

## A Distributed Coverage and Target Tracking Algorithm for Wireless Sensor Networks

<sup>1</sup>Senthamilselvi, M. and <sup>2</sup>N. Devarajan

<sup>1</sup>Department of IT, Sri Ramakrishna Engineering College, Coimbatore, India

<sup>2</sup>Department of EEE, Government College of Technology Coimbatore, India

---

**Abstract: Problem statement:** Due to communication, processing and energy constraints, tracking in sensor networks creates different challenges. Since existing algorithms utilized the information from all the sensors for tracking, it results in higher expenditure of energy and reduced lifetime for the network. **Approach:** We propose to develop a distributed coverage and target tracking algorithm for wireless sensor networks. Tracking issue is first addressed through the determination of a reduced cover for the region of interest. For reduced coverage, we have used the distributed connected coverage algorithm. We propose a target zone detection algorithm to estimate the locations of the targets from the sensors within the target zone. The locations of the sensors are estimated by the localization algorithm. **Results:** This is performed with the unknown location information by utilizing the knowledge of the positions of the sensors and inter-sensor measurements. **Conclusion:** By simulation results, we show that the proposed algorithm achieves reduced energy consumption and coverage along with delay.

**Key words:** Wireless sensor networks, distributed coverage, target tracking algorithm, Active Path Search (APS), Active Path Response (APR), target zone, detection algorithm

---

### INTRODUCTION

**Wireless sensor network:** The use of small, inexpensive, low-power, distributed devices, which have the potential of local processing and wireless communication, have been made a reality owing to the recent technological improvements. These kinds of nodes are known as sensor nodes. The ability of each sensor node is restricted in terms of limited amount of processing. However the sensor nodes have the capability to gauge a given physical environment in great detail in cases where the sensor nodes are coordinated with the information from a large number of other nodes. Therefore, a collection of sensor nodes coordinating to perform some specific action is known as a sensor network. The sensor networks are in contrast to the traditional networks, due to fact that sensor networks depend on dense deployment and coordination to carry out their tasks (Estrin *et al.*, 2001).

Sensor networks have been proposed for a wide variety of application areas, including industrial, military, biomedical and environmental areas. Some examples of sensor network applications are as follows (Estrin *et al.*, 1999).

### MATERIALS AND METHODS

**Intrusion detection and tracking:** In order to detect, classify and track intruding personnel and vehicles, sensors are arranged along the border of a battlefield.

**Environmental monitoring:** In order to give out early warning of forest fires, specialized sensor nodes which have the capability to detect temperature changes and/or smoke are arranged in high-risk areas of a forest.

**Indoor surveillance:** In order to provide security in an art gallery, shopping mall, or other facilities surveillance sensor networks can be used.

**Traffic analysis:** In order to monitor vehicle traffic on a highway or a congested part of a city, traffic sensor networks can be used.

**Tracking in sensor networks:** Identifying an object by its particular sensor signature and detecting its path over a period of time are included in tracking. This is one of the applications which can be attained by developing the characteristics of wireless networks. By using the intrinsic parallelism of the distributed sensors, multiple objects can be tracked simultaneously. Even though, based on tracking in remote or unreachable locations, the relatively low cost and simplicity of deployment enables the use of sensor network, it is essential to be deployed within a period of time. In addition, target tracking has been broadly studied for sensor networks with the assistance of large nodes and distributed tracking algorithm. Tracking in ad hoc networks with micro sensors are complex due to

---

**Corresponding Author:** Senthamilselvi, M., Department of IT, Sri Ramakrishna Engineering College, Coimbatore, India

communication, processing and energy constraints. The sensors should work together and share the data to utilize the benefits of sensor data fusion. But without sending the data requests and receiving the data from all sensors, the fusion should be performed which overloads the networks and uses the energy supply (Watfa and Commuri, 2006).

**Sensor coverage and maintenance:** Two important properties of a sensor network play critical roles in the design approach. They are coverage and connectivity. Coverage describes how well sensors in the network can monitor a geographical region in question. Connectivity simply describes the connectivity properties of the underlying network topology. It is often desirable that the network is connected (Li *et al.*, 2002; Liu *et al.*, 2009).

The primary issue in the sensor applications is the quality of monitoring which is offered by the networks. This quality is generally measured based on the coverage of a target area by the deployed sensors. Coverage is the monitoring of each point in the target area by at least one sensor, inside the sensing range which is called as 1-coverage. Each point should be within the sensing range of  $k$  or more sensors, in  $k$ -coverage ( $k \geq 1$ ) problem. Covering each point by multiple sensors is required for many applications because it provides redundancy and fault tolerance. Moreover  $k$ -coverage is required for the proper working of many applications.

The probable solution to activate from a previously deployed set of sensors is choosing the least set of sensors such that all locations are  $k$ -covered. But, obtaining a least set of sensors is crucial because of its reduction in interface among the active sensors and total energy consumption. Yet, the problem of selecting the smallest amount of sensors is NP-hard.

**Localization in sensor networks:** In wireless sensor networks, the highly desirable characteristics are the ability of self localization. The measurement data in environmental monitoring applications such as bush fire surveillance, water quality monitoring and precision agriculture are meaningless without knowing the location from where the data are attained. In addition, several applications such as inventory management, intrusion detection, road traffic monitoring, health monitoring, reconnaissance and surveillance are enabled by the location estimation (Mao *et al.*, 2007).

The locations of the sensors are estimated by the sensor network localization algorithm. This is performed with the unknown location information by utilizing the knowledge of the positions of the sensors and inter-sensor measurements such as distance and

bearing measurements. The sensors with the known location information are called as anchors and by using a Global Positioning System (GPS) or by installing anchors at points with known coordinates, their locations can be obtained. These anchors will detect the location of the sensor network in the global coordinate system where the applications requiring a global coordinate system (Mao *et al.*, 2007).

These anchors define the local coordinate system to which all other sensors are referred in the applications where a local coordinate system is sufficient. Most of the sensors do not know their locations because of the constraints on the cost and size of sensors, energy consumption, implementation environment and the deployment of the sensors. Sensors with unknown location information are called as non-anchor nodes. By using the sensor network localization algorithm, their coordinates can be estimated (Mao *et al.*, 2007).

In our previous study, we have developed an energy efficient tracking algorithm for reduced coverage in wireless sensor networks. In this study, a border cover node is included and it should send its coordinates which results in increased overhead and it is more expensive. Therefore, in this study we use a self localization technique to determine the co-ordinates of the sensor nodes.

**Related work:** Zhang and Hou (2005) have addressed the issues of maintaining sensing coverage and connectivity by keeping a minimum number of sensor nodes in the active mode in wireless sensor networks. They derived, under the ideal case in which node density is sufficiently high, a set of optimality conditions under which a subset of working sensor nodes can be chosen for complete coverage. Based on the optimality conditions, they have devised a decentralized and localized density control algorithm, OGDC. OGDC is fully localized and can maintain coverage as well as connectivity, regardless of the relationship between the radio range and the sensing range.

Wang *et al.* (2008) have studied the target tracking with wireless binary sensor networks, in which each sensor can return only 1-bit information regarding target's presence or absence in its sensing range. They have proposed a novel, real-time and distributed target tracking algorithm for an imperfect binary sensing model. They have observed that their algorithm yields good performance in terms of accuracy and estimates the target location, velocity and trajectory.

Oguz-Ekim *et al.* (2009) have solved the problem of Euclidean Distance Matrix (EDM) completion to obtain initial sensor/target positions. The likelihood function is then iteratively optimized through either a Memorization-Minimization (MM) or Newton method. To reduce the computational load, they have proposed an incremental scheme whereby each new target position is estimated from range measurements, providing additional initialization for ML without the need for solving an expanded EDM completion problem.

Taylor *et al.* (2006) have introduced the Simultaneous Localization and Tracking, called SLAT, the problem of tracking a target in a sensor network while simultaneously localizing and calibrating the nodes of the network. Their proposed solution, LaSLAT, is a Bayesian filter that provides on-line probabilistic estimates of sensor locations and target tracks which does not require globally accessible beacon signals or accurate ranging between the nodes.

Bai *et al.* (2008) have studied the issue of optimal deployment to achieve four connectivity and full coverage for Wireless Sensor Networks (WSNs) under different ratios of sensors' communication range (denoted by  $rc$ ) to their sensing range (denoted by  $rs$ ). They have worked on the "Diamond" pattern that can be viewed as a series of different evolving patterns. When  $rc/rs \geq \sqrt{3}$  the Diamond pattern coincides with the well-known triangle lattice pattern; when  $rc/rs \leq \sqrt{2}$  it degenerates to a "Square" pattern. They have proved the Diamond pattern to be asymptotically optimal when  $rc/rs > \sqrt{2}$ .

Bai *et al.* (2006) have proposed an optimal deployment pattern to achieve both full coverage and 2-connectivity and have demonstrated its optimality for all values of  $rc/rs$ ; where  $rc$  is the communication radius and  $rs$  is the sensing radius. They have put forth a strip-based deployment pattern to achieve coverage and 2-connectivity and proved its optimality. They have also shown the optimality of a previously proposed strip-based deployment pattern to achieve coverage and 1-connectivity. Finally, they have established the efficiency of popular regular patterns of deployment, thus enabling a deploer make a more informed decision.

**Distributed connected coverage algorithm:**

**Problem definition:** Now we will define the connected sensor cover problem discussed in this study. We start with a few definitions.

**Definition 1 (communication graph; communication distance):** Consider a sensor network, consisting of a set of sensors  $S$ . Then the communication graph for the sensor network is the undirected graph  $CG$  with  $S$  as the

set of vertices and an edge between any two sensors, if they can communicate directly with each other.

A communication edge between the two given sensors is described to be an edge in the communication graph. A communication path among the sensors  $S_1$  and  $S_2$  has termed from a path sensors between  $S_1$  and  $S_2$  in the communication graph. The communication distance between two sensors  $S_1$  and  $S_2$  is the length of the smallest distance between  $S_1$  and  $S_2$  in the communication graph.

**Definition 2 (connected sensor cover; sensor cover):**

A sensor network comprising of  $n$  sensors  $S_1, S_2, \dots, S_n$  has been considered. The sensing region connected with the sensor  $S_i$  is uttered as  $R_i$ . If the following two conditions hold, a set of sensors  $M = S_{i1}, S_{i2}, S_{i3}, \dots, S_{im}$  is said to be a connected sensor cover for the specified query  $q$  over a region  $P_q$  in the network:

$$P_q \subset R_{i1} \cup R_{i2} \cup \dots \cup R_{im}$$

The subgraph induced by  $M$  in  $CG$  is connected, where  $CG$  is the communication graph of the sensor network. In other words, any sensor  $S_{ij}$  in the connected sensor cover can communicate with any other sensor  $S_{ik}$  in the cover, probably through other sensors in the selected set.

A set of sensors that satisfies only the first condition is called a sensor cover for  $q$  in the network.

Given a sensor network and a query over the network, the connected sensor coverage problem is to find the smallest connected sensor cover (Senthamilselvi and Devarajan, 2008; 2009).

**Criteria for the algorithm:** For choosing a connected sensor cover of near-optimal size, a greedy algorithm has been planned. A previously chosen sensor is linked through a path of sensors to an incompletely covered sensor at every stage. With the previously chosen sensors, the preferred path is subsequently added at this stage. When the chosen set of sensors entirely covers the specified query region, the algorithm finishes.

$M$  is assumed to be the set of sensors, which has been previously chosen for insertion in the connected sensor cover by the greedy algorithm at every stage. Initially,  $M$  is empty. The algorithm begins by including in  $M$  a random sensor lying inside the query's region. Based on a criterion portrayed in the subsequent section, the greedy algorithm chooses a sensor  $C$  along with a path of sensors  $P$ , at every stage. This path of sensors  $P$  forms a communication path between  $C$  and some sensor in  $M$ . The chosen path of sensors  $P$  that contains  $C$  is then appended to  $M$ . Thus, the

communication sub graph induced by  $M$  is connected at any stage of the algorithm (Senthamilselvi and Devarajan, 2009).

Each sensor  $S_i$  has a priority  $Pr_i$  assigned to it, based on its residual energy  $E$  and battery capacity  $W$  Eq. 1:

$$Pr_i = E_i + W_i, i = 1, 2, \dots, n \quad (1)$$

The priorities are sorted in the descending order. We now describe the criteria used in selection of  $C$  and  $P$  at any given stage of the algorithm. A sensor  $S_i$  is called as an active sensor if:

- $S_i \notin M$  and the sensing region of  $S$  intersects with the sensing region of some sensor in  $M$
- $Pr_i > W$ , where  $W$  is the minimum threshold value for the priority

A series of sensors forming a communication path linking an active sensor  $C$  with a few sensors in  $M$  is known as an active path. The length of an active path  $P_i$  is indicated as  $P1$ . The active path  $P$  that covers the maximum number of uncovered valid sub elements per sensor is added to  $M$  at that stage of the algorithm. An uncovered valid sub element is a valid sub element that is not covered by any sensing region of a sensor in  $M$ , the set of sensors already selected for inclusion in the connected sensor cover by the algorithm (Senthamilselvi and Devarajan, 2009).

**Algorithm description:** To build a connected sensor cover inside the sensor network for a specified query, the distributed coverage algorithm undergoes a sequence of stages. Throughout the entire execution of the algorithm, the sensor network maintains the following values:

- $M$ , a set of sensors that have already been selected for insertion in the connected sensor cover by the algorithm. The algorithm increments  $M$  by adding an active path of sensors to  $M$  at each stage
- $SP$ , a set of active paths. An active path is a sequence of sensors that form a communication path connecting an active sensor to some sensor in  $M$ , where an active sensor is a sensor whose sensing region intersects with some sensing region of a sensor in  $M$ . Each active sensor has exactly one active path associated with it
- $P1$ , the most recently added active path and  $C1$ , the active sensor associated with  $P1$

We assume that the sensing region of the contained sensor nodes is stored by each of the above values.

Each sensor in the network knows its membership in  $M$ , or  $P1$ , or in an active path in  $SP$ .

Also, the most recently added active sensor  $C1$  stores the values  $M$ ,  $SP$  and  $P1$ . The following sequence of transmission phases is contained in each stage of the distributed algorithm (Senthamilselvi and Devarajan, 2009).

**Active path search:** To choose new active paths and active sensors, the most newly added active sensor  $C1$  transmits an Active Path Search (APS) message to all sensors inside  $2r$  communication hops, where  $r$  is the link radius of the sensor network.  $2r$  has chosen with the intention that the APS message from  $C1$  reaches even those active sensors whose sensing disks intersect with that of other sensors in  $P1$ , the most recently added active path associated with  $C1$  the most recently added active path has been carried by the APS message.

**Active path response:** After receiving an APS message, any sensor  $S$  checks whether it is a new active sensor, i.e., if  $S$ 's sensing region intersects with the sensing region of some sensor in the most recently added active path  $P1$ .  $S$  then casts an Active Path Response (APR) message to the originating sensor  $C1$  of the APS message, if it is an active sensor.

**Selection of best active path/sensor:** The sensor  $C1$ , which was the originator of the APS messages in the current stage, collects all the APR messages sent to it by the active sensors. The active path  $P$  contained in each received APR message is added by  $C1$ , after appropriate truncation, to  $SP$ , the set of active paths being maintained by the sensor network. After having received all the APR messages sent to  $C1$  during this stage, the sensor  $C1$  selects the most advantageous active path  $P_{new}$  among all the active paths in  $SP$ . Let  $C_{new}$  be the active sensor associated with the new picked active path  $P_{new}$  and let  $S_{new}$  be the sensors in the active path  $P_{new}$ . The sensor  $C1$  unicast reliably, a NewC message to  $C_{new}$  with the following updated new information:

$$M = M \cup S_{new}; P1 = P_{new}; SP = SP \cup P_{new}$$

Note that  $SP$  has also been augmented with all the active paths received in the APR messages.

**Loop:** The sensor  $C_{new}$  receives the NewC message sent to it by  $C1$ . After receiving the message,  $C_{new}$  updates the value as,  $C1 = C_{new}$ . That marks the current stage of the new algorithm. The above process repeats until the selected set of sensors  $M$  cover the entire query region in the sensor network.

**Tracking with reduced sensor coverage:** In order to cover a region, we have designed an algorithm for selecting a reduced set of sensor nodes. The results obtained from Distributed Connected Coverage algorithm is used, to develop an energy efficient tracking algorithm using the minimum subset of sensor nodes (Senthamilselvi and Devarajan, 2011).

**Performance issues:** We have summarized some key design challenges for our proposed tracking algorithm in the wireless sensor networks:

- Large number of sensors
- Low energy use
- Network self-organization
- Collaborative signal processing
- Distributed processing
- Tracking accuracy
- Computation and communication costs
- Uncertainty
- Multi-modality sensor network
- Time synchronization

In this study, these issues are illustrated through the development of a reduced cover strategy which reduces the number of sensor nodes which are active at any given time.

Based on the reduced cover, tracking algorithms use only a fewer nodes and so they are efficient from an energy point of view. The energy consumption of the network can also be reduced by activating only those nodes in the surrounding area of the target which is being tracked. This can be achieved by activating the sensor nodes in the surrounding area of the sensor nodes which detected the target.

The following steps are involved in the energy efficient tracking of a target:

- Determining a reduced sensor coverage for the given targeted region
- Determining the boundary sensor nodes of the given targeted region
- Detecting the target entry or movement
- Broadcasting the coordinates of the boundary sensor node

To determine the Reduced sensor Coverage (RC), we use the distributed connected coverage algorithm described.

In the tracking algorithm, when a moving target is detected, a reduced sensor cover is established for the

region around the target. The movement of the target is detected by the boundary sensor nodes.

As soon as it detects the target, a boundary sensor node  $N_b$  broadcast its co-ordinates  $N_b(x,y)$  and speed of the target to the reduced cover nodes.

On receiving this information, each Node  $N_i$  from the reduced sensor coverage constructs the circular region  $RC'$  with centre at  $N_b(x,y)$  with radius  $r_s$ .

If  $\text{dist}(N_i, N_b) \leq r_s + RC'$ , then the node  $N_i$  can be added to the  $RC'$ . Then the status of the nodes will be kept active to track the further movement of the target.

When a any boundary sensor node  $N_{b2}$  on  $RC'$ , detects that the target is about to leave the region, a new sub-region  $RC''$  is created with the border sensor node  $N_{b2}$  as its center.

Then again the reduced cover nodes are activated based on the radius criteria. The same steps are repeated for successive movement of the target node. Since the radius of the circular region adaptively changes based on the speed of the target, the target can be tracked continuously. The distributed target tracking algorithm is given below.

#### Algorithm:

1. If  $N_i$  is a reduced cover node, then
  - 1.1 If it forms a border cover, then
    - 1.1.1 Detect the target
    - 1.1.2 Broadcast location  $N_i(x,y)$  to the reduced cover nodes.
  - Else
    - 1.2 Change the node status to sleep
- End if
- End if
2. If  $N_j$  receives the location information, then
  - 2.1 Form the circular region  $RC'$  with  $N_i(x,y)$  as the center.
    - 2.2 If  $\text{dist}(N_i, N_j) < r_s$ , the node is added to the sub-region.
      - 2.2.1 Change the node status to active.
      - 2.2.2 If border sensor  $N_k$  from  $RC'$  detects, the target movement, then
        - 2.2.3 repeat from step 1.1.2
    - Else
      - 2.3 Change the node status to sleep
3. Stop the algorithm.

**Target zone detection and localization:** To Achieve this, we first explain an algorithm for the target zone detection and then we present the procedure to estimate the locations of the targets from the sensors within the target zone.

**Target zone detection:** The objective of our target zone detection algorithm is to find all sensors which can detect the presence of targets. The nodes which are closer to the targets usually have higher measurements. Faulty sensors may report arbitrary values.

Let  $G(S_i)$  denote a bounded closed set of  $Q^2$  that includes a sensor  $S_i$  and additional  $n-1$  sensors. The set  $G(S_i)$  represents a closed neighborhood of the sensor  $S_i$ . An example of  $G(S_i)$  is the closed disk centered at  $S_i$  with its radius equal to the radio range. Let  $Q_{i1}, Q_{i2} \dots Q_{in}$  denote the signal strength measured by the nodes in  $G(S_i)$ . A possible estimate of signal strength at location  $S_i$  is Eq. 2:

$$\bar{Q}_i = \text{med}_i \quad (2)$$

where,  $\text{med}_i$  denotes the median of the set  $\{Q_{i1}, Q_{i2} \dots Q_{in}\}$ . In other words, one could estimate  $Q_i$  by the "center" of  $\{Q_{i1}, Q_{i2} \dots Q_{in}\}$ .

Note that  $\text{med}_i$  in (2) should not be replaced by the mean  $(Q_{i1}+Q_{i2}+\dots+Q_{in})/n$  of the set  $\{Q_{i1}, Q_{i2} \dots Q_{in}\}$ . This is because the sample mean cannot represent the center of a sample when some values of the sample are extreme. But, median is extensively used to estimate the center of samples with outliers. Faulty sensors may have extreme values, representing outliers in the sample set. Faulty readings have little influence on  $\text{med}_i$  as long as most sensors behave properly.

The following is the procedure of the target zone detection.

Intuitively, an event sensor is a sensor that can detect the presence of the targets. Compared to the value fusion method for target zone detection which computes the mean after dropping  $\lambda$  highest and  $\lambda$  lowest values, Algorithm 2 employs the robust operator median so that it effectively eliminates the effects of faulty sensors without exploiting any complicated algorithm for the estimation of  $\lambda$ .

**Target zone detection algorithm:** For any given sensor  $S_i$ :

- Acquire signal measurements  $Q_{i1}, Q_{i2} \dots Q_{in}$  from all sensors  $G(S_i)$
- Calculate  $\text{med}_i$  of the set  $\{Q_{i1}, Q_{i2} \dots Q_{in}\}$  as the expected reading  $\bar{Q}_i$  at the location  $S_i$
- Establish event sensors. A sensor  $S_i$  is an event sensor if the expected value  $\bar{Q}_i$  is larger than a predefined threshold  $T_1$

**Target localization:** The algorithm presented above is used to detect the presence of targets. It does not

mention the total number of targets and the location of the targets. Due to energy consumption, it is too expensive to shift the task of targets localization to the base station by sending the measurements of all sensors in the target zone. Hence, we consider assigning one sensor to communicate with the base station for each target and estimate the position of the target locally. In order to locate the targets in the target zone, the following algorithm is used.

**Algorithm:**

- Acquire expected signal strength  $\bar{Q}_{i1}, \bar{Q}_{i2}, \dots, \bar{Q}_{im}$ , from all  $m$  event sensors in  $G(S_i)$  if  $S_i$  is an event sensor
- Establish root sensors. An event sensor  $S_i$ , is a root sensor if  $m \geq n/2$  Eq. 3:

$$\bar{Q}_i \geq \max \bar{Q}_{i1}, \bar{Q}_{i2}, \dots, \bar{Q}_{im} \quad (3)$$

- For each root sensor  $S_i$ , estimate the location of a possible target by the geometric center of event sensors  $G(S_i)$ . Let  $\{S_{i1}, S_{i2}, \dots, S_{ik}\}$  be the subset of event sensors in  $G(S_i)$  such that  $\bar{Q}_{ij} \geq \bar{Q}_i - T_2$  for  $1 \leq j \leq k$ , where  $\bar{Q}_{ij}$  is the expected signal strength from  $S_{ij}$  and  $T_2$  is a threshold that mainly characterizes the target size. Denote the  $x$  and  $y$  coordinates of  $S_{ij}$  by  $x(S_{ij})$  and  $y(S_{ij})$  respectively and set Eq. 4:

$$\begin{aligned} X_i &= (x(S_{i1}) + x(S_{i2}) + \dots + x(S_{ik}))/k \\ Y_i &= (y(S_{i1}) + y(S_{i2}) + \dots + y(S_{ik}))/k \end{aligned} \quad (4)$$

$X_i$  and  $Y_i$  are the estimated coordinates for a target close to  $S_i$ .

Note that in step (1) of the algorithm,  $m$  can be smaller than  $n$ . A sensor is selected as a root sensor if its estimated signal strength is a maxima among event sensors in  $G(S_i)$ . Nodes closer to the targets usually have larger measurements and thus have a higher probability to become root sensors. Furthermore, the number of root sensors is constrained by (4). A root sensor uses (5) to compute the location of a target based on the locations of some neighboring nodes.

## RESULTS

**Simulation parameters:** We evaluate our Distributed Coverage and Target Tracking Algorithm for Wireless Sensor Networks (DCTT) through NS2 simulation Network Simulator. We use a bounded region of  $1000 \times 1000$  sqm, in which we place nodes using a

uniform distribution. We assign the power levels of the nodes such that the transmission range and the sensing range of the nodes are all 250 m. In our simulation, the channel capacity of mobile hosts is set to the same value: 2 Mbps. We use the Distributed Coordination Function (DCF) of IEEE 802.11 for wireless LANs as the MAC layer protocol. It has the functionality to notify the network layer about link breakage. In our simulation, sensor nodes of sizes 25, 50, 75 and 100 are deployed in a 1000×1000 m rectangular region for 50 sec of simulation time. The simulated traffic is Constant Bit Rate (CBR). To measure the performance of different protocols under different ratios of communication range/sensing range, we varied the communication range by 250, 300, 350 and 450 m, in the network interface. All experimental results presented are averages of five runs on different randomly chosen scenarios. The following Table 1 summarizes the simulation parameters used.

**Performance metrics:** We compare the performance of our proposed DCTT protocol DTT (Wang *et al.*, 2008). We mainly evaluate the performance according to the following metrics:

**Average energy consumption:** The average energy consumed by the nodes in receiving and sending the packets are measured.

**Sensor coverage:** It is the number of sensor nodes that forms connected sensor coverage under the corresponding protocols.

**Average tracking delay:** Average time involved in target tracking.

### DISCUSSION

**Effect of varying nodes:** In the first experiment, we vary the number of nodes as 25, 50, 75 and 100 and measure the above metrics.

Figure 1 shows the energy consumption values when the number nodes are increased. From the Fig. 1, it can be seen that the energy consumption is less in the case of DCTT and outperforms DTT algorithm.

Figure 2 shows the sensor coverage obtained, when the number of nodes are increased. It shows that DCTT utilizes less coverage area than the DTT algorithm.

Figure 3 shows the tracking delay values when the number of nodes are increased. It is clear that DCTT has less delay when compared to DTT algorithm.

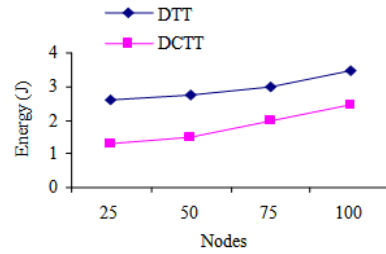


Fig. 1: Nodes Vs energy

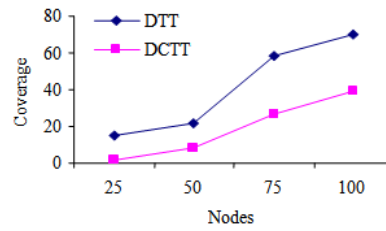


Fig. 2: Nodes Vs coverage

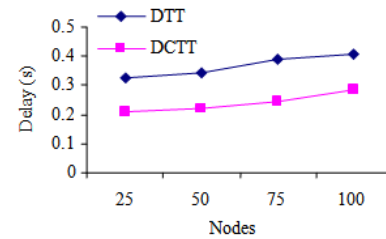


Fig. 3: Nodes Vs tracking delay

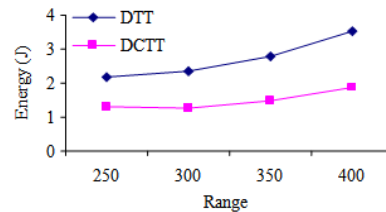


Fig. 4: Transmission Vs energy

**Effect of varying transmission range:** In the second experiment, we vary the transmission range as 250, 300, 350 and 400. Fig. 4 shows the energy consumption values when the transmission range is increased. From the figure, it can be seen that the energy consumption is less in the case of DCTT and outperforms DTT algorithm.

Figure 5 shows the sensor coverage obtained, when the transmission range is increased. It shows that DCTT utilizes less coverage area than the DTT algorithm.

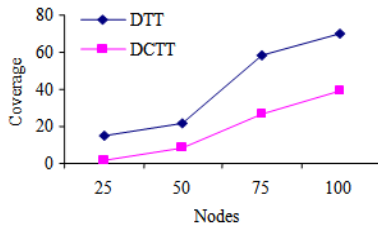


Fig 5: Transmission Vs coverage

Table: 1 Simulation settings

No. of nodes	25, 50, 75 and 100
Area size	1000x1000
Mac	802.11
Simulation time	50 sec
Traffic source	CBR
Packet size	512
Transmit power	0.360 w
Receiving power	0.395 w
Idle power	0.335 w
Transmission	250,300,350 and
Range	400
Routing protocol	AODV

## CONCLUSION

In this study, we have proposed a distributed coverage and Target Tracking Algorithm for Wireless Sensor Networks. Tracking issue is first addressed through the determination of a reduced cover for the region of interest. For reduced coverage, we have used the distributed connected coverage algorithm. Tracking algorithms are developed using a reduced set of sensor nodes. A border cover node is included in the tracking algorithm and it should send its co-ordinates which results in increased overhead and it is more expensive. Therefore, we have used a self localization technique in which we propose a target zone detection algorithm and then we present the procedure to estimate the locations of the targets from the sensors within the target zone to determine the co-ordinates of the sensor nodes. By simulation results, we show that the proposed algorithm achieves reduced energy consumption and coverage along with delay.

## REFERENCES

Bai, X., S. Kumar, D. Xuan, Z. Yun and T.H. Lai, 2006. Deploying wireless sensors to achieve both coverage and connectivity. Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, May 22-25, ACM New York, N.Y., pp: 131-142. DOI: 10.1145/1132905.1132921

Bai, X., Z. Yun, D. Xuan, T.H. Lai and W. Jia, 2008. Deploying four-connectivity and full-coverage wireless sensor networks. Proceedings of the 27th Conference on Computer Communications, Apr. 13-18, IEEE Xplore Press, Phoenix, AZ, pp: 296-300. DOI: 10.1109/INFOCOM.2008.68

Estrin, D., R. Govindan, J. Heidemann and S. Kumar, 1999. Next century challenges: Scalable coordination in sensor networks. Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, (MCN' 99), ACM New York, N.Y., USA., pp: 263-270. DOI: 10.1145/313451.313556

Estrin, D., L. Girod, G. Pottie and M. Srivastava, 2001. Instrumenting the world with wireless sensor networks. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, May 7-11, IEEE Xplore Press, Salt Lake City, Utah, USA., pp: 2033-2036. DOI: 10.1109/ICASSP.2001.940390

Li, D., K.D. Wong, Y.H. Hu and A.M. Sayeed, 2002. Detection, classification and tracking of targets. IEEE Signal Process. Mag., 19: 17-29.

Liu, H.Q., H.C. So, K.W.K. Lui and F.K.W. Chan, 2009. Sensor selection for target tracking in sensor networks. Prog. Elect. Res., 95: 267-282.

Mao, G., B. Fidan and B.D.O. Anderson, 2007. Wireless sensor network localization techniques. Comput. Netw., 51: 2529-2553. DOI: 10.1016/j.comnet.2006.11.018

Oguz-Ekim, P., J. Gomes, J. Xavier and P. Oliveira, 2009. MI-Based sensor network localization and tracking: Batch and time-recursive approaches. Proceedings of the 17th European Signal Processing Conference, Aug. 24-28, EURASIP, Glasgow, Scotland, pp: 80-84.

Senthamilselvi, M. and N. Devarajan, 2008. An energy efficient distributed algorithm for connected sensor cover in sensor networks. Int. J. Comput. Sci. Netw. Security, 8: 265-271.

Senthamilselvi, M. and N. Devarajan, 2009. Energy efficient distributed connected coverage algorithms with dynamic coverage maintenance in wireless sensor networks. Proceedings of the IEEE International Advance Computing Conference, Mar. 6-7, IEEE Xplore Press, Patiala, pp: 1580-1586. DOI: 10.1109/IADCC.2009.4809254

Senthamilselvi, M. and N. Devarajan, 2011. An energy efficient tracking algorithm for reduced coverage in wireless sensor networks. Eur. J. Sci. Res., 62: 282-293.



- Taylor, C., A. Rahimi, J. Bachrach, H. Shrobe and A. Grue, 2006. Simultaneous localization, calibration and tracking in an ad hoc sensor network. Proceedings of the 5th International Conference on Information Processing in Sensor Networks, Apr. 19-21, Nashville, TN, USA., pp: 27-33. DOI: 10.1145/1127777.1127785
- Watfa, M.K. and S. Commuri, 2006. An energy efficient approach to dynamic coverage in wireless sensor networks. *J. Netw.*, 1: 10-20.
- Wang, Z., E. Bulut and B.K. Szymanski, 2008. Distributed target tracking with imperfect binary sensor networks. Proceedings of the IEEE Global Telecommunications Conference, Nov. 30-Dec. 4, IEEE Xplore Press, New Orleans, pp: 1-5. DOI: 10.1109/GLOCOM.2008.ECP.62
- Zhang, H. and J.C. Hou, 2005. Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc Sensor Wireless Netw.*, 1: 89-124.