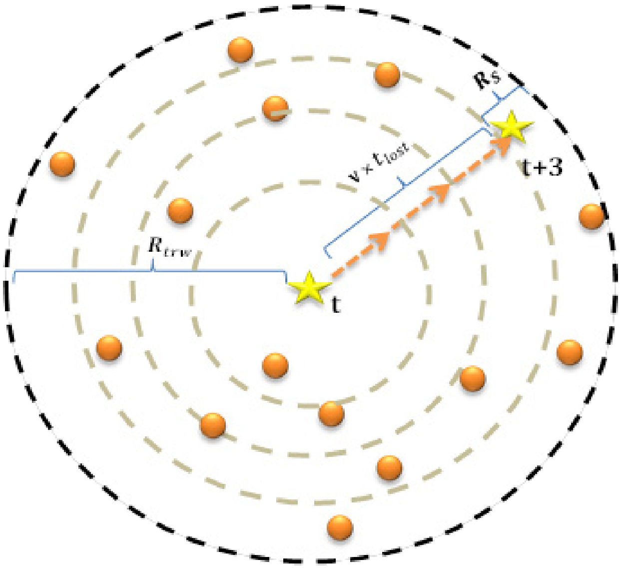

Fig 6. Tracking window for lost time 3

3.7. Dynamic clustering and update communication path

When some sensors inside the tracking window detect a target, they should deliver their sensed data to the sink node so that the consumed energy of the communication operations be as low as possible. Therefore, the sensors of tracking window are grouped into several clusters to avoid redundant transmissions. Each cluster contains a temporary cluster head (TCH) which acts as a gateway for the tracking window. A sensor node with the highest remaining energy is selected as a TCH, then the other nodes belong to the tracking window which are in the communication range of that selected TCH become its cluster members. This process continues until all sensors of the tracking window are organized in clusters. A single or multiple TCH is chosen in the tracking window according to communication range and remaining energy of sensor nodes. Because of the target mobility, the tracking window is also moving; so clusters structure is dynamic and TCHs and members are regularly updated. The dynamic cluster structure prevents heavy traffic load on cluster heads.

TCH is responsible for aggregating cluster members data and then transmitting them to the corresponding GHs. TCH(s) may not be able to communicate with corresponding GHs directly, so some intermediate nodes have been used to relay packets. The GH, related to each TCH, chooses the most appropriate nodes (among outer nodes of tracking window) by its knowledge about the members and sends them the wake-up message to change their mode into Idle for the purposes of relaying data in next time. These nodes, which are chosen by minimum hop count metric as well as the highest remaining energy, are called communicating nodes. According to these statements, we can briefly say after detecting the target, the sensing sensors send their collected data to their TCH then TCH(s) forwards data to the corresponding GHs (directly or indirectly), and at the last step, GHs deliver data to the sink node to do the final estimation operation.

It is clear that a balanced distribution, which each normal node be in the coverage range of at least one GH, has less cost; in other words, indirect communications lead to consuming more energy, because in this situation, some sensor nodes spend more time in the modes of Receive and Transmit in which the most energy is consumed.

3.8. Target tracking algorithm

During the setup phase, all GHs obtain information of their respective sensors. By entering a target into the network area, the tracking algorithm, in an online manner, is commenced; the target starts to walk through the sensor network following the mentioned target motion model. At the beginning, when there is no target location estimation, the tracking window is composed of all boundary sensors. The boundary sensors are in the Idle mode until they detect the target in their range. Once they receive the first signals, they change their mode into the Sense one. Then, as mentioned in data acquisition section, the sensing sensors start to compute the inputs of neural network by measuring RSS values. They send the results to their TCH. TCHs transmit data towards the sink through GHs. The sink node estimates the target location using the received data and the trained FF-ANN. After the new estimation, the tracking window is updated, new TCHs are assigned by reset clustering, and the multi-hop path is rebuilt for the next time. If the target is not detected in the range of any sensor, updating operations are done considering the lost time. The lost time is also incrementing based on the last target location estimation. The target tracking algorithm is described in "Fig 7".

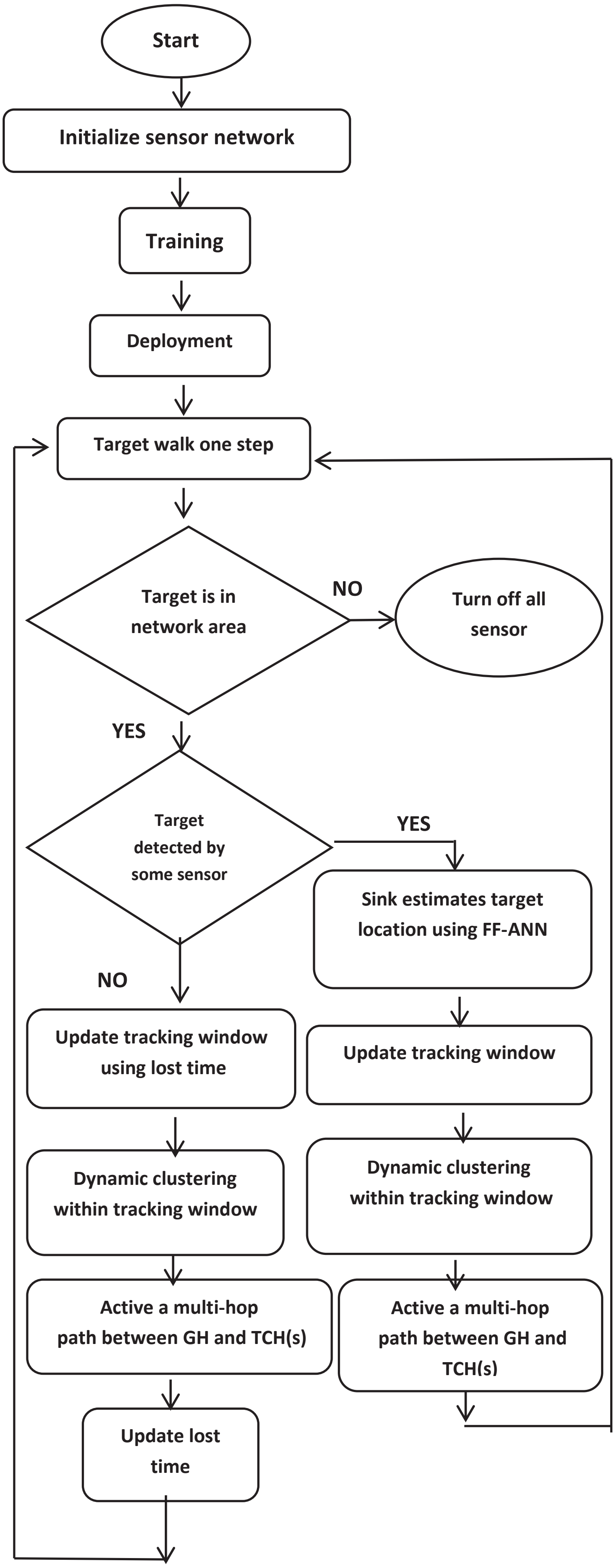

Fig 7. Target tracking algorithm

In this section, the requirements for the performed experiments and comparison results are elaborated.

4.1. Simulation parameters

The performance of the proposed algorithm is evaluated using MATLAB tools. The network model considered for simulation is a square area ($500\ m \times 500\ m$), divided into four grids and a high-end node is placed at each grid center; moreover, 256 normal sensor nodes are randomly deployed over the grids. The sensing coverage range is assumed to be 30 m and the communication range is 60 m and the time interval is set to one second. Network parameters are shown in "Table 2".

Table 2. System parameters

Parameter	Default Value
Network size	$500\ m \times 500$ m
Number of nodes	256
Simulation time	75 second
Sensor topology	random uniformly
Sensing range	30 m
Communication range	60 m
Packet size	40byte
Target velocity	$5\ {}^{\mathrm{m}}/_{\mathrm{s}}$

Two metrics are employed in order to compare the performance of the neural network tracking method. The first one is the average Euclidean distance between the estimated coordinates and the actual coordinates defined as:

$$avg\ distance\ error = \frac{\left(\sum_{t_{start}}^{t_{end}} \sqrt{\left(X_{real}^{\,t} - X_{est}^{\,t}\right)^2 + \left(Y_{real}^{\,t} - Y_{est}^{\,t}\right)^2}\right)}{\#\ of\ estimates} \tag{7}$$

where $(X_{real}^{\,t}, Y_{real}^{\,t})$ is the coordinates of real target location in time slot t ; $(X_{est}^{\,t}, Y_{est}^{\,t})$ is the coordinates of estimated target location in time slot t ; t_{start} is the start time, when the target enters the network area; t_{end} is the end time, when the target leaves the network area; and $\#\ of\ estimates$ refers to the number of target steps.

The second metric is the Root Mean Square Error (RMSE). Using this metric, the localization error for the X and Y coordinates is calculated separately as the following equations:

$$RMSE_X = \sqrt{\sum_{t_{start}}^{t_{end}} \left(X_{real}^{\,t} - X_{est}^{\,t}\right)^2 / \#\ of\ estimates} \tag{8}$$

$$RMSE_Y = \sqrt{\sum_{t_{start}}^{t_{end}} \left(Y_{real}^{t} - Y_{est}^{t}\right)^2 / \#\ of\ estimates} \qquad (9)$$

By combining the $RMSE_{\,X}$ and $RMSE_{\,Y}$ values as equation (10), $RMSE_{\,NET}$ is calculated; it describes the total net error. An interesting characteristic of the RMSE is that it is biased towards large errors; a large error makes a larger contribution in RMSE than in average distance error [30].

$$RMSE_{\,NET} = \sqrt{RMSE_{\,X}^{\;2} + RMSE_{\,Y}^{\;2}} \qquad (10)$$

"Fig 8" illustrates an example of estimation with the neural network technique. The black line shows the actual trajectories and the red line is the estimated path of the moving target. Yellow circles represent the Sensing sensors during the tracking operation.

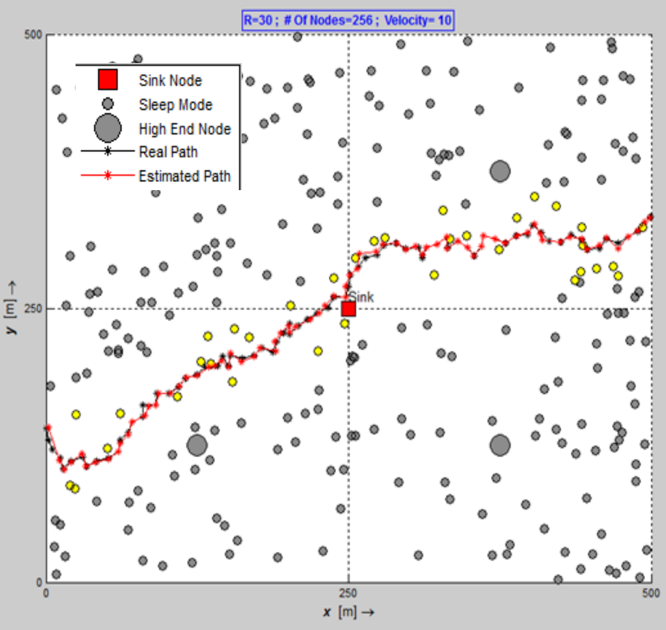

Fig. 8. Target tracking using FF-ANN

The output data have been compared with the actual data set to compute the tracking error. "Fig 9" shows the $avg\ distance\ error$ calculated using equation (7) during 75 second of tracking process.

4.2. Comparison of tracking error

The techniques used to evaluate the tracking errors are trilateration [29] and Kalman filter [30]. Trilateration is an impeccable technique for localizing; when the accurate distances of the target to at least three reference nodes are available, this technique is used and the intersection of the three circles determines the target location. However, the trilateration precision is highly dependent on the accuracy of measured distances. Hence, the trilateration technique is prone to error when noise sources impact on measurement.

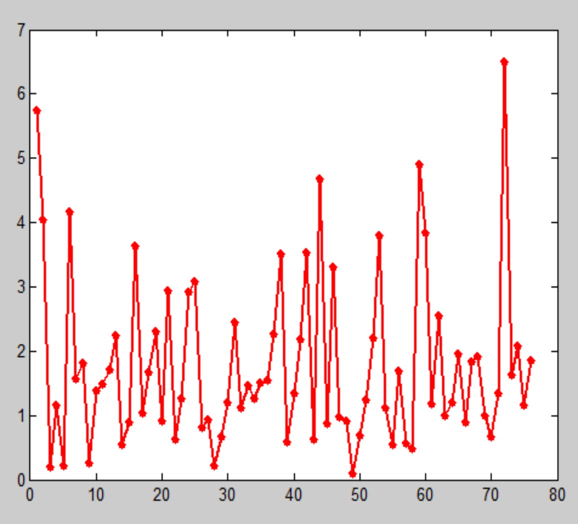

Fig. 9. Euclidean distance between estimated and real location

On the other hand, the Kalman filter technique is an iterative state estimator which is widely used to locate an object with noisy measurements. The Kalman filter uses the measured position of the target's centroid as well as the previously estimated state to determine the position of the centroid in the next step.

"Table 3" shows statistical results of these two methods and the proposed tracking approach based on the FF-ANN. As the results show, FF-ANN has been performed more efficiently than the Kalman filter and the trilateration in noisy environments; because FF-ANN learns the network area considering the noisy distances. A major benefit of using a neural network is that the prior knowledge of target movements like velocity, acceleration, and movement direction is not required.

Table 3. Comparison Results

Method	$RMSE_{\,X}$	$RMSE_{\,y}$	$RMSE_{\,NET}$	$avg\ distance\ error$
FF-ANN	1.65	1.55	2.26	1.54
Trilateration	2.05	2.33	3.10	1.81
Kalman filter	6.65	6.02	8.97	8.16

"Fig 10" shows the effect of different target velocities on the net error ($RMSE_{\,NET}$) in the three approaches. The statistical result shows that in different velocities, FF-ANN has better results than the other. It is a big achievement that we don't have velocity limitation in this approach.

4.3. Comparison of energy

In this section we discuss and compare just the energy consumption of the tracking process. With no loss of generality, the costs of set-up phase operations are omitted.

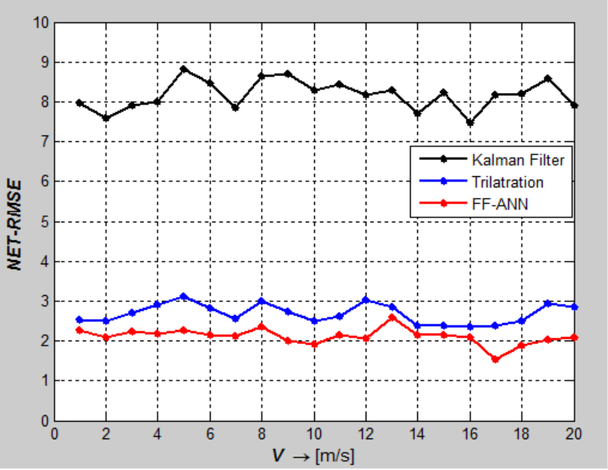

Fig 10. The effect of target velocity on MSE error in FF-ANN

While the target is moving in the network area, more sensors are activated and as "Fig 11" illustrates, the amount of consumed energy, which is measured using equation (2), is increased.

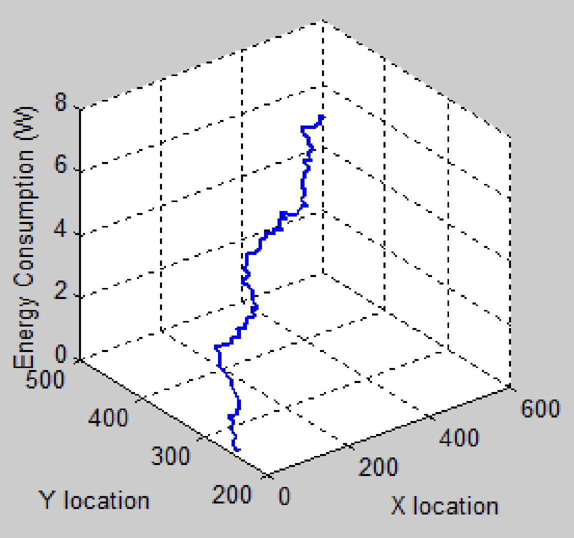

Fig 11. Energy consumption of target movement

We evaluate our proposed method in terms of energy consumption by comparing with two related methods including distributed sensor activation algorithm (DSA2) [26] and predicted region sensor activation algorithm (PRSA) [25]. DSA2 combines probabilistic sensor activation with sensors' work mode. In this algorithm, all sensors in the field are activated with a probability to detect targets or sleep to save energy. On the other hand, PRSA algorithm predicts the moving region of target in the next time interval, then using the activation strategy the fewest essential number of sensor nodes within the predicted region will be activated to monitor the target. These two algorithms and our proposed approach use common principles to save energy, i.e., sensor activation mechanism, prediction method, and sensor modes concepts.

When the target moves in the network area with a specific velocity, the network energy consumption is calculated according to equation (2). In high velocities, the number of target steps, as well as the number of active sensors, decreases and it leads that the network energy consumption will reduce. "Fig 12" shows that the total consumed energy of FF-ANN method in different velocities is less than DSA2 and PRSA algorithms.

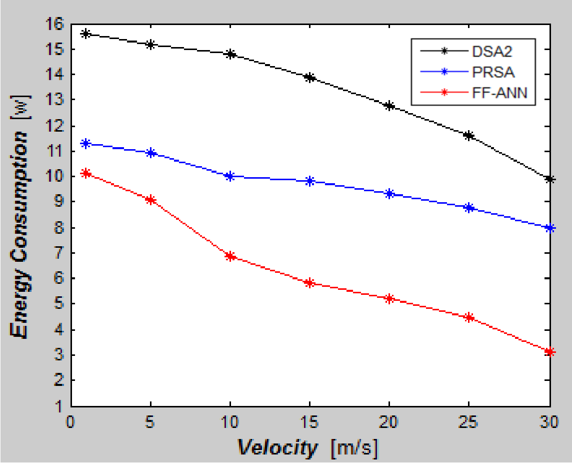

Fig 12. The effect of target velocity on the energy

"Fig 13" shows the effect of varied sensing range. When the sensing range is increased, the target will capture within the range of a certain active sensor for a longer period of time; On the other hand, by increasing the communication range, the number of members belonging to TCHs will increase and TCHs load traffic of more sensors; these reasons lead up more power consumption in high sensing range.

"Fig 14" shows the capability of our proposed method to support large scale networks. As shown, by increasing the number of deployed sensors in the network area, more energy will be consumed, but yet it is less than the two other methods.

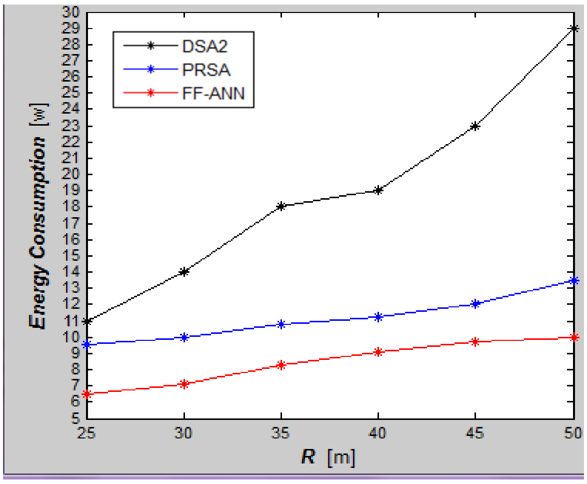

Fig 13. The effect of sensing range on the energy consumption

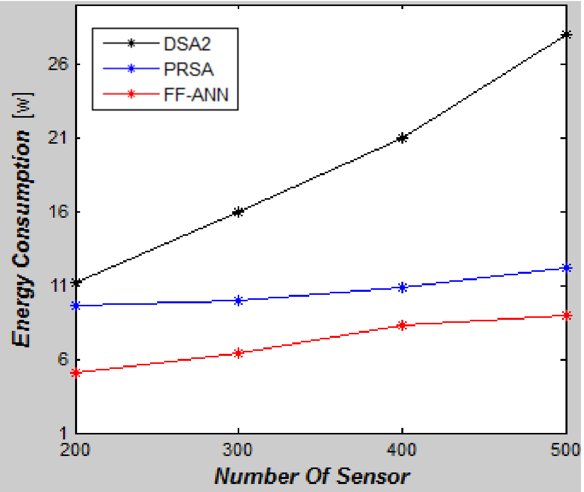

Fig 14. The effect of number of deployed sensors on the energy consumption

`ONCLUSIONS

In this paper, a grid-based topology for tracking a target in heterogeneous WSNs was proposed. Predicting the most probable region for locating the target in the next step and activating the sensors for specific tasks helped to save more energy. The proposed algorithm in each step activates only the essential sensors and turns them to the required mode; it leads to minimize the total consumed energy of the network. A feed forward neural network was used to learn the estimated target location. Simulation and comparison results showed considerable improvement in both accuracy and energy saving. Considering the energy required for changing the modes are our next research work.

5. REFERENCES

[1] L.Yu, N. Wang, W. Zhang, Ch. Zheng,"Deploying a Heterogeneous Wireless Sensor Network", Wireless Communications, Networking and Mobile Computing International Conference, Shanghai, Sept. 2007.

[2] D. Kumar1, R. B. Patel "Multi-Hop Data Communication Algorithm for Clustered Wireless Sensor Networks", International Journal of Distributed Sensor Networks, Article ID 984795, 10 pages, Feb 2011.

[3] S. Rhee, D. Seetharam, and S. Liu, "Techniques for Minimizing Power Consumption in Low Data-Rate Wireless Sensor Networks," in Proc. Of IEEE Wireless Communications and Networking Conference, Atlanta, GA, March 2004.

[4] C..F. Chiasserini, M. Garetto, "Modeling the Performance of Wireless Sensor Networks", Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, Italy, March 2004.

[5] M. Gholami, N.Cai, R.W.Brennan," An artificial neural network approach to the problem of wireless sensors network localization", Robotics and Computer-Integrated Manufacturing 29 (2013) 96–109.

[6] M. Zoghi and M. H. Kahaei, "Adaptive sensor selection in wireless sensor networks for target tracking" IET Signal Processing, vol. 4, no. 5, pp. 530–536, 2010.

[7] W. Zhou, W. Shi, X.Wang, K.Wang, "Adaptive Sensor Activation Algorithm for Target Tracking in Wireless Sensor Networks", International Journal of Distributed Sensor Networks Volume 2012, Article ID 515906, March 2012.

[8] M.Naderan, M. Dehghan, H. Pedram, V. Hakami "Survey of mobile object tracking protocols in Wireless Sensor Networks: a network-centric perspective", Int. J. Ad Hoc and Ubiquitous Computing, Vol. 11, No. 1, pp.34–63,2012.

[9] K. Ramya, K. Praveen Kumar, and Dr. V. Srinivas Rao, "A Survey on Target Tracking Techniques in Wireless Sensor Networks", International Journal of Computer Science & Engineering Survey, Vol.3, No.4, August 2012.

[10] W. S. Zhang and G. H. Cao. "DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks," IEEE Transactions on Wireless Communications, Vol. 3, No. 5, pp. 1689 1701, September 2004.

[11] W. S Zhang, G. H. Cao. "MobiHoc Poster: Optimizing Tree Reconfiguration to Track Mobile Targets in Sensor Networks," Mobile Computing and Communications Review, Vol. 7, No. 3, pp. 39-40, July 2003.

[12] Samarah, S.Al-Hajri, M. Boukerche, A.,"A Predictive Energy-Efficient Technique to Support Object-Tracking Sensor Networks", IEEE Transactions on Vehicular Technology, Vol. 60, NO. 2, pp. 656–663, 2011.

[13] Liang Xue, Zhixin Liu, Xinping Guan, "Prediction-based protocol for mobile target tracking in wireless sensor networks", Journal of Systems Engineering and Electronics Vol. 22, No. 2, pp. 347–352, 2011.

[14] E. Olule, G. Wang, M. Guo and M. Dong, "RARE: An Energy Efficient Target Tracking Protocol for Wireless Sensor Networks," 2007 International Conference on Parallel Processing Workshops, 2007.

[15] Wei-Peng Chen, "Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks", in

Mobile Computing, IEEE Transactions, Vol 3, Issue x3 , July-Aug. 2004.

[16] H. Yang and B. Sikdor, ”a protocol for tracking mobile targets using sensor network, sensor network protocols and applications”, in First IEEE International Workshop on Sensor Network Protocols and Applications, Anchorage, Alaska, pp. 71-81, 2003.

[17] Ogawa T, Yoshino S, Shimizu M, Sudah. A new indoor location detection method adopting learning algorithms. In: Proceedings of the first IEEE international conference on pervasive computing and communication (PerCom’03), IEEE Computer Society, 2003.

[18] Q. Huang, Ch. Lu, G. Roman."Mobicast: Just-in-time multicast for sensor networks under spatiotemporal constraints", in Proc. of the 2nd International Workshop on Information Processing in Sensor Networks, Palo Alto, CA, USA, pp. 442–457, 2003.

[19] V. Kulkarni, G. Kumar, “Computational Intelligence in Wireless Sensor Networks: A Survey”, IEEE communications surveys & tutorials, vol. 13, no. 1, first quarter 2011.

[20] H.Shahbazi, M.A Araghizadeh, M.Dalvi, “Minimum Power Intelligent Routing In Wireless Sensors Networks Using Self-Organizing Neural Networks”, IEEE International Symposium on Telecommunications, pp. 354—358,2008.

[21] Y. Shen, B. Guo, “Wavelet Neural Network Approach for Dynamic Power Management in Wireless Sensor Networks”, International Conference on Embedded Software and Systems, pp.376—381, 2008.

[22] Y. He, KP. Chong “Sensor scheduling for target tracking in sensor networks”. In: Proceedings of IEEE CDC. Atlantis, Paradise Island, Bahamas, pp 743–748, 2004.

[23] J. Chen, K. Cao, Y. Sun, X. Shen “Adaptive sensor activation for target tracking in wireless sensor networks”, In Proceedings of IEEE ICC. Dresden, Germany, 2009.

[24] MF. Huber, UD. Hanebeck, “Priority list sensor scheduling using optimal pruning”, In Proceedings of the 11th international conference on information fusion. Cologne, Germany, 2008.

[25] L. Yang, Ch. Feng, J.W. Rozenblit, H, Qiao, “Adaptive Tracking in Distributed Wireless Sensor Networks”, Proceedings of the 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems, March 2006.

[26] J. Chen, K. Cao, Y. Sun, “Distributed sensor activation algorithm for target tracking with binary sensor networks”, in Cluster Computing, Volume 14, Issue 1, pp 55-64, March 2011.

[27] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. “System architecture directions for network sensors”, Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX), pages 93–104, November 2000.

[28] M. Abdelhadi, M. Anan, M. Ayyash, “Efficient Artificial Intelligent-based Localization Algorithm for Wireless Sensor Networks”, Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications, Volume 3, Issue 5, May 2013.

[29] T. A. Malik, “Target Tracking In Wireless Sensor Networks”, A Thesis BE in Computer Science and Engineering, Maharshi Dayanand University, India, May 2005.

[30] A. Shareef, Y, Zhu, “Localization Using Extended Kalman Filters in Wireless Sensor Networks”, Kalman Filter Recent Advances and Applications book edited by Victor M. Moreno and Alberto Pigazo, ISBN 978-953-307-000-1, Published: April 1, 2009.

[31] F. Aghaeipoor, M. Mohammadi, V. Sattari-Naeini, “ Target tracking in noisy wireless sensor network using artificial neural network “, Proceedings of the 7th International Symposium on Telecommunications (IST), pp: 720 - 724 2014.