**The Caspian Sea Journal**
ISSN: 1578-7899

Volume 10, Issue 1, Supplement 4 (2016) 230-243

# A Novel Method for Target Tracking in Wireless Sensor Networks Using Ant Colony Algorithm

Mohsen Emami,
Dept. of Computer Engineering and Information Technology, Payame Noor University, PO BOX 19395-3697, Tehran, Iran, Email: emami.mohsen@pnu.ac.ir
Mostafa Hosseini,
SGNEC, #16, Shohada st. Farahani ave., Tehran, Iran, Email: mostafah@systemgroup.net
Mohammadali Karimi
University of Tehran, PO BOX: 11365-4563, Tehran, Iran, Email: Karimi.Mohammadali@gmail.com

***ABSTRACT*** — Recently different suggestions have been proposed on improving wide applications and challenges in wireless sensor networks. These challenges include energy consumption, accuracy, error prevention and correction, reliability and security. According to limited energy source of sensor nodes and distributed nature of WSN, using distributed agent systems and multi agent solutions would help to improve overall performance of the network. Agents are able to solve complex problems using group work and intelligence. They can decide autonomously though they can work as a team and as a result improve speed of target detection, quality and accuracy of tracking and also they can lead to save energy. In this paper we combine the power of agent based corporation for target tracking with ant colony optimization algorithm for routing and clustering, to improve performance of tacking an object in WSN environment. The simulation results approve that the proposed method improves target tracking accuracy without any negative effect on power consumption.
**KEYWORDS**: Wireless sensor network, clustering, target tracking, ant colony optimization, intelligent agent;

## 1. Introduction

Environment surveillance and target tracking is one of the most attractive applications of wireless sensor networks (WSN). Many studies have been conducted on overcoming challenges of this field (1,2).In general, tracking algorithms should be able to identify the target and determine its location within an acceptable time and with desired accuracy. To reach this goal, sensors in the environment have to communicate with each other and transfer collected data. Data transfer can be done in a distributed manner through the neighbor nodes or direct transfer to sink. Direct data transfer consumes huge amount of energy of each transmitting node. In return distributed data transfer might decrease overall network consumed energy using an optimum routing algorithm. Each wireless sensor node has limited source of energy. So minimizing the energy consumption will prolong the overall network's life. This is the reason that minimizing battery usage is one the most important challenges in designing algorithms and protocols of WSN. Developing topologies and algorithms which increase accuracy and quality of applications while decreasing consumed energy have been always considered by WSN researches. Sensor network topology defines the communication and management approach of wireless sensor network and has direct effect on target tracking method.In this study we focus on designing and developing a new algorithm to improve the accuracy of target tracking in a WSN environment. This algorithm is optimized to preserve energy consumption while sensing the object, routing and transmitting data to the sink. These challenges are handled by deploying intelligent agents concept and also ant colony optimization algorithm -a novel heuristic optimization method-. Intelligent agents are mostly used in disturbed systems and so are profitable for WSN. Simulation results have revealed enormous increase in target tracking accuracy and performance. The rest of this paper is organized as follows. Chapter 2 reviews agent and multi-agent systems and specification. The proposed method is presented in chapter 3. In chapter 4 we illustrate the simulation results. Finally, conclusion and future works will be discussed in chapter 5.

### Agent and Multi Agent Systems

Agent is an autonomous software that can migrate and transfer its code and status from one host to another in a network. In other words, agents are smart programs that learn and predict user requirements, and act based on this prediction. They can also collaborate with each other (3). Figure1 shows a schematic overview of agents interacting with environment. The environment in this Figuremight be a user or just a physical environment.

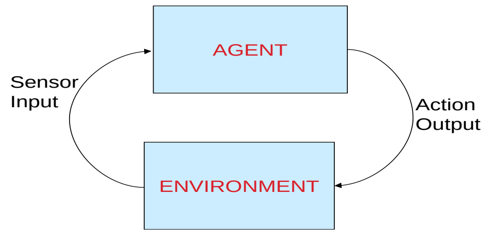


**Figure1** Architecture of an agent. The environment might be a user or just a physical environment

Agents have four major specifications (4):

- Autonomy: the ability of making decisions without direct action of a human.
- Reactivity: refers to the fact that agents need to know the environment around them. This environment can be the physical world, or a user via GUI or a group of other agents.
- Social ability: the ability of connection and wise collaboration between other users, agents and environment for collecting data and informing results.
- Proactivity: agents must be proactive, i.e. providing the precondition circumstances for reaching the goal by absorbing precise and suitable ideas from environment.

For implementing multi agent systems, all mentioned properties are required.

## Proposed Method

### Network Topology

In the proposed method we use clustering which provides a profitable network of nodes structure. In clustering methods the network is divided into smaller areas and each area will be managed by a chosen node named Cluster Head (CH). Using this method we can easily manage large scale distributed nodes. CHs are responsible for collecting cluster members' data and route it to sink node through an optimized path (5).In this method clusters should be 2-hop which means that each node should be at most two hops far from the corresponding cluster-head. The process of making two hop clusters is as follows: each node that either becomes CH or join a cluster, sends a message to its first and second level neighbors declaring the cluster ID of the CH. Each node might receive two types of message, first one is mode1 message that is received from a direct CH neighbor and second one is mode2 message that is received from a direct neighbor that is first level member of a CH. In other words when a node receives mode1 message it can become a first level neighbor of a CH and when receives mode2 message it can become a 2-hop neighbor of a CH. When a node receives more than one message, it decides based on three occurred conditions:

1. One mode 2 message is received: In this case, mode1 messages are dropped and the received CHID will be chosen.
2. More than one mode2 messages are received: In this case, mode1 messages are dropped and CH will be chosen using the decision algorithm described later.
3. No mode2 messages are received and mode 1 messages are more than zero: In this case CH will be chosen using decision algorithm.

Each CH candidate sends a message to its neighbors asking if they are members of any other CH in their first or second level neighborhood before becoming CH. The candidate cannot become a CH if at least one neighbor answers positively to the query. Each node waits for "Ts" milliseconds before answering CH candidate queries. During this period the node receives all possible CH candidacy queries and when TS is over, it decides about choosing the CH and then answers the declarations. Each CH candidate waits for "Tc" milliseconds (which is greater than Ts) after sending candidacy query to neighbors. The candidate can become a CH and declare it if no positive answer is received from neighbors during this period. A Tg timer is also set in each node during this process. If any node cannot decide to become a member or CH when Tg times out, it will be joined to the nearest cluster as a guest. This timer is longer than other timers and defines when the CH decision algorithm forcibly ends.

Possible states of a node in decision algorithm are Free, Cluster Head, Cluster Member and Guest. These states are depicted in Figure 2. When a node has not decided about becoming a CH or a member it is in free state. If it decides to become a CH or a member node, it would transit to cluster head or cluster member state respectively. If no decision is made during TG time, the node would become a guest member and transit to guest state. The proposed clustering algorithm is depicted in Figure 3.

For the first time, the decision algorithm is executed while network is initializing. Network clustering topology might need to be restructured later when power of CHs decrease during network transmits data. Restructuring causes flood effect in clustered network and wastes a lot of energy during each flood.

In the proposed method, a local election for new CH would occur instead of restructuring the whole network to prevent flood effect. Effective parameters in selecting the new CH are the remaining energy level and distance between cluster node and current CH. Equation (1) shows the candidacy index calculated for each node.

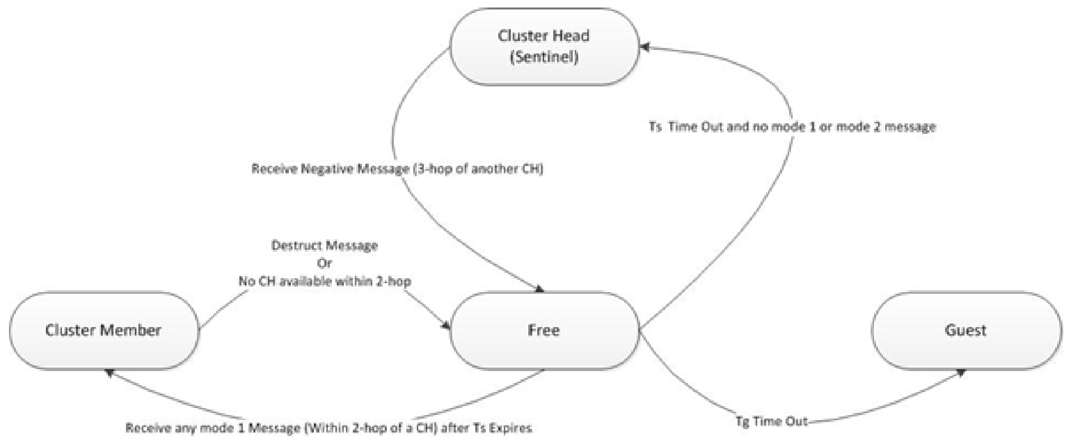


**Figure 2** Possible states of clustering algorithm. Node states and transitions are depicted. The agent is in free state initially.

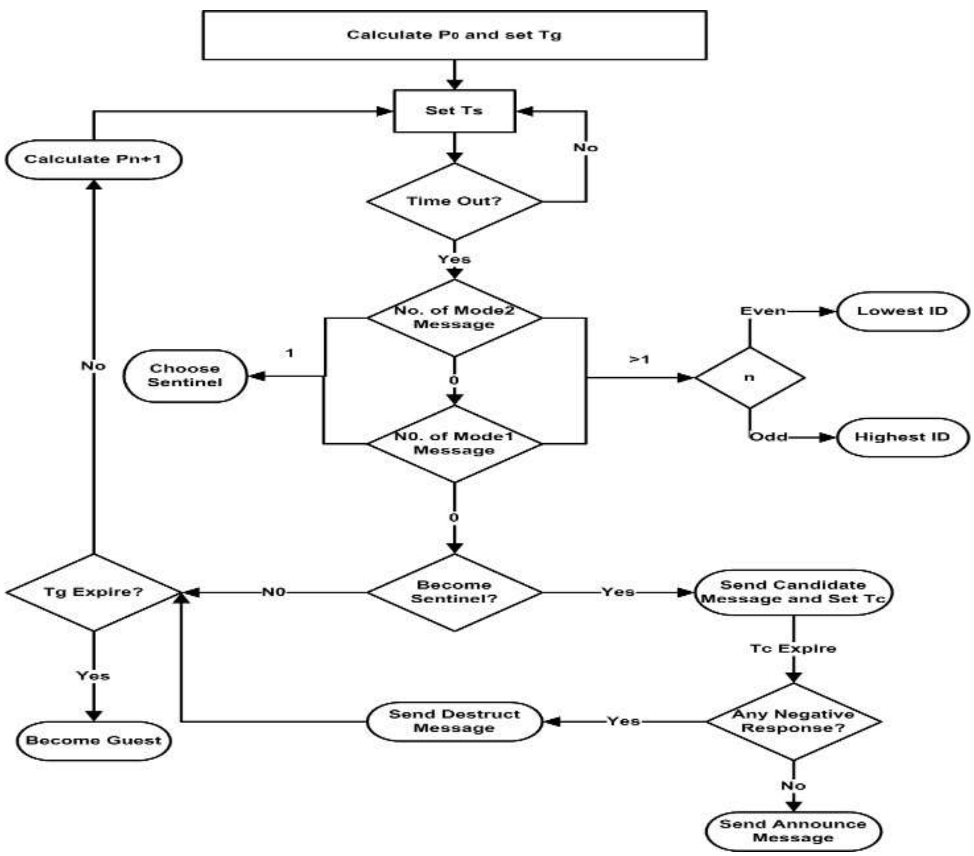


**Figure 3** Clustering Algorithm. This algorithms results in a 2-hop cluster structure. Each node might become a cluster head, member node or guest.

$$Candidacy\_Index = \left(\frac{Ei_{Now}}{E_{Max}}\right)^{\alpha} \times \left(\frac{1}{d_{hi}}\right)^{\beta} \quad (1)$$

When a CH consumes two third of its energy, a message is broadcasted in the cluster. It causes each cluster member to send to the CH information such as id, cluster id and remaining energy. The CH calculates candidacy index for each node based on received information and distance to node. It elects the node which has maximum index to be the new cluster and then transmits the required history to new CH and changes its state to member node. Due to above equation the node which has much remaining energy and less distance would be elected.

**Target Tracking**

It is assumed that all sensors are able to find their locations via GPS, know the location of their neighbors, sense target, do calculations and communicate with each other (6). Each sensor can track the target in radius r. The distance between node and target is determined by received signal strength from target.

In the proposed method at least three nodes should sense the target to determine the location of it. Suppose that the target is in triangle of sensors S1, S2 and S3. The distance of target from each sensor is r1, r2 and r3 respectively. As it is depicted in Figure 4, two probable locations are found by S2 and S3 and when we use S1 we can find one point that is an acceptable estimation of the true location of target (7).

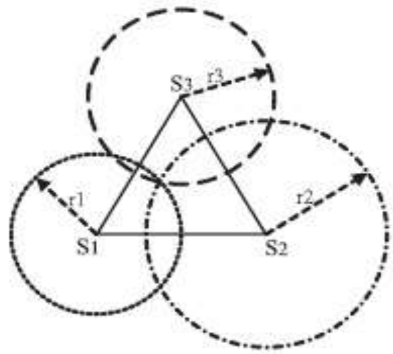


**Fig 4.** Calculating Location of Target. Three sensors S1, S2 and S3 sense the location of target and make a triangle. Sensors can sense the target in distances of r1, r2 and r3.

The area inside the triangle of three tracking nodes is called working zone in which sensors work normally together to find the target. As shown in Figure 5 there are also A1, A2 and A3 zones which handover process occurs when the target is inside them. This process is described later.

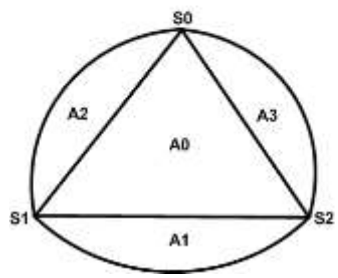


**Figure 5** Regions of target sensing. A0 is the working zone and A1, A2 and A3 are handover zones for sensors S1, S2 and S3

**Protocol Basics**

Target tracking is done by cooperation of nodes belonging to one cluster. When a target is identified, the agents in nearest sensors activate to track it. An agent would migrate between sensors during tracking process. This fact is depicted in Figure 6.

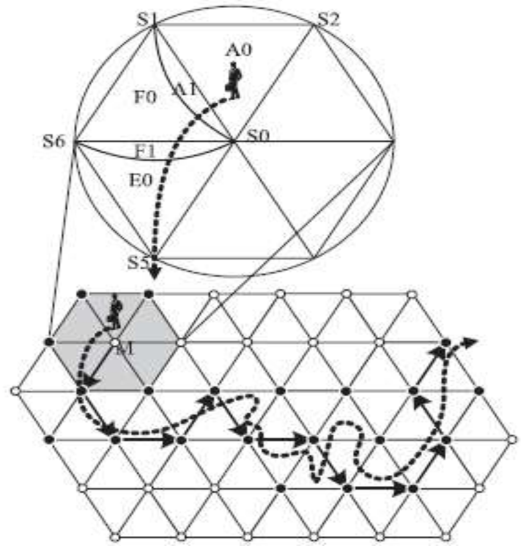


**Figure 6** Sample path of a target. S0 to S6 are sensors. A0, A1, F0, F and E0 are different working and handover zones. Dark points are master and slaves nodes during movement of the target.

One of the nodes is elected as master and sends out two slave agents to invite two neighbour nodes for corporation. These three nodes corporate in creating a triangle and execute target location finding algorithm. Slave agents send their detection results to the master agent and the master calculates accurate location of target using these information. A minimum threshold of signal strength is predefined for slaves. Slave agents would migrate to another sensor if signal strength becomes less than this threshold. The master node decides about result reporting period to prevent information transfer overload. In Figure 6 when the target enters A1 zone, detected signal strength of S2 weakens and becomes lower than minimum threshold. So the slave agent in S2 is deactivated and slave agent in S6 becomes active. Similarly when target enters F1 zone the slave agent of S1 deactivates and slave agent of S5 becomes active. The master agent is not able to detect the target when it reaches S5 and so the master agent transfers to S5. Old slave agents are deactivated and new slave agents are elected. When the target is inside backup zone of other sensors, many nodes might be able to detect the target but master and slave agents inhibit other agents from target detection to reduce energy consumption.

**Protocol Details**

At first, a unique identification code is generated for each target by which it is identified throughout the system. The proposed design is capable of detecting multiple targets simultaneously. Her we explain detection of single target because detection of each target is completely separate and do not interfere in each other’s work. Figure8 shows state diagram of each sensor. A sensor might be in a different state for each object while detecting multiple targets.

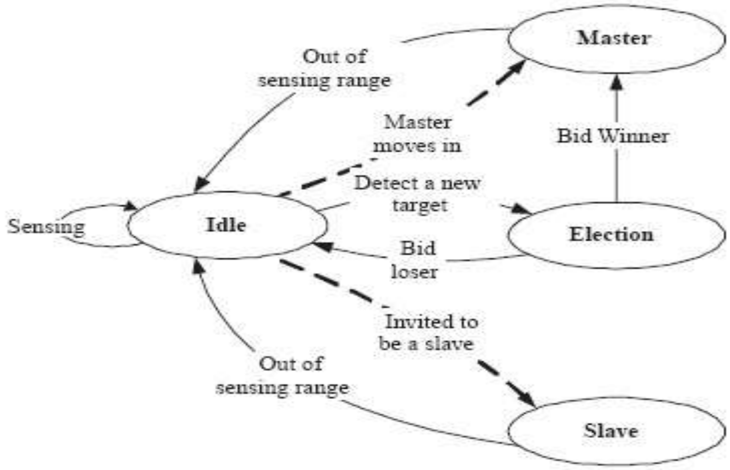


**Figure 7** State diagram and transitions of an agent node. An agent is in idle state initially.

A sensor is in idle state at first. While working in this state, it can detect every target in its detection range. It transits to election state as soon as it detects presence of an object and starts selection protocol to become either master or slave. The closest sensor to target is selected as master and sends slave agents to two sensors closer to itself. The master node then transits to master state to execute master agent protocol. The slave sensors also transit to slave state to execute slave agent protocol. The system works in this state while target is in working zone of nodes to reduce migration of agents. When the target enters backup zone, master and slave roles might be changed. The nodes that detect the target but receive inhibiting message also remain in idle state.

Sensors might lose the target because of it's high speed. In this case agents would become deactivated for a while to prevent sending inhibit messages and new nodes are elected for target tracking. This causes increase of system fault tolerance.

**Sentinels Routing**

Routing results and information from sentinel to sink is very important (8–11). This issue becomes more important if cluster heads are spread over a wide area. One solution is to send information to the sink directly. In this case sentinels should be aware of the location of the sink. This solution is simple and fast but it wastes intolerable amount of energy. So an applicable solution would be sending data via intermediate sentinels. A cluster head transmits data to neighbour sentinel and the information is transmitted to the sink so on. Swarm intelligence based algorithms are one of the most popular optimization algorithms. Furthermore it can be said that ant colony is the best way to find optimized routes in distributed systems. As this algorithm was firstly used in multi agent systems so we can implement it in agents of sensor nodes.

**Description of ant colony algorithm**

To start this algorithm, the sink sends a hello word to all nodes during cluster initialization and primary pheromone assigning, that causes the nodes to start and also get the location of sink. The nodes that receive this message send it to their neighbours. If we assume the message that has been received earlier by a node is sent from a node which is closer than other neighbours to the receiver, it is easy to find the nearest neighbour and initial pheromone values would be assigned to paths.

**First step: initialization**

At first, K ants are initialized in each cluster head. Three matrices named Tabu, R-best and A-city are initialized and an initial pheromone value is assigned to paths. Tabu includes the determined path. R-best keeps the best path between cluster head and sink. A-city includes nodes visited through the path to prevent ring creation.

**Second step**

K ants in each sentinel transmit to one of the neighbor sentinels with probability P. the selected sentinel is added to Tabu and is put on visited nodes group in A-city. Then pheromone is secreted in selected path. The neighbor selection probability function is depicted in Eq. (2).

$$P_{ij}(t)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\times\eta_{ij}^{\beta}(t)}{\sum_{K}\tau_{ij}^{\alpha}(t)\times\eta_{ij}^{\beta}(t)}, & \forall j\in N_i \text{ and } j\notin M^k\\ 0, & otherwise\end{cases} \tag{2}$$

In this equation is heuristic function between two nodes and is the value of path pheromone. and variables are controllers of importance of two functions. is the set of neighbor nodes. If a node is visited already, its probability function would become zero for that edge.

**Third step**

All paths are rated with pheromones when all the ants reach the sink. Steps two and three are repeated iteratively to select the best paths.

**Fourth step**

The cluster heads send their information from the optimized path until the remaining energy of one of involved sentinels reach minimum threshold and it is needed to elect a new sentinel.

**Heuristic function design**

heuristic function should be chosen in a way that probability of selecting optimized path maximizes. The function could be based on remaining energy of neighbours, distance of neighbours or both of them. Equations (3) and (4) show the heuristic and pheromone production functions used in proposed algorithm.

$$\eta_{ij}(t)=\frac{1}{D(i,j)}=\frac{1}{\sqrt{(x_j-x_i)^2+(y_j-y_i)^2}} \tag{3}$$

$$\tau_{ij}\leftarrow\mu\tau_{ij}+(1-\mu)\frac{E_{Rj}}{E_{init}} \tag{4}$$

By choosing these two functions, probability function would not become too much complex and also both factors of energy and distance are involved in finding optimized path. In addition, a new ant is sent back to the source via received path after receiving ant from sentinel to update pheromone according to Eq. (5) which optimizes the path based on path efficiency computed by sink. The produced pheromone should evaporate with an evaporation factor gradually. It is obvious that acts like evaporation factor. is also the overall efficiency function for path p.

$$\tau_{ij} \leftarrow \lambda\tau_{ij} + (1-\lambda)F_p \quad (5)$$

**Simulation Results**

There are many WSN simulators from which we have selected Castalia. Castalia is a simulator for wireless sensor network, body area network and low energy distributed systems. This simulator is based on OMNeT++ platform and is an ideal event based environment for testing distributed algorithms and protocols, with perfect radio channel model (12).

**Evaluation parameters**

The following three parameters are evaluated in the proposed algorithm.

1. Target positioning error: We can calculate the error between computed position and actual position of target when sensors verify target and information is sent to sink. This parameter shows the accuracy of proposed target positioning algorithm.
2. Target miss rate: when the target runs out of reach of sensors and agents cannot verify it correctly, we say the target is missed. This parameter helps to test power of the proposed method in tracking targets in different scenarios with different speeds. The parameter is investigated deeper in later chapters.
3. Remaining or consumed energy of nodes: Energy plays an important role in wireless sensor networks. It is feared that too much energy is used by the system due to calculations and complex implemented algorithms.

**Simulation environment**

The simulation environment is square shaped that its size varies with number of nodes. The space between nodes is assumed to be 5 meters. For example if we want to locate 25 nodes, a square of 20 x 20 with 5 nodes in each edge is needed that the distance between each two nodes is 5 meters. This scenario is depicted in Figure 8.

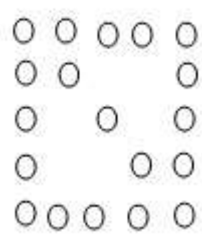

**Figure 8** Sample square 2-d simulation environment with 25 nodes. The distance between each two nodes is 5 meters.

**Simulation scenario**

Three main scenarios are simulated each with 25, 100 and 225 nodes. 3 different target speeds are considered for each main scenario that include 0.5 m/s, 1 m/s and 2 m/s. The path of target is designed linear or zigzag.

- Linear path: in which the target starts running from point (0, 0) and ends up in point (n, n). The linear path of target is shown in Figure 9.

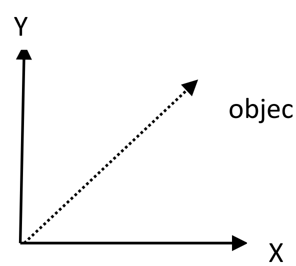


**Figure 9** Linear path of a sample object. The object starts from point (0,0) and moves in a linear path with a constant speed.

- Zigzag path: the zigzag path and milestone points are shown in Figure 10 The path between each two point is linear.

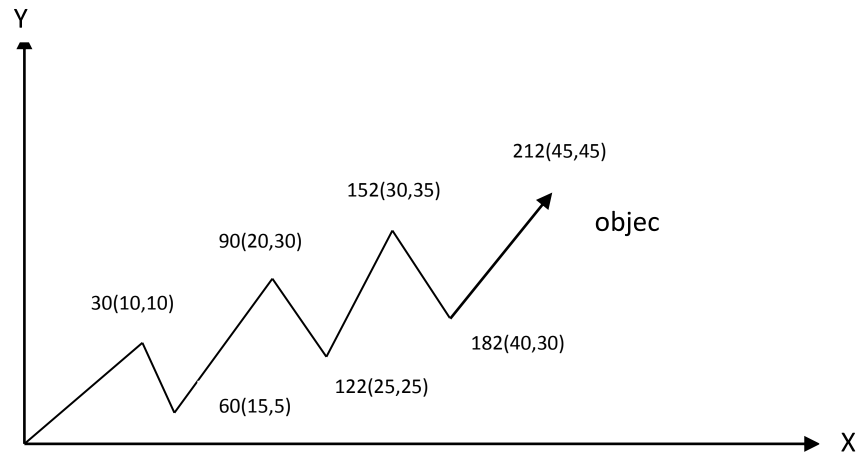


**Figure 10** Zigzag path of a sample object. The start time and location of each milestone point is depicted. The object starts from location (0, 0). The object moves between points with constant speed but the speeds on different linear paths are different. Paths are generated randomly.

In this Figurethe first number shows the start time and the numbers in parentheses show the location of point. It is obvious that the speed and direction of target is different in each direct path. This scenario is picked randomly and doesn't follow any specific model. To reduce random error we execute each scenario 5 times and report the average as the final result.
The specifications of each scenario are summarized in Table 1.

**Table 1** Specifications of simulation scenarios.

| Scenario | Movement | Speed | # of Nodes |
|---|---|---|---|
| **25_0.5** | Linear | 0.5 m/s | 25 |
| **25_1** | Linear | 1 m/s | 25 |
| **25_2** | Linear | 2 m/s | 25 |
| **100_0.5** | Linear | 0.5 m/s | 100 |
| **100_1** | Linear | 1 m/s | 100 |
| **100_2** | Linear | 2 m/s | 100 |
| **225_0.5** | Linear | 0.5 m/s | 225 |
| **225_1** | Linear | 1 m/s | 225 |
| **225_2** | Linear | 2 m/s | 225 |
| **ZigZag** | ZigZag | Random | 225 |

**Base comparison algorithm**

In order to better understand the effects of using proposed method we should compare it with a base algorithm. To ensure the equality of execution environments we develop and implement both algorithms with Castalia.The agents are present in base method and work just like the proposed method, but the proposed clustering algorithm is omitted and the nodes are clustered with common algorithm. Also novel ant colony routing algorithm is disabled and traditional flood based routing protocols are used which are implemented by Castalia.

**Target positioning error**

Figure11 compares the error of real position and estimated position of target between proposed and based algorithms in different scenarios. It is obvious that this error is below 1 meter for the proposed method in almost all scenarios, even though the error raises up to more than 5 for the base method.The main reason of this improvement is clustering method which causes better communication between nodes. Also ant colony algorithm helps the sentinels not to waste processing resources on routing, with finding the best route effectively and delivering data packets to the sink in shortest time possible.

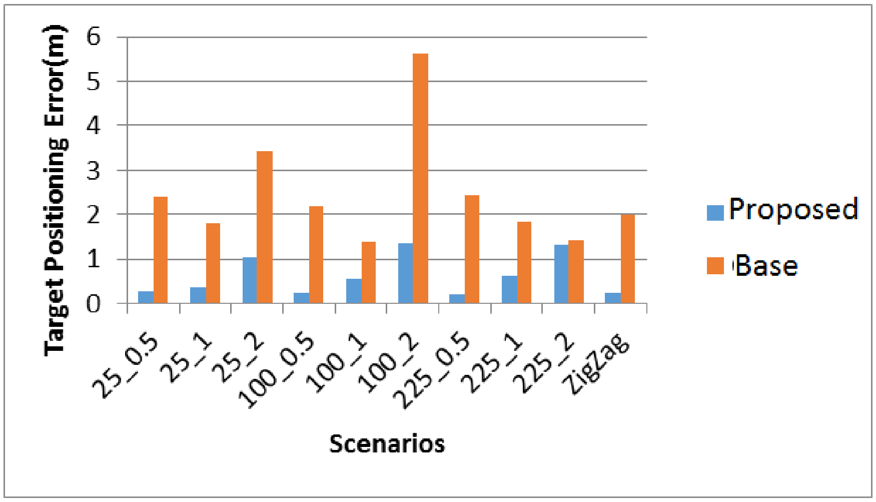


**Figure 11** Target positioning error for proposed and base algorithms. The vertical axis shows the error between real and estimated position of target in meters. The horizontal axis shows the simulation scenario. On this axes the first number shows the number of nodes and the second number shows the speed of target in m/s. The last scenario is zigzag as described in chapter 4.2.1.

The scenarios of 100 nodes with speed of 2 m/s and zigzag show the most difference between two compared algorithms. In these scenarios the target positioning error of proposed algorithm is less than base method 5 times and 10 times respectively. Much difference in zigzag scenario is caused by varying speed and direction of target. Also inappropriate clustering used by base algorithm, results in improper target tracking in crowded scenario of 100 nodes when target is running with higher speeds.

**Target miss rate**

One of the important characteristics to evaluate is the miss rate of target during execution. We should find a parameter that could well represent this characteristic.

We can consider a target to be missed if it takes more than  seconds from the last time that sink was informed of it. The number of times a target is missed during execution time can be a good representative for miss rate. But as execution time differs in different scenarios it might be better to divide this number by execution time to calculate miss rate per time unit.

If a target is missed fewer in number of misses but with a longer time period, the parameter would lead us to wrong result of better target tracking because of just counting the number of missing the target. So we should refine the parameter in a way that it is affected by missing time too. The miss count can be calculated as division of total miss time to miss time threshold to fulfil this requirement. It means that if a target is missed for 5 seconds and the target miss threshold is 1 seconds, it counts 5 misses for it. Equation (6) shows the calculation formula for this parameter.

$$\text{Miss rate} = \frac{t_{miss}}{t_{thresh} \times t_{sim}} \quad (6)$$

in which  is the total target miss time,  is the threshold time for target to be considered missed and  is the total simulation time. Figures 12 to 16 show the comparison results of miss rate between the proposed method and base method for different simulation times. The vertical axis shows the miss rate and horizontal axis shows different scenarios. As it is obvious in these figures, loss rate is much less in the proposed algorithm than the base method for all scenarios and execution times.

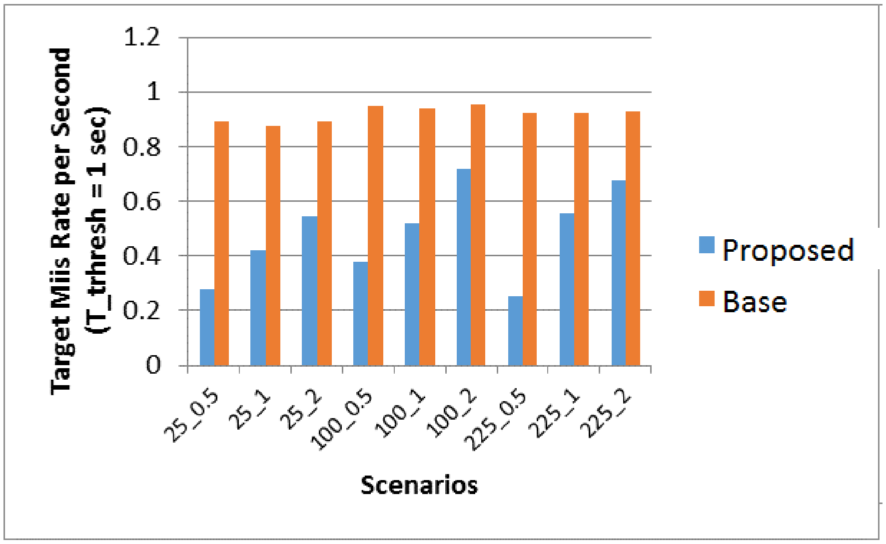


**Figure 12** Miss rate of proposed and base algorithms for threshold time of 1 second. The vertical axis shows the target miss rate per second. The horizontal axis shows the simulation scenario. On this axes the first number shows the number of nodes and the second number shows the speed of target in m/s.

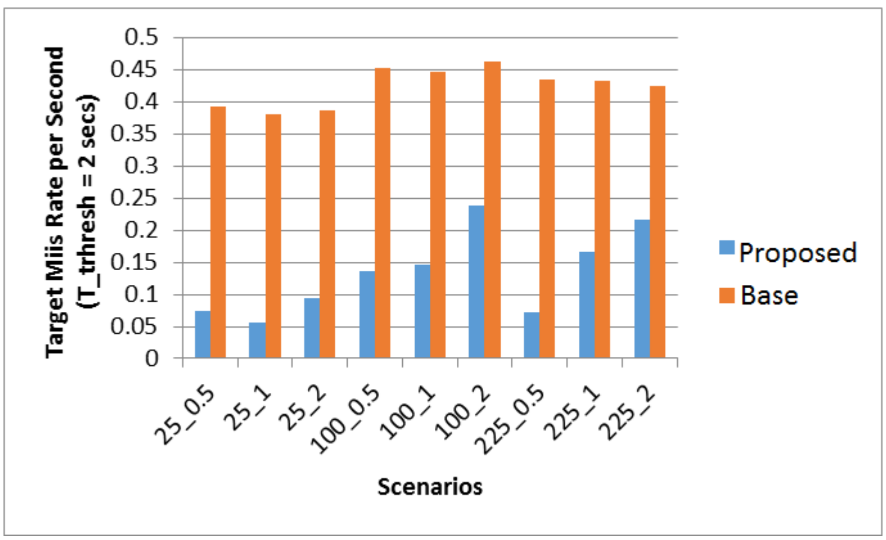


**Figure 13** Miss rate of proposed and base algorithms for threshold time of 2 seconds. The vertical axis shows the target miss rate per second. The horizontal axis shows the simulation scenario. On this axes the first number shows the number of nodes and the second number shows the speed of target in m/s.

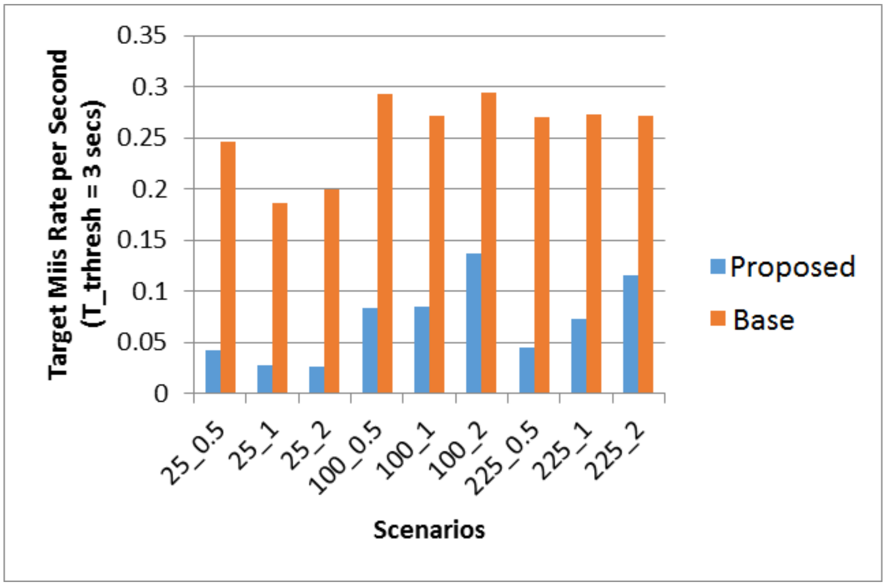


**Figure 14** Miss rate of proposed and base algorithms for threshold time of 3 seconds. The vertical axis shows the target miss rate per second. The horizontal axis shows the simulation scenario. On this axes the first number shows the number of nodes and the second number shows the speed of target in m/s.

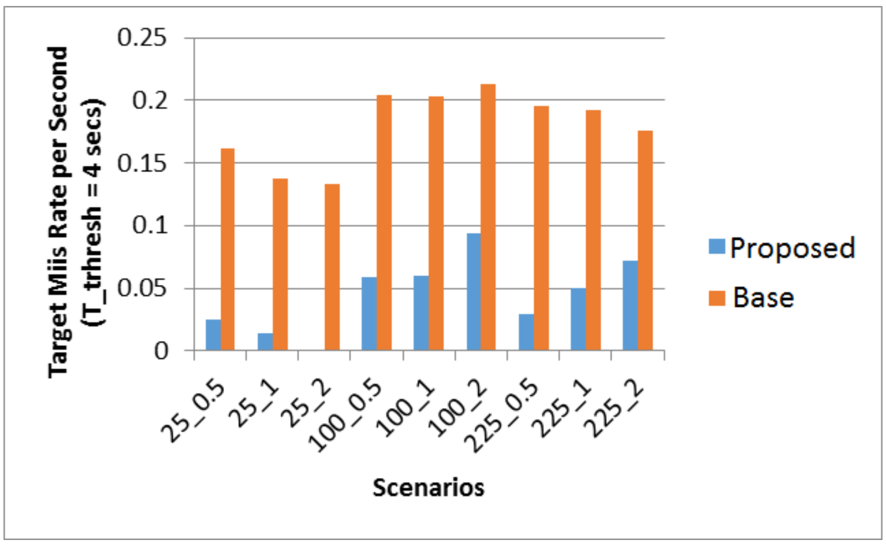


**Figure 15** Miss rate of proposed and base algorithms for threshold time of 4 seconds. The vertical axis shows the target miss rate per second. The horizontal axis shows the simulation scenario. On this axes the first number shows the number of nodes and the second number shows the speed of target in m/s.

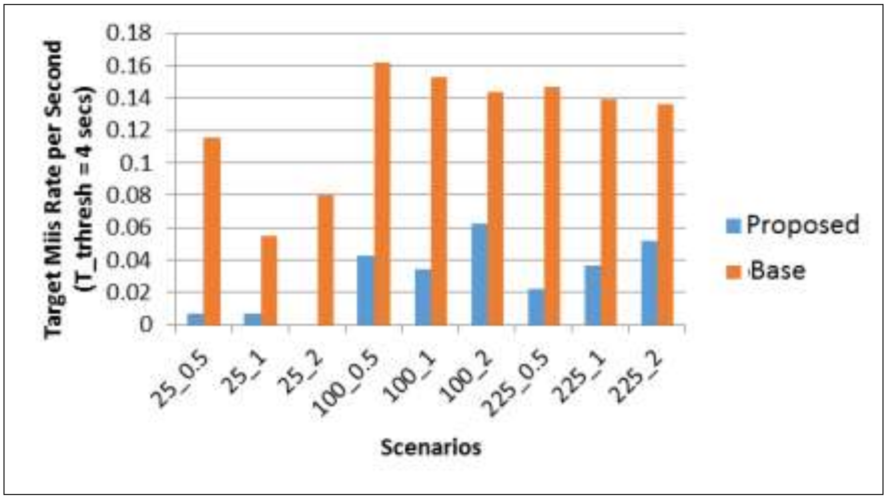


**Figure 16** Miss rate of proposed and base algorithms for threshold time of 4 seconds. The vertical axis shows the target miss rate per second. The horizontal axis shows the simulation scenario. On this axes the first number shows the number of nodes and the second number shows the speed of target in m/s.

Because zigzag path is generated randomly, it is a good representative of the real word target movement. The miss rates for zigzag scenario using base and proposed methods are shown in Figure 17. The vertical axis shows the miss rate of target. The horizontal axis shows different miss durations. Figure18 shows target miss rate per second with different miss thresholds. The horizontal axis shows different assumed miss thresholds and the vertical axis shows the miss rate calculated as described in previous section. These two figures reveal that number of misses is a little higher in proposed method but the duration of miss is much lower than base algorithm and so the proposed method overperforms the base method for zigzag simulation scenario.

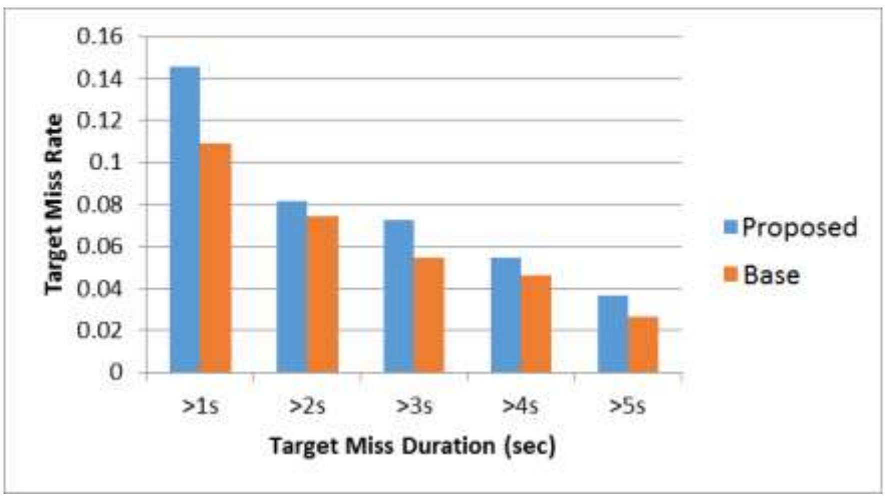


**Figure 17** Target miss rate of proposed and base algorithms. The vertical axis shows the target miss rate in zigzag scenario. The horizontal axis shows the miss duration of target in seconds.

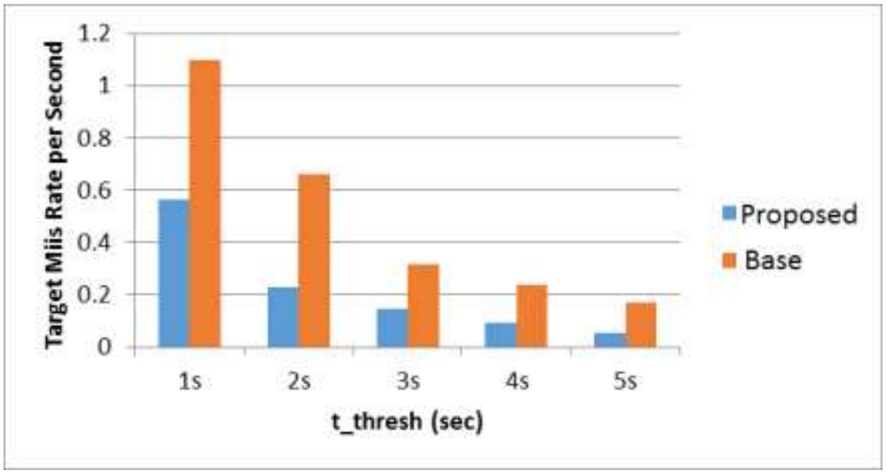


**Figure 18** Target miss rate per second for proposed and base algorithms with different t_threshs. The vertical axis shows the target miss rate per second. The horizontal axis shows the t_thresh in seconds.

**Energy consumption**

Energy consumption would become the main concern while adding algorithms and increasing calculations complexity. So we added the estimation of energy consumption of each node to the simulation.Figure19 shows energy consumption after target enters into the scenario (without network initialization) while the Figure 20 shows energy consumption during whole simulation time (including network initialization phase with clustering and ant colony algorithm initialization).These charts reveal that not only the proposed method consumes no more energy than the base method but also consumes less energy in some of the scenarios. This is caused by the harmony between proposed clustering and ant colony algorithms. Though the ant colony do many calculations in the initialization phase but the cluster heads can route the packets through the optimized path in least time and with minimum energy during rest of network execution time. Also we can consider initial ant packets to be small in order to prevent further loss of energy.2-hop clustering causes nodes to better communicate. Due to this type of clustering, nodes can send data to new agents quickly and easily when target runs out of master's or slaves' sight. In addition, restricting data transfer task only to cluster heads reduces processing overhead of other nodes and prevent redundant data transfer and waste of energy.

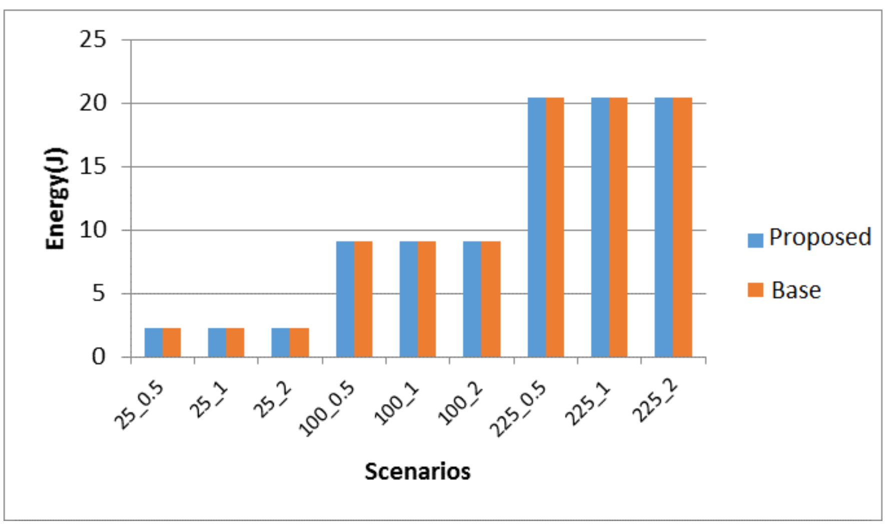


**Figure 19** Energy consumption for proposed and base algorithm after target enters into the scenario (ignoring network initialization). The vertical axis shows the energy consumption in joule. The horizontal axis shows different scenarios. In this axis the first number shows the number of nodes and the second number shows the target speed in m/s.

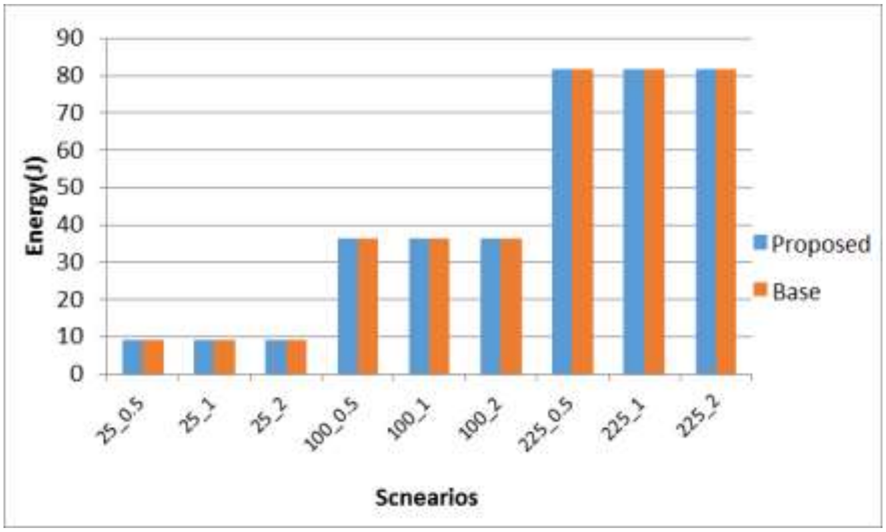


**Figure 20** Energy consumption for proposed and base algorithm during whole simulation time. The vertical axis shows the energy consumption in joule. The horizontal axis shows different scenarios. In this axis the first number shows the number of nodes and the second number shows the target speed in m/s.

**Conclusions**

The proposed method uses clustering technique which causes the network to split into some logical sub networks. Each cluster works autonomously to reach the network goal. In such a network, using smart agents results in stronger communication between nodes. Each node should assume part of the responsibility to reduce the processing overhead of cluster head. We use three nodes simultaneously to track a target in this study. The agents activate only on these three nodes. Also as an optimized and low energy solution is needed to reach the data packets to the sink, we use ant colony method which is a novel and optimized solution to find paths in distributed systems. The simulation results show that the combination of algorithms used in the proposed method reduce energy consumption in all scenarios. Also other parameters such as loss rate and target positioning error have improved in comparison to basic method. This study reveals that using agents in combination with other methods can improve the overall performance of distributed systems. Some future works can be suggested as follows:

1. Using hardware smart agents to transfer data and provide energy for sensors.
2. Using more target tracking agents to increase accuracy.
3. Combining other technologies such as RFID with sensors for obtaining information from environment and energy management.
4. Studying other novel heuristic algorithms such as PSO or bee colony.

**References**

1. Chong C-Y, Kumar SP. Sensor networks: evolution, opportunities, and challenges. Proc IEEE. 2003;91(8):1247–56.

2. YICK J, MUKHERJEE B, GHOSAL D. Wireless sensor network survey. Comput Networks [Internet]. 2008 Aug;52(12):2292–330. Available from: http://linkinghub.elsevier.com/retrieve/pii/S1389128608001254

3. Anthony A. A comparison of agent paradigms for resource management in distributed sensor networks. Vol. Ph.D. The University of Alabama at Birmingham; 2007. p. 143.

4. Babbar R. Agent based resource management in 3G wireless networks. University of Calgary; 2005.

5. Gupta R, Das SR. Tracking moving targets in a smart sensor network. In: 2003 IEEE 58th Vehicular Technology Conference VTC 2003-Fall (IEEE Cat No03CH37484) [Internet]. IEEE; 2003. p. 3035–9 Vol.5. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1286181

6. Tseng Y-C. Location Tracking in a Wireless Sensor Network by Mobile Agents and Its Data Fusion Strategies. Comput J [Internet]. 2004 Apr 1;47(4):448–60. Available from: http://comjnl.oupjournals.org/cgi/doi/10.1093/comjnl/47.4.448

7. Karlsson B, Bäckström O, Kulesza W, Axelsson L. Intelligent sensor networks-an agent-oriented approach. In: Real-World Wireless Sensor Networks (REALWSN"05). Stockholm, Sweden; 2005.

8. TAN L, ZHANG S, QI J. An Intelligent Cooperative Protocol in Clustered Wireless Sensor Networks for Energy-Efficiency. J Comput Inf Syst. 2012;8(4):1415–23.

9. Haiping HUANGCAO X, WANG R, SUN L. A Novel Clustering-based Ant Colony Routing Algorithm in Large Scale Wireless Multimedia Sensor Networks. J Comput Inf Syst. 2011;7(5):1753–60.

10. Wang G, Wang Y, Tao X. An Ant Colony Clustering Routing Algorithm for Wireless Sensor Networks. In: 2009 Third International Conference on Genetic and Evolutionary Computing [Internet]. IEEE; 2009. p. 670–3. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5402746

11. Yang J, Xu M, Zhao W, Xu B. A Multipath Routing Protocol Based on Clustering and Ant Colony Optimization for Wireless Sensor Networks. Sensors [Internet]. 2010 May 4;10(5):4521–40. Available from: http://www.mdpi.com/1424-8220/10/5/4521/

12. Castalia, Wireless Sensor Network Simulator [Internet]. 2016 [cited 2016 May 18]. Available from: https://castalia.forge.nicta.com.au/index.php/en/