

Research Article

Recovery of Lost Target Using Target Tracking in Event Driven Clustered Wireless Sensor Network

Shailaja Patil,¹ Ashish Gupta,² and Mukesh Zaveri³

¹Faculty of Electronics and Telecommunication Engineering, Rajarshi Shahu College of Engineering, Tathawade, Pune 411033, India

²Faculty of Computer Science and Engineering, Jaypee University of Engineering and Technology, Guna 473226, India

³Faculty of Computer Engineering, Sardar Vallabhbhai National Institute of Technology, Surat 395007, India

Correspondence should be addressed to Shailaja Patil; shailaja_ptl@yahoo.com

Received 20 June 2014; Revised 22 November 2014; Accepted 22 November 2014; Published 14 December 2014

Academic Editor: Tzonelih Hwang

Copyright © 2014 Shailaja Patil et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The prediction based target tracking in wireless sensor network is being studied from many years. However, the presence of a coverage hole(s) in the network may hamper tracking by causing temporary loss of target, which is a crucial issue in mission critical applications. This raises the need for robust recovery mechanism to capture the lost target. However, the problem of recovery of lost target during tracking has received less attention in the literature. In this paper, we present an energy efficient recovery mechanism. The performance of proposed algorithm for lost target recovery using different tracking filters has been evaluated with and without awakening hole boundary nodes. The efficacy of the algorithm has been tested using various causes of losing target with and without energy saving modes. We have conducted exhaustive simulations for recovery of the lost target and presented the analysis of how the recovery gets affected with linear and nonlinear filters. From the simulation results and energy analysis, it is evident that the proposed recovery algorithm outperforms the existing work in the literature.

1. Introduction

Wireless sensor network (WSN) is emerging as an important area which is attracting considerable research interest due to advancement in low cost embedded processor and wireless transmission technology. Target tracking in WSN has been successfully implemented in applications like aqueous surveillance system for drinking water reservoir [1], wildlife habitat observation system [2], network to monitor the behaviour of glaciers [3], in-house person tracking [4], battlefield data collection network [5], and so forth. However, there are several challenges and limitations in developing tracking algorithm in WSN. As the sensor nodes have limited energy and bandwidth, they cannot be always active in sensing, computing, and data forwarding. The tracking algorithm should be energy efficient to prolong the network lifetime by switching nodes periodically in active and sleep mode. Moreover, tracking algorithm is vulnerable to loss of target due to anisotropic nature of WSN containing coverage hole(s), where some part of the area under observation may not be monitored.

Aforementioned limitations of node energy in sensor network raise the need for prediction of precise location of the target, so that number of nodes participating in tracking is as small as possible. The prediction is performed using different linear and nonlinear filters in WSN [6, 7]. Though there have been efforts towards improving performance in tracking by modifying these filters, there is no considerable amount of work done on tracking in presence of hole(s). The presence of hole badly hampers the tracking in the network. Hole is that region of the network where nodes are depleted. This is also referred to as communication void since it is an obstacle for communication [8]. Such holes make deep impact on the sensor network's performance. Once the target enters into the hole, tracking nodes cannot detect its position and in turn fail to provide the useful information about the target. When the target enters into the hole it gets temporarily lost. Such an event can be a dangerous situation in mission critical applications such as enemy tracking, harmful or wild animal tracking inside a habitat, and tracking a terrorist. Various causes of losing the target may be summarized as follows.

(1) *Communication Failure due to Coverage Hole(s) in the Network.* It may occur due to insufficient coverage in the network. The sensor nodes are randomly deployed in the area of interest. For example, if the inspected area is vast and precarious for human intervention such as dense forest or battlefield, then sensor nodes are dropped from aeroplane. Here, nodes may form anisotropic network with large coverage holes.

(2) *Node Failures.* The main cause of failure is battery exhaustion of node. In addition, it may occur due to physical damage of the nodes. For example, when the nodes are deployed in battle field, group of nodes may be destroyed by means of warheads such as grenades and bombs or physically capturing and destroying the nodes by enemy [9]. Similarly, when the nodes are deployed randomly through aeroplane, these may fall into pond resulting in failure, if they are not water resistant. Similarly, with a sudden outburst of fire in the forest, nodes in a particular region may be gutted by fire creating a hole in that region.

(3) *Location Estimation Errors.* If erroneous localization is performed and such coordinates are assigned to nodes, target's current location may be estimated wrongly by the tracking nodes. Similarly, the localization error may have cumulative effect on tracking [10].

(4) *Prediction Errors.* If erroneous location is predicted, wrong node and cluster may be alerted for tracking. Similarly, the delay in disseminating predicted location by intermediate nodes may miss the target as the corresponding cluster may not wake up timely.

(5) *Random Manoeuvring.* If the target is manoeuvring with sudden variation in speed, target may go undetected by the sensing nodes as the target object may flee the detection areas by the time they wake up.

(6) *Faulty Node.* If the node is incapacitated by adversary, it may stop working or give results which may significantly deviate from the results of the neighbour nodes [11], which needs to be detected immediately.

Thus, these facts need to be considered while developing the tracking with recovery protocol. Few researchers have addressed the issue of tracking in presence of hole and recovery of the lost target [10, 12–15]. To the best of our knowledge, none of the tracking algorithms mentioned in literature address how the role of filter affects recovery performance for the lost target. In this paper, we present how the filter based tracking algorithms perform in the presence of hole, with and without awakening hole boundary nodes. The performance of proposed recovery algorithm is compared with recovery algorithm presented in literature [12]. In addition, the recovery analysis is conducted in the presence of static and event driven clustered architecture.

The rest of the paper is organized as follows. Section 2 deals with related work in the literature. Section 3 introduces network architecture used for tracking, and proposed algorithm for recovery of lost target is presented in Section 4.

Performance evaluation and results are discussed in Section 5. Conclusion is presented in Section 6.

2. Related Work

In this section, the existing recovery mechanisms are reviewed for lost target during tracking. Typically, before initiating tracking, it is essential to detect the target entering into the network and its current location. For detection all sensor nodes either need to be awake all the time [16] or may periodically switch between active and inactive (sleep) state for energy conservation as explained in probabilistic sensing approaches [17]. Once the target is detected, it is necessary to compute its location. For prolonging the network lifetime minimum number of sensor nodes should participate in location estimation and tracking. Trilateration [18] and triangulation [19] are such techniques where minimum of three and two nodes, respectively, are sufficient for location estimation.

The lost target recovery can be performed using various tracking techniques. Hence, we review the tracking techniques in brief. The tracking algorithms are classified according to the underlying techniques. A detailed survey of tracking algorithms can be found in [20]. The prediction based tracking approaches either use prediction mechanisms based on target's movements, velocity, and direction as in [21–23] or use various linear and nonlinear filters. An optimal or linear filtering technique used in tracking is Kalman filter (KF) [6, 24]. An estimate of the state of the process is performed recursively so that each time mean of squared error is reduced while tracking a target with random trajectory. The KF assumes linear system dynamics, that is, linear state vector and Gaussian noise. For nonlinear system dynamics, the tracking performance of Kalman filter is degraded. To address this problem, researchers developed suboptimal extension of KF, namely, extended Kalman filter (EKF) [25], which is the most widely used approximate filter for tracking [25, 26]. Though EKF performs better than KF, it suffers from serious drawbacks [27].

The EKF linearizes the transformation by substituting Jacobian matrices for the linear transformations of the KF equations assuming that all the transformations are quasi-linear. Linearization can be applied only if Jacobian matrix exists. Also, the series approximations of the EKF algorithm can lead to poor representations of the nonlinear functions and probability distributions, so this filter may diverge.

To overcome these drawbacks another technique is devised for nonlinear and non-Gaussian system dynamics, which is known as particle filter (PF). In PF any probability distribution function (pdf) can be represented as a set of samples known as particles [7, 25]. Each particle has one set of values for the state variables, and the density of samples in the area of the state space depicts the probability of that region. Hence, in PF the posterior distribution is represented by a set of k weighted samples. These are updated recursively in time using the importance sampling method [25]. Besides these, there are other filtering techniques used in tracking such as unscented KF [28] and interactive multiple models [29].

The aforementioned tracking algorithms may suffer from temporary loss of target due to various reasons mentioned earlier. To circumvent this issue, recovery mechanisms have been devised in literature. For example, authors in [13] presented a cluster based distributed tracking protocol (DTP) on the principal of RADAR beam. This protocol describes a level based recovery mechanism for temporary loss of target, mentioned as follows.

(1) *First Level Recovery.* When the target is lost, the sensor-triplet which was tracking the target earlier switches from normal beam (r) to high beam (R). If this is successful, then the normal tracking mechanism is followed; otherwise second level recovery is initiated.

(2) *Second Level of Recovery.* In this level, the region of the search is increased and the nodes at a distance of p meters from the last node are activated; otherwise N th level recovery is initiated.

(3) *N th Level Recovery.* In this level of recovery the group of sensor nodes that are $(2N - 3)p$ meters away is activated to locate the target.

However, this mechanism does not consider past information of object motion. Authors in [14] have presented prediction based strategies for energy saving (PES) and recovery mechanism for the lost target. Three cases, namely, naive monitoring, continuous monitoring (CM), and scheduled monitoring (SM), for target tracking and recovery are discussed. The performance of PES is compared with aforementioned three cases. However, no optimal wake up is defined for recovery of the lost target.

As an improvement towards reducing the active nodes participation during recovery, authors in [14] have considered different motion features of objects for simulation. Location prediction is done by integrating the current information of location, velocity, and motion direction. When prediction fails the target is lost, and the network starts hierarchical recovery process according to its past record of motion. This kind of recovery mechanism wakes up a lot of nodes every time, resulting in increased energy consumption and network lifetime reduction.

Another recovery mechanism for clustered network for lost target is described in [10]. This protocol consists of four phases described as follows.

(1) *Declaration.* When target is lost, current cluster head (CH) checks its presence in the previous clusters. If it is not found, then it declares that the target is missing.

(2) *Search.* To avoid the false initiation of recovery mechanism, current CH checks the presence of target in its own cluster. If it is not present then active recovery is initiated.

(3) *Active Recovery.* This phase consists of different recovery levels. In the first level, the current CH wakes up all its single hop clusters. If it is not found here, then two hop clusters are woken up, and this process is repeated till the target is recovered.

(4) *Sleep.* On successful recovery of the target, the CH and cluster members that are participating currently in tracking remain awake and the remaining cluster members go to sleep mode.

However, this recovery mechanism wakes up large number of nodes during recovery. In the above case, as soon as the target is lost all single hop clusters are woken to detect it. Towards reducing the active participation of nodes, authors in [12] present another recovery mechanism. For recovering the target in the first level recovery, only the nearest one hop cluster to the predicted position is woken up instead of waking up all the single hop cluster heads. If the target is not detected in first level recovery, the two hop clusters nearer to the predicted location are woken up. If the target is not captured here also, then nearest three hop clusters are woken up in third level recovery. In this way less number of nodes needs to be activated for recovery of the lost target.

The aforementioned protocols use the prediction mechanisms based on its last location, velocity, angular movement, and so forth to locate the target in case of missing situation. However, to the best of our knowledge, hitherto, none of the recovery algorithms consider waking up hole boundary nodes or use nonlinear filter based prediction for target recovery. In this paper, we are proposing an energy efficient recovery mechanism for anisotropic networks. In the proposed work, we introduce two kinds of network scenarios, namely, wireless sensor hole aware network and unaware network. When the network knows hole boundary, it is called wireless sensor hole aware network (WSHAN), and if hole boundary nodes (HBNs) are unknown, then it is a wireless sensor hole unaware network (WSHUN). In WSHAN the base station (BS) stores information of the boundary nodes (BN). The performance of recovery mechanism has been evaluated with static and event driven clustered architecture. The event driven clustering is an energy saving architecture of the network as compared to static or dynamic clustered architectures [30, 31]. Details of formation of such network are explained in the following section. The significant contributions of this paper are as follows.

We propose an energy efficient, nonlinear filter based recovery mechanism and compare its performance with existing recovery algorithm [12] in presence of static and event driven clustered architecture.

We propose a recovery mechanism in WSHAN for linear and nonlinear filters. Since less number of nodes is woken up in recovery of the lost target with WSHAN as compared to WSHUN, the network energy is saved.

The energy saving in WSHAN is twofold, first due to the event driven clustered architecture and second due to waking up optimum number of nodes.

3. Network Architecture

The following section describes the network architecture and the energy model used in the recovery algorithm. WSN consists of randomly spread nodes in the area of interest. All these nodes communicate with neighbour node and form network through self organisation. For a distributed

operation, the network is divided into clusters [32], where only cluster head (CH) and one (or more than one according to application requirement) cluster member are in active mode and the remaining cluster members are in sleep mode.

In this section, we first discuss the static and event driven network architecture used for tracking. The advantage of having static clustered network is that the clustering is required to be performed only once during formation of network. In such architecture, the CHs are fixed and all the cluster members know their corresponding CHs. These CHs are responsible for communication and computation.

Figure 1 shows the communication in static cluster. The boundary node BN1 locates the target with the help of other two boundary nodes that have detected the target. The trilateration is performed to compute its current location.

Let A, B, C, and T denote sensor nodes and target, respectively. Distance of target from nodes A, B, and C is d_1 , d_2 , and d_3 , respectively, and is obtained using the following equations. The localization is performed by trilateration, and the equations used are as described in the following:

$$\begin{aligned} d_1^2 &= (x_1 - x_t)^2 + (y_1 - y_t)^2, \\ d_2^2 &= (x_2 - x_t)^2 + (y_2 - y_t)^2, \\ d_3^2 &= (x_3 - x_t)^2 + (y_3 - y_t)^2. \end{aligned} \quad (1)$$

Now, the CH of BN1 computes target's future location using KF/EKF/PF filters and finds which cluster it may enter. It sends wake up message to respective CH through nodes BN1 and B. The CH finds the nearest node to target and wakes it for tracking. Thus, to start target tracking the amount of communication performed inside cluster is (5 transmission + 5 reception), from BN to node B, next node B to CH, and CH to nodes D, C, and F. In next step, this CH predicts the future location of the target and if it is inside its own cluster then it wakes up nearest node to that location. Thus, only CH performs communication and computation; hence there is more possibility of failing this node. Failure of CH prohibits participation of its cluster members in tracking. To deal with this problem, we adopted dynamic clustering in [33]. The dynamic clustering was performed using LEACH protocol [34], where the role of CH is performed by each node in rotation to increase the lifetime of the network. However, in this kind of architecture the communication required may be more than static clustered network. The location of the CH is not fixed and hence more numbers of communications are required to be performed to intimate the CH and initiate tracking. A more flexible network design can be achieved with dynamic clustering. However, its overhead in formation of clusters is considered to be a serious drawback [35]. Thus, for many practical situations, event driven clustering is an energy efficient solution [30, 31]. Hence, in this paper, we use an event driven clustering, where clusters are not formed a priori, but on the occurrence of an event nearby nodes form cluster to observe the event, that is, to track the target.

3.1. Event Driven Clustering. Figure 2 illustrates tracking inside the network by event driven clustered network. Here,

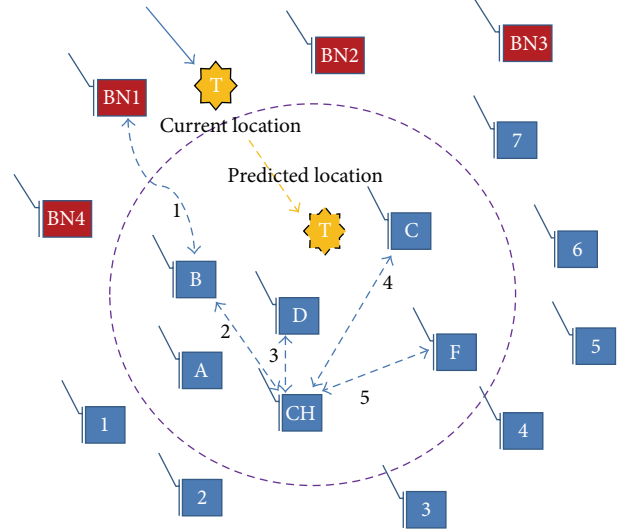


FIGURE 1: Static clustered network.

we assume that every node including boundary node has the distance and location information of all the nodes in the network. Thus, every node has that information which a CH has in static clustering. That means here every node can perform all the activities of CH. Similarly, we also assume that BS knows the locations of all the nodes in the network. BS identifies the network and hole boundary nodes by applying boundary detection algorithm such as [36]. The identity of hole boundary is transmitted to corresponding nodes, since they have to be awake for intrusion detection. Keeping nodes awake all the time leads to node running out of battery; hence these network boundary nodes are divided into two groups. For example, if there are n boundary nodes with IDs 1 to n assuming they are placed sequentially, all the odd numbered nodes go to the first group and even to the second group. When the first group is awake, second is in sleep mode and vice versa. These groups go to active mode alternately, so that energy is balanced during target's intrusion.

Only nodes with energy greater than threshold (τ), also called as live nodes, can participate in the network activities. Each node maintains an alive nodes list. There is no communication needed to know energy of a node; as a result, the energy consumption expended in communication can be saved conspicuously. As soon as the target enters into the network, the BN1 localizes the target. Next, it predicts the future location of the target with the help of filters such as Kalman, extended Kalman, or particle.

Now, BN1 finds the nearest node to the predicted location as B. The node B becomes the CH and sends wake up message to nodes C and D. As these nodes have energy greater than threshold (τ), acknowledgements are sent to the node B. On receipt of acknowledgement from this node the tracking begins. Thus only three duplex communications (3 transmissions + 3 receptions) are required to be performed to initiate tracking. On obtaining the current position of the target, the node B predicts its next location. If the nearest node to the predicted location is in range then node B

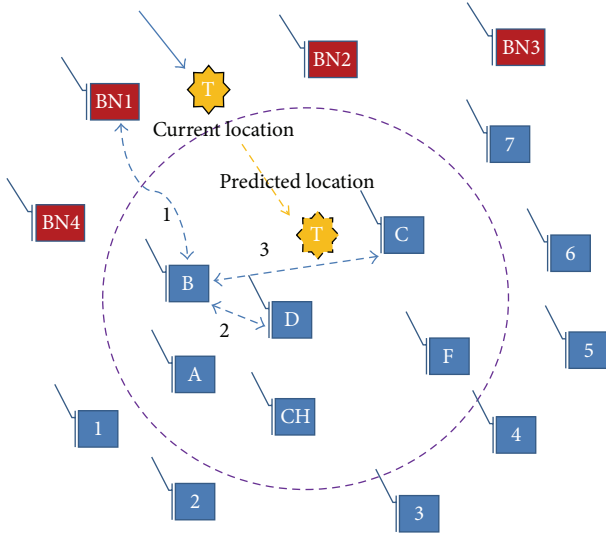


FIGURE 2: Tracking with event driven clustered network.

continues to be CH; otherwise, it handovers the role of CH to the next nearest node. This procedure is repeated as long as the target is not located or the energy of all nodes is not exhausted. This ensures the saving of energy by performing less communication. In the next subsection the energy model used in lost target recovery is presented.

3.2. Energy Analysis. For the energy analysis, we adopt the energy model presented by Shnayder et al. [37]. The MICA 2 platform has been used in this model, which uses ATMEL 128 processor and radio unit CC1000. The Mica2 mote is powered by a 3 V source. The nominal charge capacity (C) of these batteries is about 1300–3000 mAh. Considering $C = 2500$ mAh and two AA batteries, the energy per mote can be computed as follows:

$$\text{Energy} = 2500 \cdot 10^{-3} * 3600 * 3 = 27000 \text{ Joules.} \quad (2)$$

The following equations show the energy expended for communication:

$$\text{Energy} = (\text{Current} * \text{Voltage} * \text{Time}) \text{ Joules,} \quad (3)$$

where current is in Amperes, voltage is in Volts, and time is in seconds. For the energy computation of transmission or reception of message, the CPU is considered to be in active state [37]. The energy cost of transmission and reception can be evaluated using the following equations:

$$\begin{aligned} &\text{Transmission-Energy} \\ &= (I_{\text{CPU}} + I_{\text{Tx}} * \text{Voltage} * \text{Time}), \end{aligned} \quad (4)$$

$$\begin{aligned} &\text{Reception-Energy} \\ &= (I_{\text{CPU}} + I_{\text{Rx}} * \text{Voltage} * \text{Time}), \end{aligned}$$

where I_{CPU} is the current drawn by CPU; I_{Tx} and I_{Rx} are the current drawn by transmitter and receiver. As already

mentioned, we have formed two types of WSN depending on network's hole awareness. If the network knows hole(s) exists, then it is called as WSHAN; otherwise it is a WSHUN. In WSHAN, the hole boundary nodes location is available with BS. These HBNs are awakened by the currently tracking node as soon as the target comes in their vicinity. A WSHUN is that network in which hole boundary nodes do not participate in tracking in the specific way as mentioned above.

In our earlier work [12], we have proposed a lost target recovery mechanism using Kalman filter for static clustered network. The Kalman filter assumes Gaussian distribution for the state vector with Gaussian noise. However, its performance degrades with nonlinear dynamics of the target. Hence, we evaluate and compare the performance of KF based recovery algorithm [12] with the other two types of nonlinear filters, that is, EKF and PF. These filters are selected because the EKF assumes nonlinear state vector with Gaussian noise, and PF assumes nonlinear state vector with non-Gaussian noise. In addition, we also propose a recovery mechanism in hole aware network (WSHAN) using these linear and nonlinear filters and compare their performance in terms of recovery time and number of nodes waken during recovery and energy. The following subsection presents the system model used for tracking using KF, EKF, and PF.

3.3. System Model of Tracking. The KF, EKF, and PF follow two steps for tracking, updating, and prediction. A constant velocity model is used for tracking. The transition and measurement system equations are described in the following section.

3.3.1. Kalman Filter. For each time step (t) the state model is expressed as

$$x_t = F_t x_{t-1} + B_t u_t + w_t, \quad (5)$$

where F_t is the state transition model that is applied to the previous state x_{t-1} ; B_t is the control-input model that is applied to the control vector u_t ; w_t is the process noise which is assumed to be drawn from a multivariate normal distribution having zero mean with covariance Q_t . The F matrix can be expressed in terms of time step T and velocity V_x, V_y in (x, y) direction as depicted in

$$F = \begin{bmatrix} x + T * V_x \\ V_x \\ y + T * V_y \\ V_y \end{bmatrix}. \quad (6)$$

At time t , an observation or measurement z_t of the true state x_t is made according to

$$z_t = H_t x_t + v_t, \quad (7)$$

where H_t is the observation model which is used to map the true state space into the observed space and v_t is the observation noise which is assumed to be a Gaussian white noise having zero mean with covariance R_t :

$$v_t = N(0, R_t). \quad (8)$$

The measurement matrix consists of trilateration equations as described in

$$z_t = \begin{bmatrix} \sqrt{(x_1 - x_t)^2 + (y_1 - y_t)^2} \\ \sqrt{(x_2 - x_t)^2 + (y_2 - y_t)^2} \\ \sqrt{(x_3 - x_t)^2 + (y_3 - y_t)^2} \end{bmatrix}. \quad (9)$$

(1) *Prediction Step*. For the prediction phase, the state estimate from the previous step is used for producing an estimate of the current state as depicted by the following equations:

$$\begin{aligned} \hat{x}_{t|t-1} &= F_t \hat{x}_{t-1|t-1} + B_t u_t + w_t, \\ P_{t|t-1} &= F_t P_{t-1|t-1} F_t^T + Q_t, \end{aligned} \quad (10)$$

where $\hat{x}_{t|t-1}$ is the predicted or a priori state estimate and $P_{t|t-1}$ is the predicted or a priori covariance estimate.

(2) *Update Step*. In the update phase, state estimate is refined using the current state estimate and observation. Such an estimate is known as the a posteriori state estimate. The equations used in update step are as given in (11). To compute these equations the measurement residual \tilde{y}_t and residual covariance S_t are required. The innovation or measurement residual is $\tilde{y}_t = z_t - H_t(\hat{x}_{t|t-1})$ and the innovation or residual covariance is $S_t = H_t(P_{t|t-1})H_t^T + R_t$. The optimal Kalman gain K_t , the a posteriori state estimate $\hat{x}_{t|t}$, and the a posteriori estimate covariance $P_{t|t}$ are given in the following equations:

$$\begin{aligned} K_t &= P_{t|t-1} H_t^T S_t^{-1}, \\ \hat{x}_{t|t} &= \hat{x}_{t|t-1} + K_t \tilde{y}_t, \\ P_{t|t} &= (I - K_t H_t) P_{t|t-1}. \end{aligned} \quad (11)$$

The update step incorporates the observation and predicts the next location. As KF works only for linear dynamics of the target, to handle the nonlinearity in the trajectory, the EKF is used.

3.3.2. *Extended Kalman Filter*. In the EKF, the state transition and observation models are differentiable functions:

$$\begin{aligned} x_t &= f(x_{t-1}, u_{t-1}) + w_{t-1}, \\ z_t &= h(x_t) + v_t, \end{aligned} \quad (12)$$

where w_t and v_t are the process and observation noises which are assumed to be Gaussian noises with covariance Q_t covariance and R_t , respectively. For linearization, in EKF the Jacobian of process model and measurement model is computed and the states are updated.

The function f is used to compute the predicted state using the previous estimate. The function h is used to compute the predicted measurement using the predicted state. The matrix of partial derivatives (the Jacobian) is computed for f and h before applying the covariance. At each step, the Jacobian is evaluated with the help of current predicted states. This process linearizes the nonlinear function around

the current estimate. The Jacobian of the matrix H is as shown in the following equation:

$$H_t = \begin{bmatrix} \frac{\partial r_1}{\partial x} & \frac{\partial r_1}{\partial \dot{x}} & \frac{\partial r_1}{\partial y} & \frac{\partial r_1}{\partial \dot{y}} \\ \frac{\partial r_2}{\partial x} & \frac{\partial r_2}{\partial \dot{x}} & \frac{\partial r_2}{\partial y} & \frac{\partial r_2}{\partial \dot{y}} \\ \frac{\partial r_3}{\partial x} & \frac{\partial r_3}{\partial \dot{x}} & \frac{\partial r_3}{\partial y} & \frac{\partial r_3}{\partial \dot{y}} \end{bmatrix}, \quad (13)$$

where r_1 , r_2 , and r_3 are the ranges of the three nearest nodes from the target.

3.3.3. *Particle Filter*. In PF, the process x_t and the observations z_t can be modelled in the form as follows: x_0, x_1, \dots is a first order Markov process such that $x_t | x_{t-1} \sim p_{x_t|x_{t-1}}(x | x_{t-1})$ having an initial distribution of $p(x_0)$. The observations z_0, z_1, \dots are independent, provided that x_0, x_1, \dots are known. Thus, each z_t only depends on x_t :

$$z_t | x_t \sim p_{z|x}(z | x_t). \quad (14)$$

The system equations can be written as

$$\begin{aligned} x_t &= g(x_{t-1}) + w_t, \\ z_t &= h(x_t) + v_t, \end{aligned} \quad (15)$$

where both w_t and v_t are process noise and measurement noise, respectively, that are mutually independent and identically distributed sequences with known probability density functions. The $g(\cdot)$ and $h(\cdot)$ are known functions. The proposed algorithm is described in the following section.

4. Proposed Recovery Algorithm

As mentioned earlier, all the geographic locations of node, network boundary, and hole boundary nodes are available with BS. Similarly, the information of hole boundary nodes is available with their single hop neighbour node, and each boundary node knows the complete list of other boundary node locations. To facilitate energy conservation all the nodes are in sleep mode by default, except network boundary nodes that are awake alternately. Every node is capable of performing all the computation needed for prediction, tracking, and localization on recovery. The activity of tracking is informed to BS of the network and the BS communicates with all nodes in the network by multihop communication.

The proposed algorithm is divided into the following phases.

4.1. *Target Tracking Phase*. Boundary nodes detect the target's intrusion. The tracking is initiated with the current location estimation.

4.1.1. *Location Estimation of Target*. If three boundary nodes can detect target simultaneously then the trilateration is initiated by any one of the boundary nodes. Otherwise, their single hop neighbours are sent alert to wake up and detect

the target to perform trilateration. Once target is located then CH of respective BN starts prediction of next location of target.

The location prediction is performed with Kalman, extended Kalman, or particle filter as explained in Section 3. This predicted location is further updated on receiving target's actual localized value.

4.1.2. Prediction of Location. KF/EKF/PF predicts the future location of target based on current and previous locations and updating is performed accordingly. The following subalgorithms are used for KF, EKF, and PF, respectively.

(1) Kalman Filter Algorithm

(1) Time update is as follows.

- (i) State update $\hat{x}_{t+1} = F_t \hat{x}_t + B_t u_t + w_t$.
- (ii) Error covariance $P_{t+1} = F P_t F^T + Q$.

(2) Measurement update is as follows.

- (i) Compute Kalman gain $K_t = P_t^{-1} H^T (H P_t^{-1} H^T + R)^{-1}$.
- (ii) Update the estimate via z_t :

$$\hat{x}_t = \hat{x}_t + K_t (z_t - H \hat{x}_t^{-1}). \quad (16)$$

(iii) Update the error covariance:

$$P_t = (I - K_t H) P_t^{-1}. \quad (17)$$

(2) Extended Kalman Filter Algorithm

- (1) Obtain the current state estimate \hat{x}_t and error covariance P_t .
- (2) Linearise the system dynamics $\hat{x}_{t+1} = F_t \hat{x}_t + w_t$ around \hat{x}_t :

$$F_t = \nabla f_{t|\hat{x}_t}. \quad (18)$$

- (3) Apply the prediction step of the Kalman filter to yield next state estimate and error covariance.
- (4) Linearise the observation dynamics $\hat{z}_t = h_t \hat{x}_t + v_t$ around $\hat{x}_{t+1|t}$:

$$H_{t+1} = \nabla h_{t|\hat{x}_{t+1|t}}. \quad (19)$$

- (5) Apply the update step of the Kalman filter to yield the state estimate and error covariance.

(3) Particle Filter Algorithm

- (1) Generate a set of random particles (prior) from given current estimate.
- (2) Prediction—obtain samples from the prior:

$$x_{t+1}(i) = f(x_t(i), w_t(i)). \quad (20)$$

- (3) On receipt of the measurement z_t , evaluate the likelihood of each prior sample and obtain the normalized weight for each sample:

$$q_i = \frac{P(z_t | x_t(i))}{\sum_{j=0}^n P(z_t | x_t(j))}. \quad (21)$$

- (4) Resample the pdf.

- (5) Obtain the current estimate as the mean of the particles.

The current tracking node (CTN) performs the following steps:

- (1) predicting the next location of target,
- (2) finding member nodes in the vicinity of predicted location,
- (3) waking up nearest nodes to target's predicted location.

4.2. Recovery Phase. Recovery takes place in three steps, declaration of missing target, relocation, and hibernation or switching to the sleep mode.

4.2.1. Declaration of Missing Target. The declaration of loss of target is done as follows:

- (1) when it enters into the hole of WSHAN,
- (2) when CTN does not receive any acknowledgement in stipulated time period from the next selected node. The nonreceipt of acknowledgement may be caused due to any of the reasons mentioned earlier.

4.2.2. Recovery of the Lost Target. Following algorithms have been devised for lost target recovery with hole aware and unaware network.

(1) *Recovery in WSHUN.* As the loss of target occurs, CTN initiates the recovery process described as shown in Algorithm 1.

(2) *Recovery in WSHAN.* The algorithm of recovery in WSHAN is described as shown in Algorithm 2.

4.2.3. Hibernation. All the active nodes involved in recovery go to sleep mode as soon as target recovery message is received except those that are currently tracing the target.

The performance of the proposed algorithm is discussed in next section. The performance metric used for estimating error in tracking is root mean square error (RMSE). Similarly, for recovery of lost target, the algorithms are compared in terms of number of nodes awakened with recovery time.

5. Performance Evaluation

The proposed algorithm has been simulated in Matlab. Various trajectories of the target have been used for simulation. The experimental set up includes 225 nodes randomly

```

(1) If (!found target) Then
(2)     search in past few locations if target is available
(3) Else
(4)     Initiate Recovery
(5)     Predict next locations as few locations ahead
(6)     Wake up all SHNs around this location
(7)     If (Found by neighbouring nodes)
(8)         If (energy >  $\tau$ )
(9)             Become the CTN and localize target with Trilateration
(10)        End if
(11)     Send target found message to previous tracking node
(12)     Else
(13)         Wake up all two hop nodes around this location
(14)         Repeat till target is found
(15)     End if

```

ALGORITHM 1: Lost target recovery in WSHUN algorithm.

```

(1) If (target is in range of HBN) Then
(2)     Wake up all HBNs
(3)     Track the target till in reach (As the target becomes unreachable it is lost)
(4)     If (in range) Then
(5)         Initiate Recovery
(6)         If (target is in vicinity of three HBNs) Then
(7)             If (energy >  $\tau$ )
(8)                 Become the CTN and Localize target with trilateration
(9)             End If
(10)        Send the target found message to previous tracking node
(11)        End If
(12)     End If
(13)     Else If (!found target)
(14)         Call (KF/EKF/PF algorithm)
(15)     End If

```

ALGORITHM 2: Lost target recovery in WSHAN algorithm.

deployed in the field of area $140\text{ m} \times 140\text{ m}$. The radio range of all these nodes is 15 m. Each node has average connectivity of seven. The alternate boundary nodes are active to detect the intrusion as mentioned earlier. As soon as the target enters into the field, the boundary nodes detect the presence. As already stated, with minimum three nodes the target is localized using trilateration. Figure 3 shows deployment of the nodes and the connectivity diagram of the network used in tracking.

Figure 3 shows sensor node with its ID using blue asterisk (*), and the border nodes are shown by circled asterisk (*). Green lines joining the nodes are radio link or connectivity. We have evaluated performance of the proposed algorithm for the following three cases where target may be lost:

- (1) node failure,
- (2) communication failure,
- (3) presence of coverage hole in the network.

All the cases are discussed and analysed in detail in the subsequent subsections.

5.1. Loss of Target due to Node Failure. The recovery algorithm has been analyzed with various trajectories, to simulate the loss of target due to hole created by failing nodes or group of failing nodes at different locations. Here we present two case scenarios for loss of target due to node failure.

Case I. In this scenario, a hole is synthetically generated by failing a group of nine nodes with IDs 163, 138, 144, 147, 148, 155, 156, 165, and 178 as shown in Figure 4(a). These node IDs are not shown in every figure as these will become unreadable in the figure having many different symbols. Therefore readers are referred to experimental setup shown in Figure 3 to see IDs. The failing nodes are shown by squared asterisk (*). Thus, the node density in the network is 216. The target is moving with a zigzag trajectory inside the network with a constant speed as shown in Figure 4(a). The red diamonds are points on trajectory, blue circles (+) are predictions of KF, black (•) is predictions of EKF, and magenta is predictions of PF. The target is lost at one of the turning points of the trajectory since a hole is created due to failure of

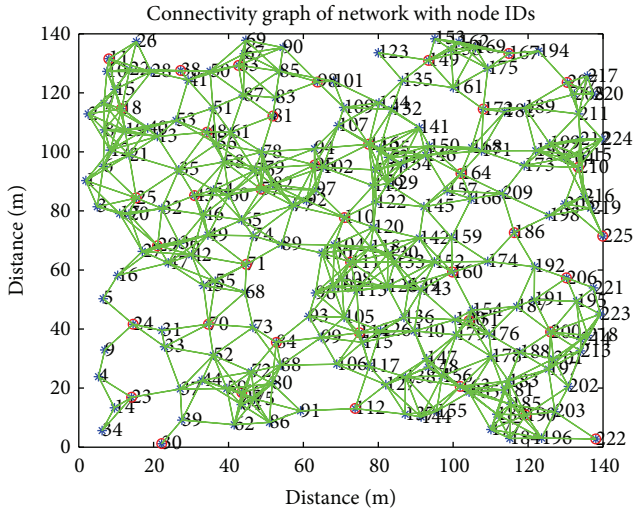


FIGURE 3: The network with connectivity 7% and 5% range error.

nodes in that region, and hollow blue circles are the positions of the target where it is undetected by the network (i.e., lost) as shown in the figure. As soon as the target exits the hole or comes in the vicinity of any three sensing nodes, it is detected and localized by them. Figure 4(b) shows the comparison graph for zigzag track.

As can be seen from this figure the PF based tracking performs best as compared to KF and EKF. The performance comparison of target location estimation using all the filters is summarized in Table 1. The total number of nodes required to wake up for recovery are the same for all the algorithms based on filter. However, the RMSE as well as the recovery time is the least in PF based algorithms. The RMS error is large in KF since the first prediction after the recovery of the target is erroneous as shown in Figure 4(b) and Table 1. The energy consumed in complete target tracking with recovery for static and event driven clustered architecture is computed according to (4).

Thus, there is approximately 34.1% saving in energy with event driven architecture as compared to static. The total time required for tracking with recovery using static clustering is 12.78 s and for event driven clustering is 11.2 s.

Case II. Another case scenario is simulated with a different trajectory and failing another group of nodes as shown in Figure 5(a). Figure 5(b) shows the comparison graph for this track. The target enters the network at location (0, 0) and leaves from (43, 140). Total strength of the nodes in the network is 225; however due to failure of 11 numbers of nodes, only 214 numbers of nodes are alive. The failing nodes are shown in square block, where hole is generated.

The performance comparison of the recovery algorithm with filters is summarised in Table 2. The energy consumption in event driven architecture is less as compared to static. There is approximately 34.2% saving in energy using event driven clustering. The total time required for static and event driven clustering is 13.99 s and 11.34 s, respectively.

TABLE 1: Performance comparison: node failure—Case I.

Filter	Node awaken	RMSE	Time (S)
Existing recovery (KF) [12]	24	5.08	7.08
Proposed recovery (EKF)	24	0.80	5.03
Proposed recovery (PF)	24	0.72	5.04

TABLE 2: Performance comparison: node failure—Case II.

Filter	Node awaken	RMSE	Time (S)
Existing recovery (KF) [12]	14	3.2	5.04
Proposed recovery (EKF)	14	0.85	5.01
Proposed recovery (PF)	14	0.81	5.03

TABLE 3: Performance comparison: communication failure—Case I (A) 1st recovery.

Filter	Node awaken	RMSE	Time (S)
Existing recovery (KF) [12]	23	3.9	5.22
Proposed recovery (EKF)	23	0.9	5.00
Proposed recovery (PF)	23	0.8	5.04

5.2. Loss of Target due to Communication Failure. In this scenario we simulate a situation where two holes exist, first due to uneven deployment of nodes and second due to failing nodes. Two cases are discussed here, considering different trajectories and holes at different locations.

Case I. The target's manoeuvring inside the network and its corresponding trajectory is shown in Figure 6(a). Figure 6(b) shows the comparison graph for this track. To synthetically generate the hole nine nodes are removed from the experimental network shown in Figure 3. There are total 217 nodes in the network, and one more hole is created by failing eight nodes. Thus, there are total 208 alive nodes in the network. The eight failing nodes are shown by squared asterisk (*). The first loss of target has occurred due to failing nodes. The recovery with KF is delayed, whereas the EKF and PF have quick and accurate recovery. The second loss of target is due to hole present in the network.

The performance comparison of the recovery algorithm is stated in Tables 3 and 4. The energy saving in this scenario is about 34.57% with the event driven architecture as compared to static clustered network. The tracking time with static and event driven clustering is 15.99 s and 13.78 s, respectively.

Case II. Figure 7(a) shows loss of target due to communication failure at two places.

Figure 7(b) shows the comparison graph for this track. The target enters the network at location (0, 0) and leaves at location (42, 140). There are two holes present in the network. The first hole is supposed to be generated at the time of deployment (10 nodes removed), and second hole is due to

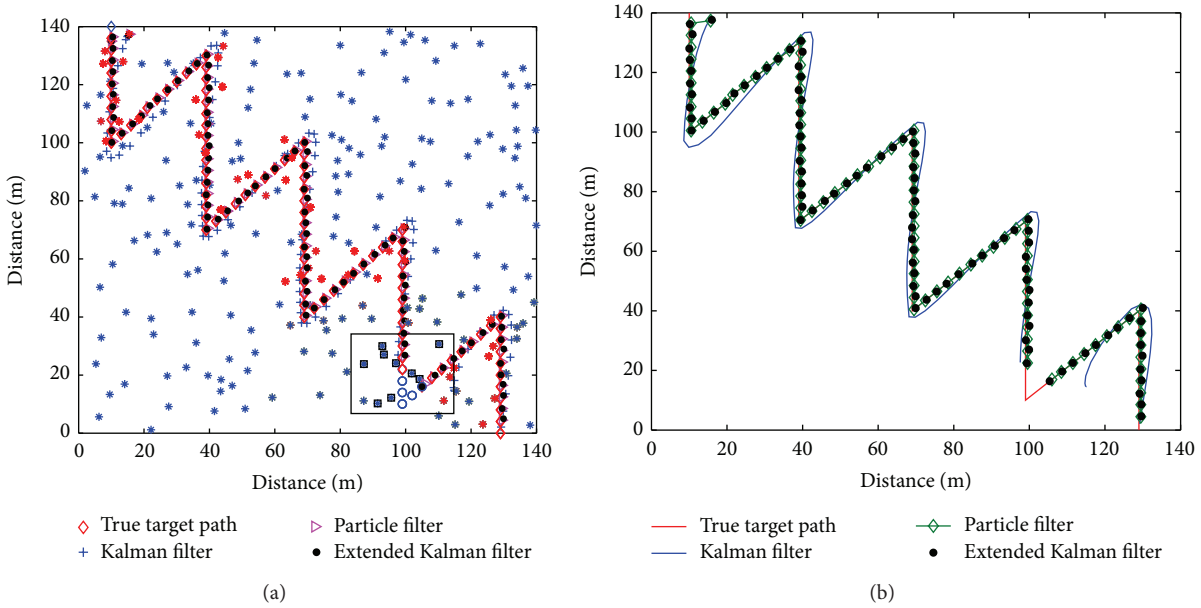


FIGURE 4: Loss of target due to failing nodes.

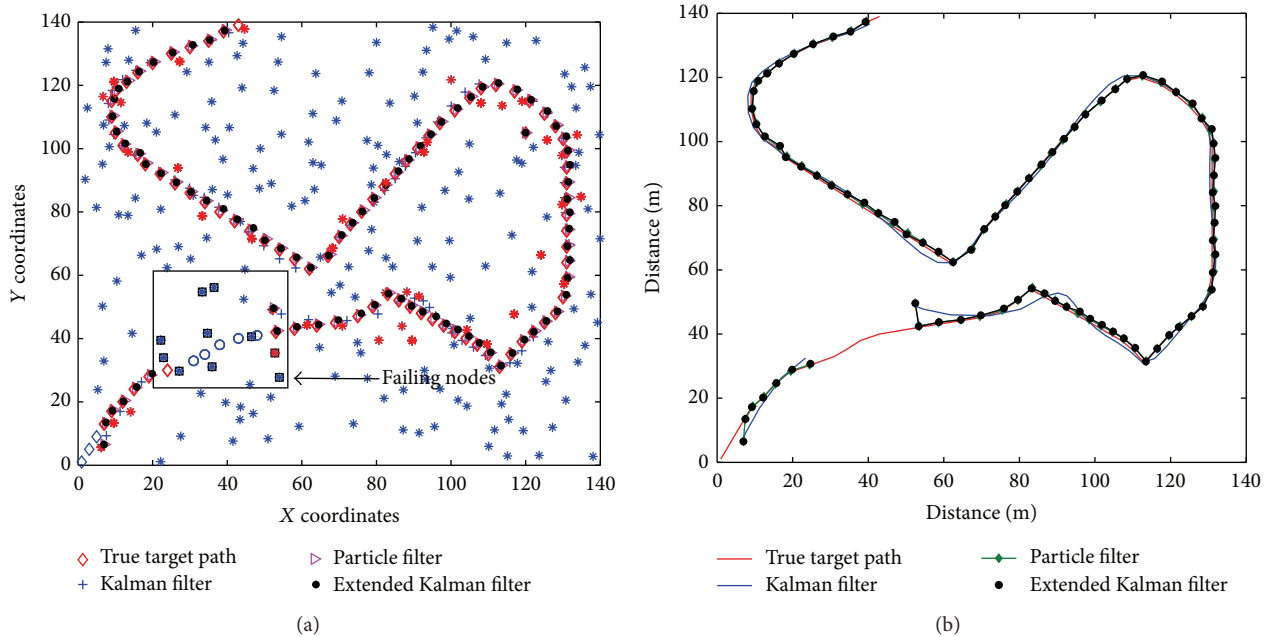


FIGURE 5: Loss of target due to failing nodes.

failure of ten nodes as shown in figure. The total alive nodes present inside the network are 205.

The performance comparison is shown in Tables 5 and 6. The detection and localization of lost target by KF is late and erroneous; hence a large error is obtained as compared to other counterparts. Also a significant difference can be seen in number of nodes required to wake up for recovery of the target as compared to others. The PF, EKF recover the target

at second level waking up 10 numbers of nodes with RMSE of 1.97% and 1.75%, respectively.

The second recovery of the lost target requires almost the same time, number of nodes waken, and RMSE as depicted in the table. Thus, the PF and EKF are better than KF and perform almost the same in terms of nodes required to wake up, time, and accuracy. The energy saving using event driven clustering in this case is about 34.21%. The tracking time with

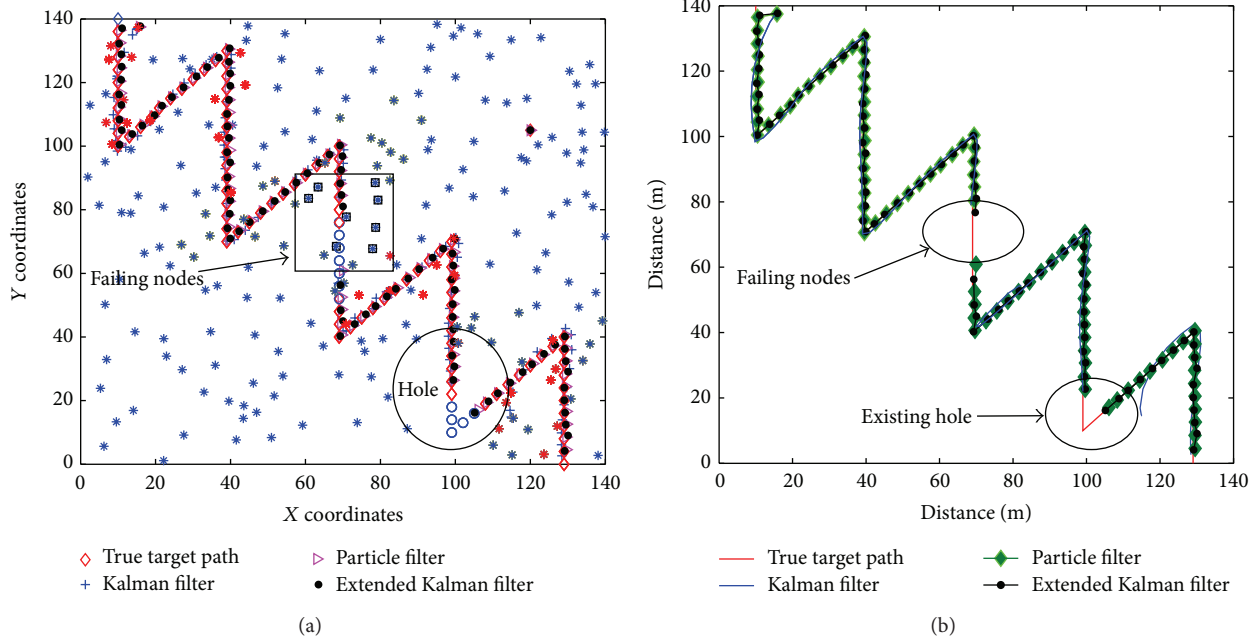


FIGURE 6: (a) Loss of target due to communication failure—Case I trajectory. (b) Loss of target due to communication failure—Case I graph.

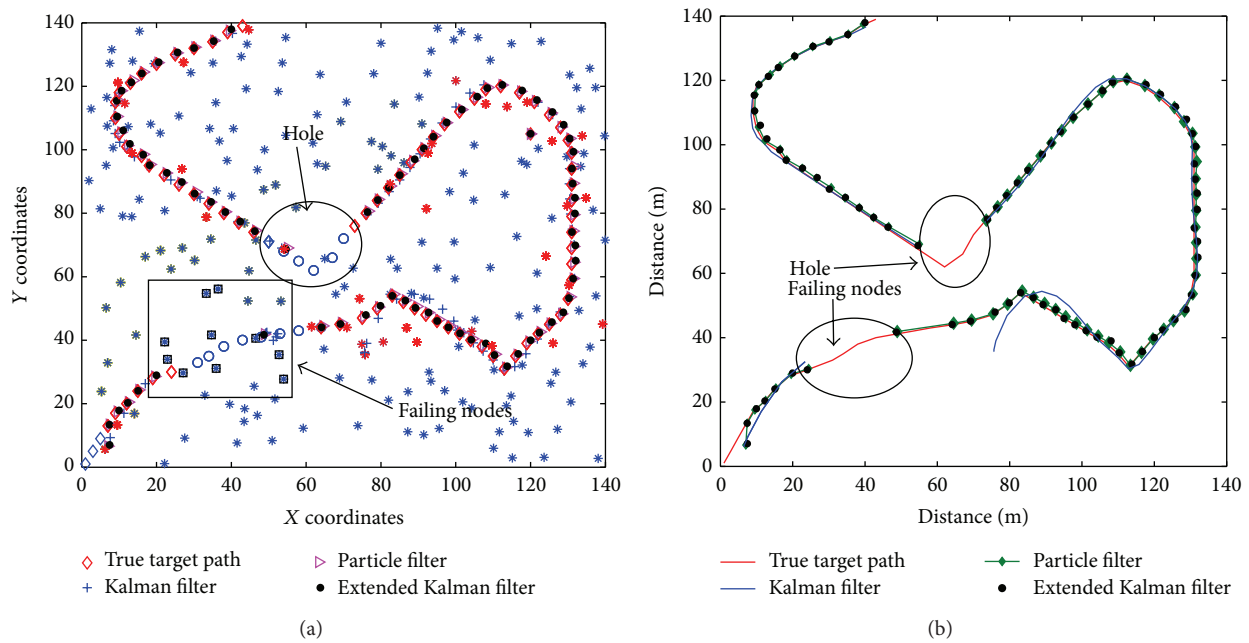


FIGURE 7: (a) Loss of target due to communication failure—Case II trajectory. (b) Loss of target due to communication failure—Case II graph.

static and event driven clustering is about 15.99 s and 14.21 s, respectively.

5.3. Loss of Target due to Coverage Hole in Network. A very interesting case is discussed in this subsection. The hole exists in the network due to uneven deployment of the nodes as shown in Figure 8(a) by thick blue lines. In this case, we have performed simulations with WSHAN and WSHUN.

The target enters into the network at location (140, 105) and exits at location (0, 35) as shown in Figure 8(a).

5.3.1. Tracking with WSHUN. In this scenario the target is lost twice as it enters the hole twice. First time when target enters into the hole it is tracked as long as it is in range of the tracking nodes. As soon as it becomes unreachable, it is declared as lost by the current cluster head. Now, the recovery mechanism is initiated, and the target is recovered

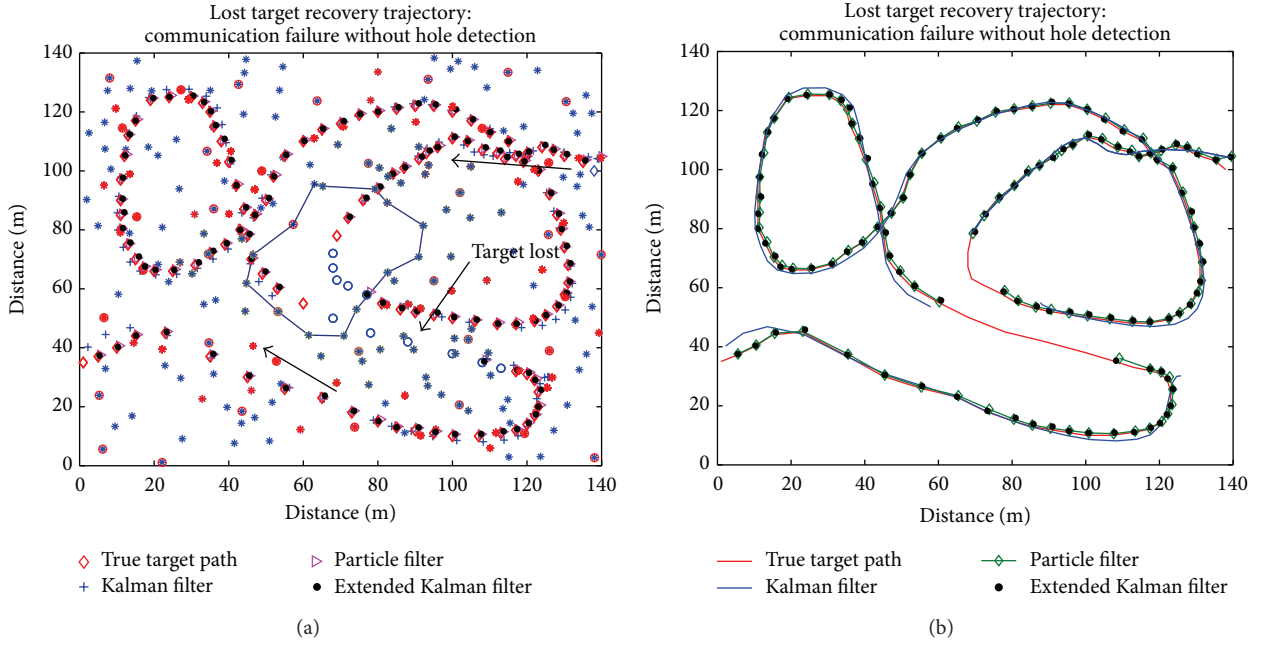


FIGURE 8: (a) Loss of target due to coverage hole in the WSHUN network trajectory. (b) Loss of target due to coverage hole in the WSHUN network graph.

TABLE 4: Performance comparison: communication failure—Case I (B) 2nd recovery.

Filter	Node awoken	RMSE	Time (S)
Existing recovery (KF) [12]	13	2.79	5.10
Proposed recovery (EKF)	13	0.87	5.02
Proposed recovery (PF)	13	0.75	5.04

TABLE 5: Performance comparison: communication failure—Case II (A) 1st recovery.

Filter	Node awoken	RMSE	Time (S)
Existing recovery (KF) [12]	19	3.9	5.22
Proposed recovery (EKF)	10	1.97	5.02
Proposed recovery (PF)	10	1.75	5.04

TABLE 6: Performance comparison: communication failure—Case II (B) 2nd recovery.

Filter	Node awoken	RMSE	Time (S)
Existing recovery (KF) [12]	6	2.1	5.02
Proposed recovery (EKF)	5	1.8	5.01
Proposed recovery (PF)	5	1.7	5.02

with second level recovery waking up 28 nodes. Irrespective of filters, same number of nodes is woken up for recovery. However, the recovery time and RMSE in KF are greater

TABLE 7: Performance comparison: coverage hole in WSHUN—(A) 1st recovery.

Filter	Node awoken	RMSE	Time (S)
Existing recovery (KF) [12]	28	2.09	5.12
Proposed recovery (EKF)	28	0.79	5.02
Proposed recovery (PF)	28	0.78	5.04

TABLE 8: Performance comparison: coverage hole in WSHUN—(B) 2nd recovery.

Filter	Node awoken	RMSE	Time (S)
Existing recovery (KF) [12]	29	2.990	5.83
Proposed recovery (EKF)	29	0.77	5.04
Proposed recovery (PF)	29	0.71	5.06

than the other two nonlinear filters. Thus, it is clear that if prediction is good less numbers of nodes are required to wake up for recovery, and delay in recovering the target is less. The detailed statistics of recovery are given in Tables 7 and 8. When the target heads towards the hole second time, again it is tracked inside till it is in range of tracking nodes and is lost later as shown in Figure 8(b).

Now the recovery mechanism is initiated; however the target is not recovered immediately outside the hole boundary because the target speed is slightly increased. The target is not found at the location predicted by recovery mechanism. So single hop neighbours are woken up nearer to the expected

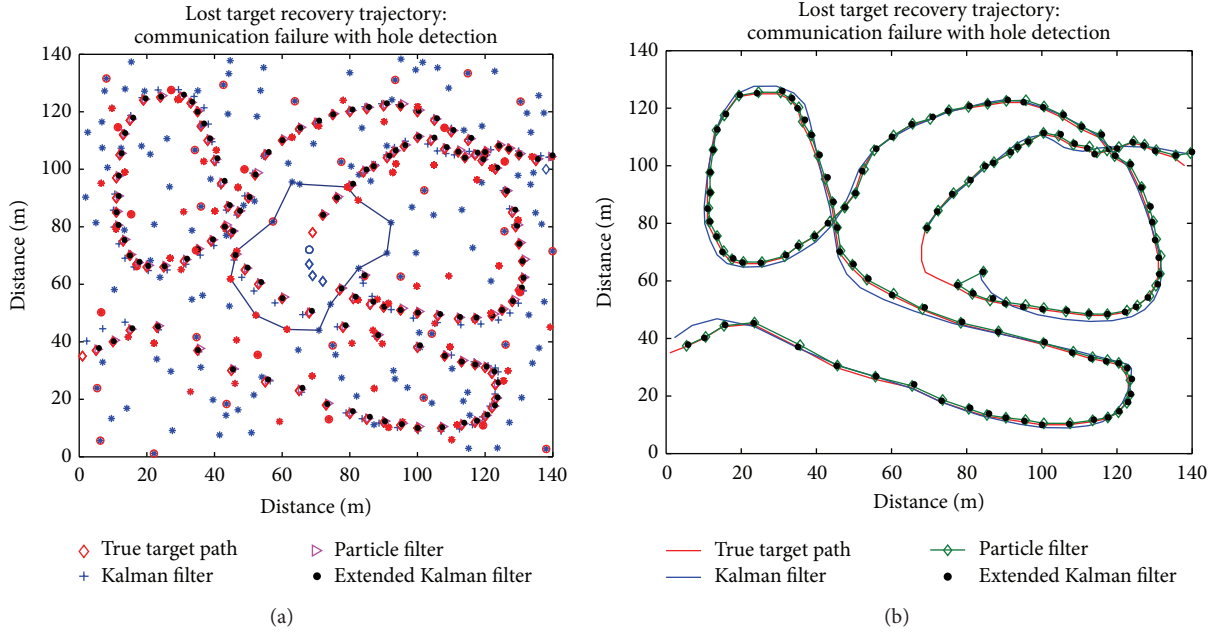


FIGURE 9: (a) Loss of target due to coverage hole in the WSHAN network trajectory. (b) Loss of target due to coverage hole in the WSHAN network graph.

location of the target. However, the target is missed since it has already passed that location. Later target is recovered by waking up two hop neighbours. Nearly for about 5 steps the target goes undetected even if it is outside a hole, which is shown by blue circle in the figure. The difference in tracking with all filters is shown in Tables 7 and 8 and Figure 8(b). The total time required for tracking is about 21 seconds with static and 18.19 seconds for event driven clustering. The energy saving in this scenario is about 35% using the event driven clustering.

5.3.2. Tracking with WSHAN. In WSHAN, the single hop tracking node wakes up boundary nodes as it enters the hole; therefore there are less chances of losing the target and less energy and time are spent in recovering it. When the target is heading towards the hole, for the first time it is tracked till it is in range and later it is lost as shown in Figure 9(a).

Now, the target is recovered by HBN near boundary. However, initial prediction of the target location is somewhat erroneous for all the filters, and it is diverted little away from the track as shown in figure. From the next location onwards it is tracked properly. When the target enters the hole second time, again all the hole boundary nodes are alerted and tracking is performed. As can be seen from Figure 9(b), unlike the case of WSHUN, the target is completely tracked throughout the coverage hole without losing it.

The advantage of having WSHAN is that there are less chances of losing of target, and it is recovered early if lost. The noticeable advantage is that very little number of nodes is required for recovery; hence less network energy is spent. In this case only 14 nodes are required to be woken up which is boundary nodes. Thus including both entries inside the hole only (14 + 14) nodes are required in WSHAN and (28 + 29)

TABLE 9: Performance comparison: coverage hole in WSHAN.

Filter	Node awoken	RMSE	Time (S)
Existing recovery (KF) [12]	14	1.97	5.12
Proposed recovery (EKF)	14	0.84	5.02
Proposed recovery (PF)	14	0.96	5.02

nodes are required for WSHUN. The tracking and recovery performance can be seen in Table 9 and Figure 9(b). The energy saving is about 37% with the event driven architecture. Thus, the energy saving is due to hole aware network as well as event driven clustered network architecture.

6. Conclusion

The very daunting task of keeping the tracking protocol energy efficient is due to the constraint on resources. Most of the existing work concentrates on optimizing communication cost. However, timely waking up of nodes during tracking with correct prediction of location is also important; thus if a right location is predicted at right time tracking error will be reduced. This plays a very important role in recovery of lost target. In this paper, we have evaluated different schemes of the recovery of lost target using linear and nonlinear filters. Various reasons of losing objects are also discussed and recovery mechanisms are devised. It is proved that the recovery can be done faster with less energy, if the hole boundary is already known and boundary nodes are alerted as soon as target enters into the hole area. The energy analysis of each case scenario shows that the event driven clustering is more energy efficient than static clustering.

In addition, the hole aware network saves energy by waking up optimum nodes for recovery. Thus, from the performance analysis it is observed that prediction accuracy and recovery of PF based algorithm are the best, making it an energy efficient tracking protocol. From the simulation results it is evident that the proposed recovery mechanism outperforms the existing algorithm for efficient recovery of the target.

Conflict of Interests

The authors declare that they have no competing interests.

References

- [1] X. Yang, K. G. Ong, W. R. Dreschel, K. Zeng, C. S. Mungle, and C. A. Grimes, "Design of a wireless sensor network for long-term, in-situ monitoring of an aqueous environment," *Sensors*, vol. 2, no. 11, pp. 455–472, 2002.
- [2] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, New York, NY, USA, September 2002.
- [3] K. Martinez, J. K. Hart, and R. Ong, "Environmental sensor networks," *Computer*, vol. 37, no. 8, pp. 50–56, 2004.
- [4] M. Bal, H. Xu, W. Shen, and H. Ghenniwa, "A 3-D indoor location tracking and visualization system based on wireless sensor networks," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 1584–1590, October 2010.
- [5] Y. X. Li, H. S. Shi, and S. P. Zhang, "An optimized scheme for battlefield target tracking in wireless sensor network," in *Proceedings of the 2nd International Conference on Industrial and Information Systems (IIS '10)*, vol. 2, pp. 428–431, 2010.
- [6] M. Di, E. M. Joo, and L. H. Beng, "A comprehensive study of kalman filter and extended kalman filter for target tracking in wireless sensor networks," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '08)*, pp. 2792–2797, Singapore, October 2008.
- [7] M. Vemula, J. Míguez, and A. Artés-Rodríguez, "A sequential Monte Carlo method for target tracking in an asynchronous wireless sensor network," in *Proceedings of the 4th Workshop on Positioning, Navigation and Communication (WPNC '07)*, pp. 49–54, March 2007.
- [8] Q. Fang, J. Gao, and L. J. Guibas, "Locating and bypassing routing holes in sensor networks," in *Proceedings IEEE 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, 2004.
- [9] Z. Chi, Z. Yanchao, and F. Yuguang, "Defending against physical destruction attacks on wireless sensor networks," in *Proceedings of the IEEE Conference on Military Communications*, pp. 1061–1067, Piscataway, NJ, USA, 2006.
- [10] A. Khare and K. M. Sivalingam, "On recovery of lost targets in a cluster-based wireless sensor network," in *Proceedings of the 9th Annual IEEE International Conference on Pervasive Computing and Communications*, pp. 208–213, Seattle, Wash, USA, March 2011.
- [11] S. Jessica, B. Dirk, and D. Glenn, "Efficient tracing of failed nodes in sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 122–130, New York, NY, USA, September 2002.
- [12] A. Gupta, S. Patil, and M. Zaveri, "Lost target recovery in wireless sensor network using tracking," in *Proceedings of the International Conference on Communication Systems and Network Technologies (CSNT '12)*, pp. 352–356, May 2012.
- [13] Y. Haiming and B. Sikdar, "A protocol for tracking mobile targets using sensor networks," in *Proceedings of the IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 71–81, 2003.
- [14] Y. Xu, J. Winter, and W.-C. Lee, "Prediction-based strategies for energy saving in object tracking sensor networks," in *Proceedings of the IEEE International Conference on Mobile Data Management (MDM '04)*, pp. 346–357, January 2004.
- [15] Y. Hai-bo, Q. Ning, and C. You-rong, "An object tracking technique in wireless sensor network based on prediction," in *Proceedings of the International Conference on Communication Software and Networks*, pp. 3–8, February 2009.
- [16] S. Ren, Q. Li, H. Wang, and X. Zhang, "Design and analysis of wave sensing scheduling protocols for object-tracking applications," in *Proceedings of the 1st IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '05)*, pp. 228–243, Berlin, Germany, July 2005.
- [17] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, pp. 129–143, New York, NY, USA, October 2004.
- [18] Z. Yang and Y. Liu, "Quality of Trilateration: confidence based iterative localization," in *Proceedings of the 28th International Conference on Distributed Computing Systems (ICDCS '08)*, pp. 446–453, IEEE, Beijing, China, July 2008.
- [19] S. Kamath, E. Meisner, and V. Isler, "Triangulation based multi target tracking with mobile sensor networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 3283–3288, April 2007.
- [20] S. Patil and M. Zaveri, "Target tracking approaches in wireless sensor network," in *Proceedings of the International Conference on Computer Communications and Networks (CCN '10)*, pp. 130–137, Orlando, Fla, USA, 2010.
- [21] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.
- [22] M. Bal, H. Xu, W. Shen, and H. Ghenniwa, "A 3-D indoor location tracking and visualization system based on wireless sensor networks," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '10)*, pp. 1584–1590, October 2010.
- [23] F. Deldar and M. H. Yaghmaee, "Designing an energy efficient prediction-based algorithm for target tracking in wireless sensor networks," in *Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP '11)*, pp. 1–6, IEEE, November 2011.
- [24] H. Simon, *Adaptive Filter Theory*, Prentice Hall, Upper Saddle River, NJ, USA, 3rd edition, 1996.
- [25] R. van der Merwe, A. Doucet, N. de Freitas, and E. A. Wan, "The unscented particle filter," in *Proceedings of the 15th Conference on Neural Information Processing Systems (NIPS '00)*, pp. 584–590, 2000.
- [26] P. Muraca, P. Pugliese, and G. Rocca, "Extended Kalman filtering using wireless sensor networks," in *Proceedings of the 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA '08)*, pp. 1084–1087, September 2008.

- [27] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.
- [28] Z.-J. Yu, S.-L. Dong, J.-M. Wei, T. Xing, and H.-T. Liu, "Neural network aided unscented kalman filter for maneuvering target tracking in distributed acoustic sensor networks," in *Proceedings of the International Conference on Computing: Theory and Applications (ICCTA '07)*, pp. 245–249, IEEE Computer Society, Kolkata, India, March 2007.
- [29] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: a survey," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 103–123, 2002.
- [30] Z.-W. Zheng, Z.-H. Wu, and H.-Z. Lin, "An event-driven clustering routing algorithm for wireless sensor networks," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*, vol. 2, pp. 1802–1806, Sendai, Japan, October 2004.
- [31] M. Noori and M. Ardakani, "Lifetime analysis of random event-driven clustered wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 10, pp. 1448–1458, 2011.
- [32] P. Kumarawadu, D. J. Dechene, M. Luccini, and A. Sauer, "Algorithms for node clustering in wireless sensor networks: a survey," in *Proceedings of the 4th International Conference on Information and Automation for Sustainability (ICIAFS '08)*, pp. 295–300, Colombo, Sri Lanka, December 2008.
- [33] S. Patil, A. Gupta, and M. Zaveri, "Efficient target recovery in wireless sensor network," *Advances in Intelligent Systems and Computing*, vol. 176, no. 1, pp. 385–394, 2012.
- [34] H. W. Rabiner, C. Anantha, and B. Hari, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences*, vol. 8, pp. 8020–8030, IEEE Computer Society, Washington, DC, USA, 2000.
- [35] O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks: recent developments and deployment challenges," *IEEE Network*, vol. 20, no. 3, pp. 20–25, 2006.
- [36] F. Yan, P. Martins, and L. Decreusefond, "Connectivity-based distributed coverage hole detection in wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '11)*, vol. 16, pp. 1–6, 2011.
- [37] V. Shnayder, M. Hempstead, B.-R. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 188–200, Baltimore, Md, USA, November 2004.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

