

An obstacle aware mobile sink path strategy in WSN

T. H. Feiroz Khan¹, D. Siva Kumar²

¹Department of Computer Science and Engineering, Sathyabama Institute of Science & Technology, India

²Department of Electronics and Communication Engineering, Easwari Engineering College, India

Article Info

Article history:

Received Oct 11, 2018

Revised Feb 10, 2019

Accepted Mar 21, 2019

Keywords:

Cuckoo search algorithm

wireless sensor network

Mobile sink

Obstacle aware path strategy

Obstacle detection

ABSTRACT

In Wireless Sensor Networks, Mobile Sink accomplishes considerable achievement on network lifetime improvement. In sensing environment, more chances to present the obstacle. But, in the mobile sink, how to identify the obstacle and make the obstacle aware path strategy is a challenging task. To overcome this problem, we propose an Obstacle Aware Mobile sink Path Strategy (OAMPS) that detects any obstacles which enter within the network lifetime and design a shortest mobile sink movement path avoiding detected obstacles. In this scheme, the mobile sink collects the sensing data from the super node then it sends the data to the base station. Here, static or moving obstacles are present when the mobile sink moves the path scheduling by spanning graph. This algorithm is introduced to discovering the obstacle avoiding shortest path. The source selects the route by the updated cuckoo search algorithm. The simulation results show that the OAMPS improved the throughput and minimized the delay in the network.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

T. H. Feiroz Khan,

Department of Computer Science & Engineering,

Sathyabama Institute of Science & Technology,

Rajiv Gandhi Salai, Jeppiaar Nagar, Chennai, Tamil Nadu 600119.

Email: feirozkhan36@gmail.com

1. INTRODUCTION

Wireless Sensor Networks (WSNs) have enabled a broad spectrum of applications through low-cost low-power sensor nodes, e.g., habitat monitoring, precision agriculture, and forest fire detection. In these applications, the sensor network will operate under a few human interventions either because of the hostile environment or high management complexity for manual maintenance. Since sensor nodes have limited battery life, energy saving is of paramount importance in the design of sensor network protocols. Recent research on data collection reveals that, rather than reporting data through long, multi-hop, and error-prone routes to a static sink using tree or cluster network structure, allowing and leveraging sink mobility is more promising for energy efficient data gathering. Mobile Sinks (MSs), such as animals or vehicles equipped with radio devices, are sent into a field and communicate directly with sensor nodes, resulting in shorter data transmission paths and reduced energy consumption.

However, data gathering using mobile sinks introduces new challenges to sensor network applications. To better benefit from the sink's mobility, many research efforts have been focused on studying or scheduling movement patterns of a mobile sink to visit some special places in a deployed area, to minimize data gathering time. In such approaches, a mobile sink moves to pre-determine sojourn points and query each sensor node individually. Although several mobile elements scheduling (MES) protocols have been proposed to achieve efficient data collection via controlled sink mobility, determining an optimal moving trajectory for a mobile sink is itself an NP-hard problem, and may not be able to adapt to constrained access areas and changing field situations.

Early obstacle detection and avoidance scheme [1] for detecting obstacles then make an optimal path. This scheme detects the obstacles by path optimality evaluation, made at the node level, gradually over

time. Thus this scheme improves the message delivery rate and minimizes the overhead. Obstacle Resistant Robot Deployment (ORRD) algorithm [2] that manages node placement policy, a serpentine movement policy, and obstacle handling rules in WSN. This ORRD handles the unpredicted irregular obstacles in the simulation environment. The obstacle impacts by Denial of Service or congestion in a WSN. Decentralized Boundary Detection algorithm [3] for recognizing presents sensors near obstacle in the WSN. This algorithm does not need any awareness of the sensor locations among two nodes. However, this scheme requires knowledge of the three hop neighbors in the network. Obstacle Boundary Detection Algorithm [4] detects the dissimilar shaped obstacle. However, this scheme detects an only a static obstacle in the network. The prime reason for using anchors is for comparatively less involvement of sensors in its localization and obstacle discovery process considering the optimal use of its limited energy resources.

Goa Goal-Seeking Obstacle and Collision Evasion scheme (GOCE) [5] is used to detect the obstacle and avoid the network collision in the network. In obstacle evasion, the nodes and packets exchange state information to communicate the existence of bottlenecks. This scheme reduces both network delay and bandwidth consumption. However, during high congestion, the latency is increased. The MS is used to decrease the energy utilization of nodes, and it prevents the creation of energy holes in WSN. These are all depending upon the path which is taken, especially on delay-sensitive applications. The proposed method has overcome this problem by forming a hybrid moving pattern, in that Rendezvous Points (RPs) are only visited by mobile sink nodes. The primary difficulty then becomes compute a tour that visits all these RPs in a given delay bound. Weighted Rendezvous Planning is proposed, whereby every sensor node is assigned a weight after its hop distance from the tour as well as the amount of data packets that it onwards to the neighbouring RP. The efficiency can be improved by taking account of data with dissimilar delay needs [6].

Decentralized Boundary Detection (DBD) algorithm [7] is proposed for identifying the sensor nodes close to a hole or obstruction in the WSN. This algorithm does not need any information of the node location or space among two nodes. The main advantage of using this algorithm is to provide location-free. The outcomes effectively detect all boundaries in the WSNs with the lesser control overhead. Obstacle Avoidance in Mobile Robotic Sensors [8] investigates on the importance of connectivity and restoration on mobile robotic sensors that work under harsh environments. The existing systems avoid the problem of obstacle avoidance. The paper suggests an effective methodology named, online connectivity restoration strategy, which withstands under real-world working conditions. Autonomic Obstacle Detection and Avoidance [9] investigate instead the dynamic and autonomic detection of obstacles that might stand within the network. This is accomplished using the enhanced cartography optimized link state routing with no extra signalling overhead.

Geographic Routing in Clustered WSNs among Obstacles [10] proposed a clustering technique in WSNs named energy-efficient homogeneous clustering that periodically selects the cluster heads according to a hybrid of their residual energy and a secondary parameter, such as the utility of the sensor to its neighbours. In this way, the selected cluster heads have an equal number of neighbours and residual energy. Virtual Coordinate-Based Bypassing Void Routing protocol [11] is to transform a random structure composed of void edges into a regular one by mapping edge nodes coordinates to a virtual circle. By utilizing the virtual circle, the greedy forwarding can be prevented from failing, so that there is no routing void in the forwarding process from source to destination and control overhead can be reduced. Furthermore, the virtual circle is beneficial to reduce the average length of routing paths and decrease transmission delay. However, this method increases the load when large voids in the network. An obstacle-aware path loss model [12] is used to reflect the effect of the obstacles on the communication among sensors. It maintaining an acceptable energy consumption and packet delivery ratio in the network. An obstacle aware connectivity restoration strategy [13] is designed for optimizing the obstacle-avoiding minimum spanning tree of the segments and determine the relays. K vertex-disjoint shortest paths in WSN [14] that procedures counting paths to diminish the relay nodes count. Swarm intelligence optimization technique [15] for obstacle-avoidance path planning in WSNs. This planning is considered by two swarm intelligence algorithms, that is grey wolf optimizer and whale optimization algorithm.

2. AN OBSTACLE AWARE MOBILE SINK PATH STRATEGY

In recent years, WSNs are employing collecting information from various environments via sensor nodes. To protect the sensor node from the energy hole problem, the MS is used. This MS moves the sensing environment, and it is gathering information through sensor nodes. However, the obstacle present the sensing environment the MS is can't move regularly. To solve this problem, we propose An Obstacle Aware Mobile Sink Path Strategy in WSN.

Figure 1 demonstrates the architecture of the proposed scheme. The base station is an owner of a WSN. Here, a number of sensor nodes are disseminated over where some obstacles exist. Also presents some

Super node. It has additional communication range and energy. It collects the information from the sensor node, and it sends the information to the MS. Initially, the MS collects the super node and sensor nodes information from the base station. Then the MS start data gathering process; the MS sends RREQ message to the Super node. The super node broadcast this message to within communication range sensor nodes. Then the sensor nodes send the updated information to MS via the super node.

Any sensor node communicates the sensing information to the base station through the super node. Suppose, the sensor node or super node cannot be received or transmit which information at a time the super decides possible to present in the obstacle. Then the super tracked the obstacle location and identified the obstacle is static or movable. If the obstacle is static, the MS path is predetermined. The detected obstacle is movable when finding out the optimal path.

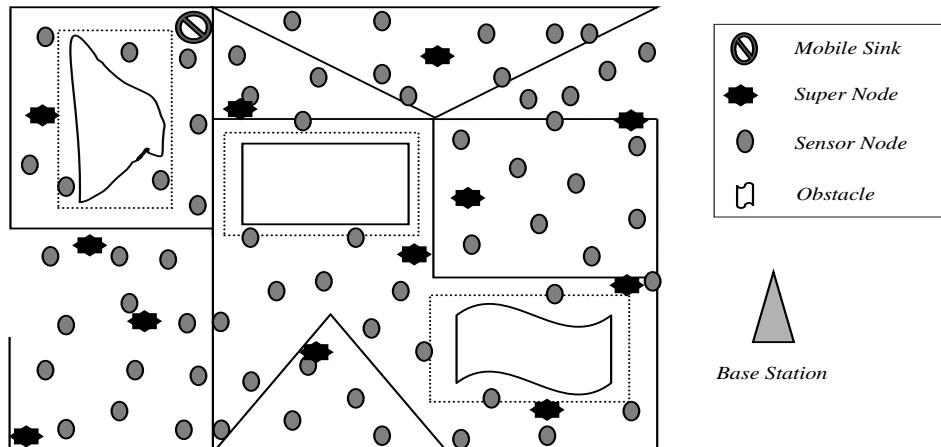


Figure 1. Architecture of Obstacle Aware Scheme

2.1. Spanning Graph Algorithm

In this scheme, we use the minimum spanning tree to discover an obstacle-avoiding shortest route for the mobile sink. The obstacle-avoiding spanning graph is the set of edges that can be designed by making associations between sensors and obstacle bends. Once a spanning graph is assembled, the unbounded possible sites for the MS movement will be minimized to a limited set of sites. Therefore, the algorithm based on the spanning graph makes it more efficient to schedule for the MS.

The obstacle-avoiding spanning graph is the set of edges that can be formed by making connections between terminals and obstacle bends. Once a spanning graph is constructed, the infinite possible sites for the mobile sink movement will be reduced to a finite set of sites.

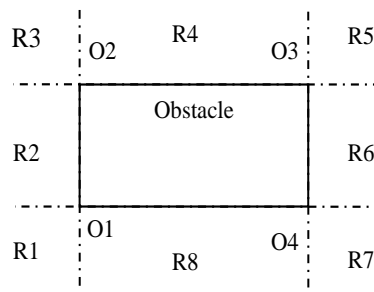


Figure 2. Sensor Vertices with obstacle bends

Figure 2 explains the sensor vertices and separation for an obstacle corner correspondingly. In this figure, the region R2 and R6 obstacle corners are joined to the spanning graph. Here, sort elements of the list V by non-decreasing x coordinates.

Similarly, the region R4 and R8 also obstacle corners but, here sort elements of the list V by non-decreasing y coordinates in the region R4 and R8 of all obstacle corners. Besides the region R1 and R5 of obstacle corners, the region R1 and R3 of join vertices. We sort elements of the list V by non-decreasing x + y coordinates in the region R1 and R5 of obstacle corners and the region R1 and R3 of joining vertices. Similarly, the region R3 and R7 of all obstacle corners and the region R2 and R4 of join vertices. Here, the elements of the list V are sorted by non-decreasing y-x coordinates. According to the quadrant partition for an obstacle corner and a pinned vertex, we finally construct the spanning graph. As a result, the MS movement distance can be minimized. In WSN, the environment consists of obstacles with different shapes owing to the irregular shape of obstacles. Thus, the spanning graph algorithm to discover an obstacle-avoiding shortest route minimized MS movement strategy.

2.2. Static Obstacle

In a static obstacle protesting situation, obstacles locations are permanent when the MS sees every obstacle boundary by travelling salesman problem. The elaborated process of MS motion method in static obstacle protesting surrounding is explained below: if one obstacle exists within the area, then super node informs MS earlier than visiting the area. While the MS initiates data gathering within the obstacle area, the MS make minimum spanning tree to discover an obstacle-avoiding shortest route. In obstacle protesting surroundings, MS chooses exacting data gathering points within the area. Super nodes are selects the data gathering points thus sensor to communicate data to MS with least energy loss. Then MS visits super node points within the area next every sensor node communicate data to MS which stopped via any obstacle.

2.3. Moving Obstacle

In moving obstacle protesting surroundings, obstacle exchanges its location with regard to the time period. As a result, MS also changes its motion pathway with regard to obstacle motion policy. The elaborated process of MS motion method in moving obstacle protesting surrounding is explained below:

If obstacle goes from one area to another area with MS within the similar super node, then MS overcome the obstacle if this obstacle is stopped the MS motion pathway. Then, the super node informs obstacle entropy with a definite time by the border sensor nodes of the obstacle. While super node obtains update information for MS motion, then MS exchanges its motion pathway if obstacle stops the MS motions. If obstacle goes with the MS and attains to next visiting area with MS, then super node does not assign this area for MS travel. In that situation, super nodes assign different nearby non-visiting area after that MS detects spanning graph algorithm to overcome the discovered obstacle.

2.4. Route Formation

After detecting the obstacle, and then the source node form the route based on the Cuckoo Search (CS) Algorithm. CS is a heuristic search algorithm that cuckoos select nests (neighbor nodes) of another bird to place their eggs (nodes). Providentially, the other bird easily recognizes that the eggs are not its personal and may be destructs the cuckoo egg. Therefore, the cuckoo originated evolutionally generates eggs that appear related to the host birds. Thus, three significant stages lead an optimization procedure, which is as follows:

- a) Initially, the egg of cuckoo acts as a set of solutions that are randomly positioned at different nests.
- b) Next, the best eggs with suitable solutions will be permitted to shift into the next step.
- c) Finally, one of the eggs is recognized as extraordinary, then that solution will be removed, and the latest egg will restore this strange in a new nest.

The process of the CS is explained:

Initially Pop host nests, a rest of population.
$$P = \sum_{i=1}^{pop} p_i$$

Calculate fitness of nests based on node energy
$$Fit(nest) = \sum_{i=1}^{pop} f(p_i)$$

Do Iterations < Maximum Iterations {

Make a cuckoo egg P'_i of a random nest

Calculate the fitness $f(p'_i)$

Check present egg: select another a random nest j

If $f(p'_i) < f(p_j)$ {

$P_j = p'_i$

}

Construct fresh nests randomly at new positions

}

The original CS is straightforward, and it has less parameter to adjust therefore there is a large possibility for local convergence getting fewer quality results. Thus we update the CS technique by the different operator that randomly updates the values at one or more investigate places of the elected eggs.

Along with this process, for every egg $P_i^t = \{p_1, p_2, \dots, p_n\}$, $P_i^{t+1} = \{p'_1, p'_2, \dots, p'_n\}$ in the population of t^{th} iteration, a new egg $P_i^{t+1} = \{p'_1, p'_2, \dots, p'_n\}$ is formed as:

$$P'_k = \begin{cases} p_k + \Delta(I, p_A - p_k), & \text{if a rand}(0,1) < 0.5 \\ p_k + \Delta(I, p_k - p_R), & \text{if a rand}(0,1) > 0.5 \end{cases} \tag{1}$$

Where P_A and P_R are the acceptance possibility and rejecting possibility rates of the variables p_i . The amount $\Delta(I,y)$ Generates areal number in the range $[0, p_i]$ i.e, $\Delta(I,p_i)$ approaches zero as the count of iterations raises. The above formula builds the fresh eggs to investigate the space equally through the start of the run while iteration quantity is small and adjust the search space as the iteration number movement. This process raises the chances of making fresh eggs close to its best-lived eggs. The quantity $\Delta(I,y)$ is computed by the subsequent function:

$$\Delta(I, y) = y[1 - \gamma(1 - t/T)p] \tag{2}$$

Where γ = random number among $[0, 1]$,

T = maximum number of iterations,

p = system parameter that determines the degree of dependency.

Therefore, UCS will produce the fresh nests and the fresh egg should accomplish all the restraints. This is understood by taking into account the following criterion and is explained as follows:

- a) If the egg structure is negative or zero, then substitute the fresh egg's complete value by a recently yielded random number which fulfills the limitation.
- b) If new egg structure is higher than one, then replace the new egg structure by one. Then, the new nest is formed; it is measured by the fitness function.

3. SIMULATION SETTINGS

In this section, we apply An Obstacle Aware Mobile Sink Path Strategy for the sensing field and evaluate the performance of the algorithm by using network simulator. In this experiment, we assume that 50 sensor nodes are randomly distributed over a 500x500m field. Simulation settings and Evaluation Results are as follows.

3.1. Packet Received Rate

Packet Received Rate (PRR) is defined as the rate of packets delivered to the destination node. PDR is measured by (3). Where, n number of sensor nodes.

$$PDR = \sum_0^n \frac{PacketsDelv}{Time} \tag{3}$$

The total numbers of packets received in the sensing field of the OAMPS plotted against those of the existing protocol GOCE in Figure 3 respectively. Here, the red color curve represents the packet received rate of GOCE and green color curve represents the OAMPS. The checking of the messages for every data transmission in GOCE causes greater overhead and packet loss. Therefore OAMPS performs better than the GOCE protocol.

3.2. Packet Loss Rate (PLR)

Packet Loss Rate is defined as the number of packets lost per unit time. PLR is measured by (4).

$$PLR = \sum_0^n \frac{PacketsLost}{Time} \tag{4}$$

The total numbers of packets lost of the OAMPS are plotted against the corresponding GOCE in Figure 4 respectively. The reason for the packet loss is the same as that of the packet delivery in both the cases of WSNs since packet loss only contradicts the packet delivery. The number of packets received by the existing methods is lesser due to the energy consumption and delay. But the OAMPS method is to reduce the packet loss compared to the GOCE.

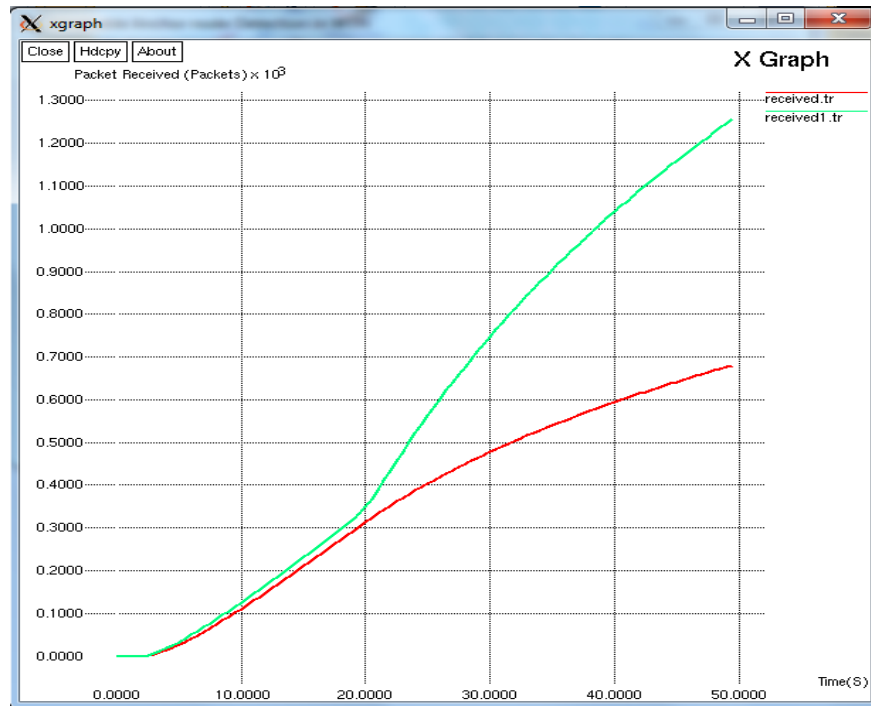


Figure 3. Packet Received rate of OAMPS and GOCE

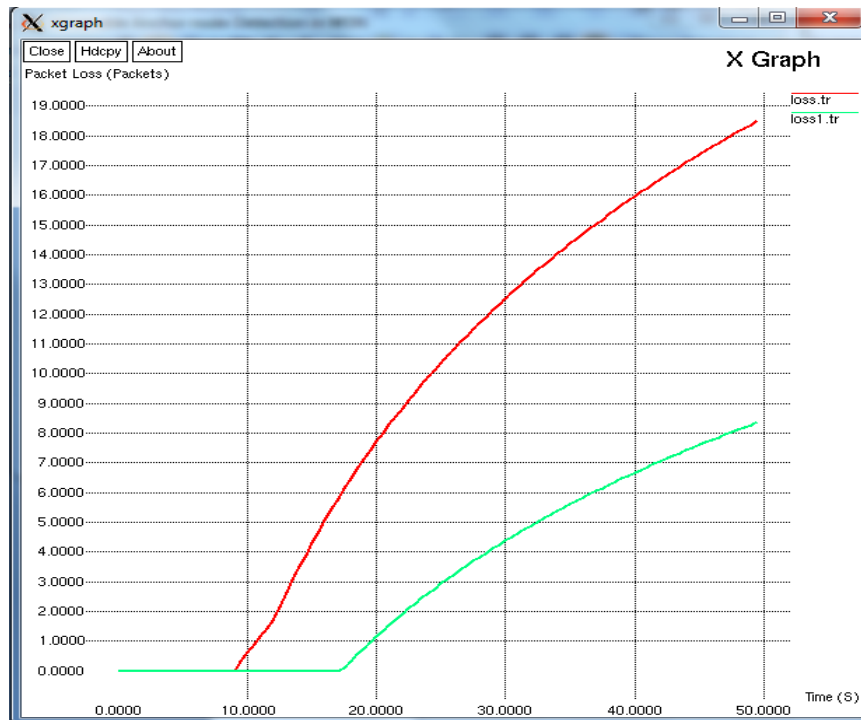


Figure 4. Packet Loss rate of OAMPS and GOCE

3.3. Average Delay

The difference of time among the current packets received and the previous packet received is called Average Delay. It is measured by the (5). Figure 5 indicates that the range of delay is small for the proposed method OAMPS than the existing method GOCE. The delay in the network may lead to losing the information.

$$AD = \frac{\sum_0^n Pkt\ Recvd\ Time - Pkt\ Sent\ Time}{n} \tag{5}$$

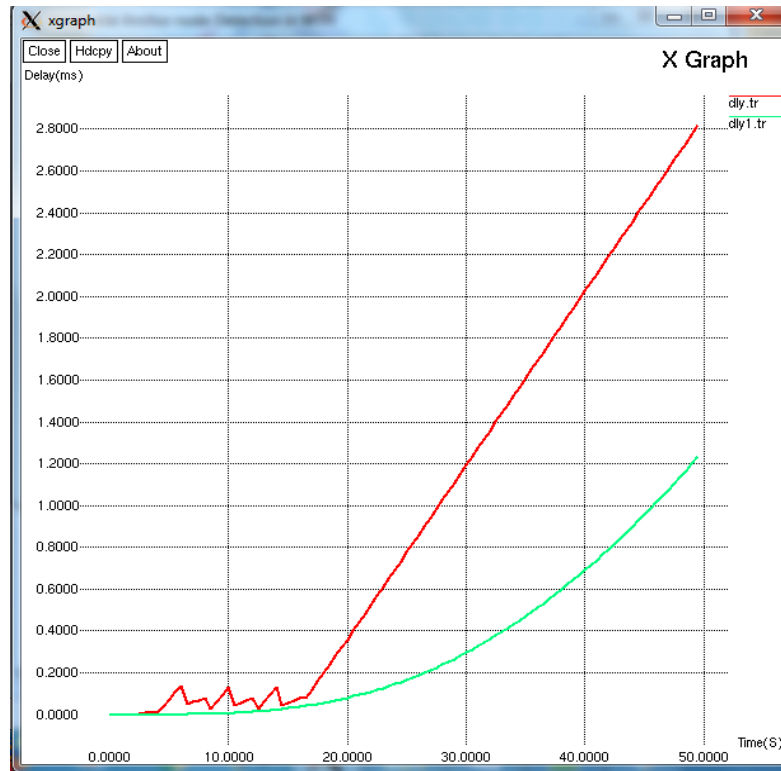


Figure 5. Average Delay of OAMPS and GOCE

3.4. Throughput

Throughput is defined as the average packets delivered to the destination successfully. The throughput is estimated using (6).

$$Throughput = \frac{\sum_0^n Pkts\ Received\ (n) * Pkt\ Size * 8}{1000} \tag{6}$$

Throughput is measured in terms of bytes, and hence the obtained value is multiplied with 8, so that obtained value is converted into bits. Then the value is again divided by 1000 for the conversion process from bits into kilobits. Figure 6 indicates that the proposed protocol OAMPS has better throughput when compared to the existing protocol GOCE.

3.5. Residual Energy

The quantity of remaining energy in a sensor node at the present case of time is called as residual energy. In the network operations, the rate of consumed energy is calculated by the residual energy. The residual energy is estimated using (7).

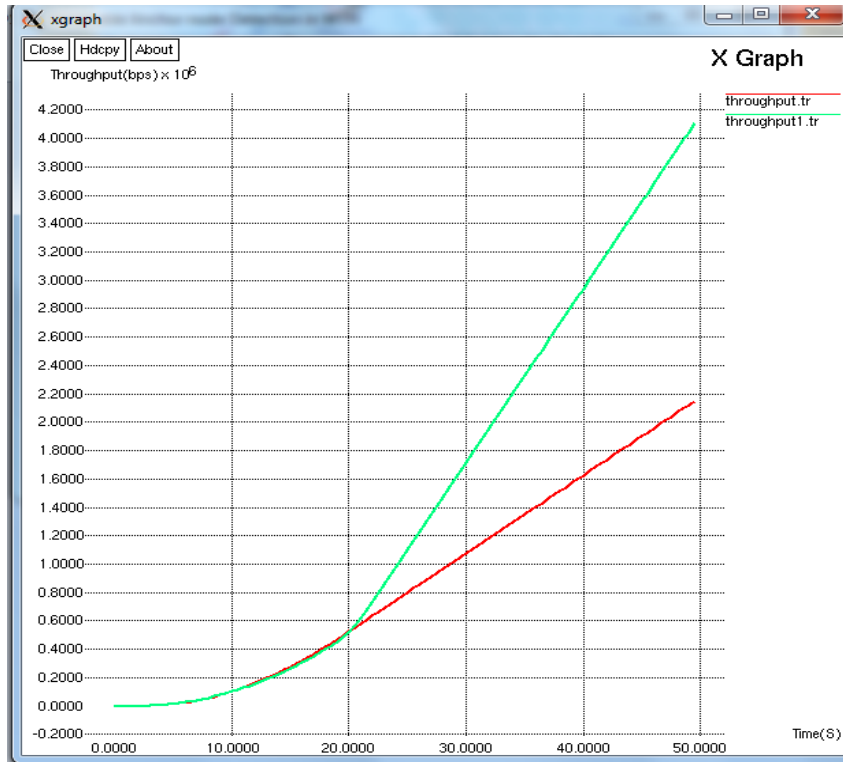


Figure 6. Throughput of OAMPS and GOCE

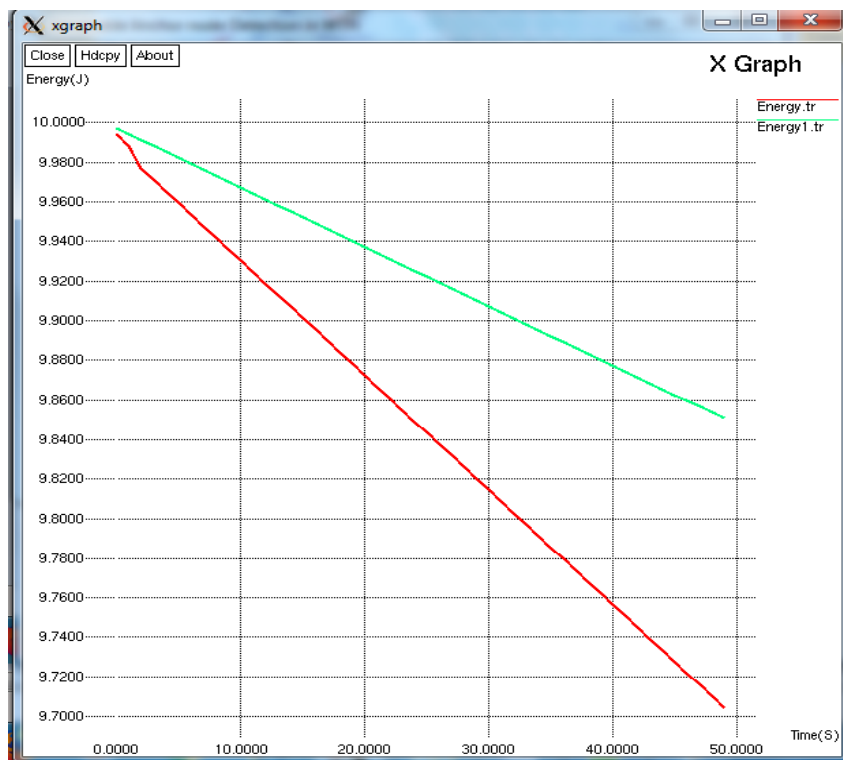


Figure 7. Residual Energy of OAMPS and GOCE

$$ResEn = InitialEnergy - CurrEnergy$$

(7)

Figure 7 indicates that in the network the residual energy is enhanced for the proposed protocol OAMPS when compared with the existing method GOCE. In GOCE, reduce the sensor node will be dead also as long as there is residual energy, the network would have an enhanced network lifetime in the network.

4. CONCLUSION

In this scheme, we build the spanning graph for scheduling the mobile sink path to discover the obstacle-avoiding shortest route. This scheme makes the obstacle avoiding path scheduling during static and moving obstacle. Here, we are using the mobile sink and super nodes to improving the network lifetime. The super node is acting as a mobile sink data gathering point thus minimizes the sensor energy expenditure. The source selects the next hop by updated cuckoo search algorithm hence reduce the network delay. In this scheme, the simulation results are done by network simulator-2, and results demonstrate that improve the throughput and reduce the network delay.

REFERENCES

- [1] L. Moraru, *et al.*, "Geographic Routing with Early Obstacles Detection and Avoidance in Dense Wireless Sensor Networks," *In International Conference on Ad-Hoc Networks and Wireless*, Springer, Berlin, Heidelberg, pp. 148-161, 2008.
- [2] C. Y. Chang, *et al.*, "Obstacle-Resistant Deployment Algorithms for Wireless Sensor Networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 6, pp. 2925-2941, 2009.
- [3] W. C. Chu, and K. F. Ssu, "Decentralized Boundary Detection Without Location Information in Wireless Sensor Networks," *In Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pp. 1720-1724, 2012.
- [4] S. Das, *et al.*, "Sensor Localization and Obstacle Boundary Detection Algorithm in WSN," *In Advances in Computing and Communications (ICACC), 2013 Third International Conference on IEEE*, pp. 412-415, 2013.
- [5] J. Wei and D. Kundur, "Goalie: Goal-Seeking Obstacle and Collision Evasion for Resilient Multicast Routing in Smart Grid," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 567-579, 2016.
- [6] H. Salarian, *et al.*, "An Energy-Efficient Mobile-Sink Path Selection Strategy for Wireless Sensor Networks," *IEEE Transactions on vehicular technology*, vol. 63, no. 5, pp. 2407-2419, 2014.
- [7] W. C. Chu, and K. F. Ssu, "Decentralized Boundary Detection without Location Information in Wireless Sensor Networks," *In Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pp. 1720-1724, 2012.
- [8] S. A. Sebi, and D. Sunny, "Obstacle Avoidance in Mobile Robotic Sensors and Establishing Connection," *Procedia Technology*, vol. 25, pp. 364-371.
- [9] A. Belghith, *et al.*, "Autonomic Obstacle Detection and Avoidance in MANETs Driven by Cartography Enhanced OLSR," *Mobile Information Systems*, 2015.
- [10] H. P. Gupta, *et al.*, "Geographic Routing in Clustered Wireless Sensor Networks Among Obstacles," *IEEE Sensors Journal*, vol. 15, no. 5, pp. 2984-2992.
- [11] D. Zhang, and E. Dong, "A Virtual Coordinate-Based Bypassing Void Routing for Wireless Sensor Networks," *IEEE sensors journal*, vol. 15, no. 7, pp. 3853-3862, 2015.
- [12] R. Elhabyan, W. Shi, and M. St-Hilaire, M. "An Obstacle-Aware Clustering Protocol for Wireless Sensor Networks with Irregular Terrain", *International Conference on Wired/Wireless Internet Communication*, Springer, Cham, pp. 54-66. 2018.
- [13] G. Ma, *et al.*, "Obstacle aware connectivity restoration for disjoint wireless sensor networks using a mix of stationary and mobile nodes", *International Journal of Distributed Sensor Networks*, vol.12,no.5, pp.6469341,2016.
- [14] L. Sitanayah, *et al.*, "A fault-tolerant relay placement algorithm for ensuring k vertex-disjoint shortest paths in wireless sensor networks", *Ad Hoc Networks*, vol.23, pp.145-162,2014.
- [15] A. Alomari, *et al.*, "Swarm intelligence optimization techniques for obstacle-avoidance mobility-assisted localization in wireless sensor networks", *IEEE Access*, vol.6, pp.22368-22385,2018.