Maximizing Lifetime of Barrier Coverage Wireless Sensor Network with Tell Neighbor Algorithm

Vinay Chowdary[a], Mukul Kumar Gupta[b], Sushabhan Choudhury[c]
Department of Electrical and Electronics
School of Engineering
University of Petroleum and Energy Studies, Dehradun, India
E-mails: [a]vchowdary@ddn.upes.ac.in, [b]mkgupta@ddn.upes.ac.in, [c]schoudhury@ddn.upes.ac.in

Abstract
The maximizing lifetime of Barrier coverage in wireless sensor network has attracted researchers in recent past. In barrier coverage applications it is essential to prolong the lifetime of sensor networks which guarantees extended operating lifetime of the overall network. In this paper, we propose a novel *Tell Neighbor Algorithm* (*TNA*), which increases the operating lifetime of the overall network. Simulation results show that the lifetime obtained by *TNA* is better than some of the already existing state-of-art algorithms. Lifetime obtained by *TNA* is doubled when compared already existing algorithms.

Keywords- Barrier coverage, Lifetime of network, TNA algorithm, Sensor node.

1. Introduction

Coverage in Wireless Sensor Networks (WSNs) has attracted researchers in recent past. One of the widely accepted metric for measurement of Quality of Service in WSN is coverage. In general three types of coverage problems exist in WSN, viz, Point coverage, Area coverage and Barrier coverage. Area coverage/full coverage requires sensors to be deployed throughout the area of interest which results in more number of sensors for deployment. Deployed nodes in area coverage need to monitor the complete length and width of the area. Full coverage as given in Figure 1, where every point in the region of interest is covered has a disadvantage in terms of a number of sensors required. Barrier coverage, as given in Figure 2, is a coverage model where the monitoring parameter is confined to movement detection in the area of interest. Sensor nodes *s* and *t* are virtual nodes added to the left and right boundaries of the area respectively.
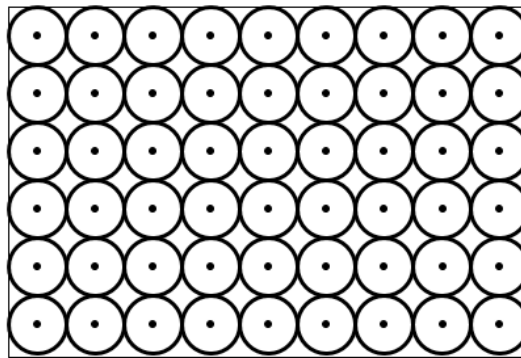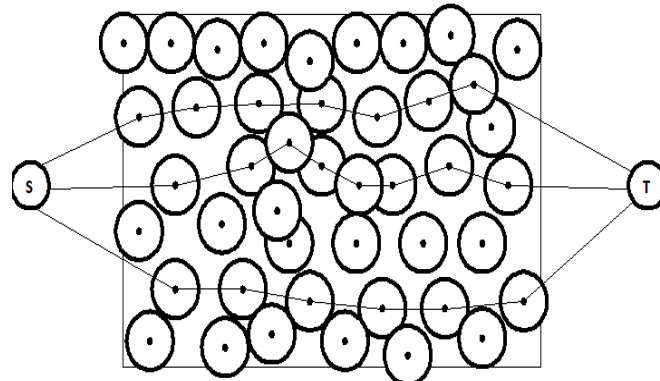

Figure 1. Full coverage

Figure 2. Barrier coverage

This guarantees that paths that are formed traverses from left to right of the deployed area. Each path formed is one barrier. Barrier coverage also referred as *k*-barrier coverage, where *k* is a number, in comparison, requires fewer sensors and has the advantage of reducing unnecessary coverage of every point in the field. Lifetime of sensors involved in barrier coverage can be considerably increased compared to sensors in full coverage as scheduling technique can be implemented. Scheduling the on-off time (Kumar, et al. 2005) of sensors involved in barrier formation greatly increases the lifetime of overall network in barrier coverage, whereas the same is not possible in full coverage. The objective of this paper is thus to increase the lifetime of the network by implementing barrier coverage. The overall lifetime achieved by our algorithm is more than the already existing algorithms in the literature.

The main contributions of this paper are as follows:
 (i)   Proposing a novel *Tell Neighbor Algorithm (TNA)* that maximizes the network lifetime for barrier coverage.
 (ii) Evaluating the performance of the proposed algorithm with the already state-of-art works from literature.
 (iii) To prove the effectiveness of our work by carrying extensive simulations.

2. Related Work
This section briefly reviews the work in the area of enhancing the lifetime of the network in barrier coverage. In Mostafaei (2015), the authors introduced a distributed learning automata-based algorithm to illuminate stochastic barrier coverage scope in WSNs and proposed an optimal algorithm to discover the shortest path in the edge-weighted graph of barrier coverage. He et al. (2014) conceived a curve-based deployment technique for sensor nodes. They have given a deployment strategy that finds a number of required nodes to form barrier coverage. Kim (2011) has given a scheduling approach that utilizes the approach of sleep/wakeup to increase the lifetime of sensors that are randomly deployed in the network. Using this sleep/wakeup scheduling authors of (Kumar, et al. 2005) has proposed an approach named as Randomized Independent Scheduling (RIS) where a node chooses, based on predefined probability, to be in sleep or active state. Researchers of the article Chen et al. (2007) proposed a localized coverage protocol for barrier formation which can only work for the specific network.

Barrier coverage with mobile sensors in another approach, where sensors nodes are moved after initial deployment. Ban et al. (2010) used linear integer programming approach to relocate sensors after initial random deployment to achieve *k*-barrier coverage. They proved that this problem is NP-hard and also showed that the value of *k* can be maximized while minimizing the energy consumption. Saipulla et al. (2010) studied an efficient method of using mobile sensors to improve the coverage with limited mobility of deployed sensors. Wang et al. (2014) studied problems under two scenarios, first, only when stationary nodes are deployed and second when both stationary and mobile nodes have been deployed. In both cases, they showed how to increase barrier coverage.

Zhang et al. (2009) proposed disjoint path algorithm (DPath) using linear integer programming. They showed that their results are close to optimal and better than a simple greedy algorithm. Two round maximum flow algorithm (TMFA) proposed by authors Han et al. (2016) uses directional properties of sensor forming the barriers. Their approach is towards directional sensor networks. Mostafaei et al. (2017) proposes an imperialist competitive algorithm for barrier coverage (ICABC). ICA is a computational method mainly used in optimization problems. We compare the results of our algorithm with that of DPath, TMFA and ICABC and show that the overall network lifetime achieved by our algorithm is better. Utilizing the information of redundant sensors, Du et al. (2013) showed how to increase the lifetime of the network while achieving *k*-barrier coverage. One of the main drawbacks of the technique proposed by the authors is that it works only for sensors that are deployed along a straight line. Huang et al. (2010) used a probability based technique for barrier coverage with fault detection. This algorithm selects sensors based on the probability distribution of random deployment and information on fault distribution.

TNA algorithm proposed by the authors of this article uses a novel technique where nodes are made to exchange minimum information that is required for barrier formation. Our results show that *TNA* algorithms give a better operating lifetime of the network compared to *ICABC, TMFA* and *DPath* algorithms.

3. Parameters and Definitions

Authors of this article define simulation environment (a two-dimensional space) by defining the *L* and *W* as the maximum X and maximum Y coordinates of the area. In this space, a total of N nodes will be scattered randomly where every node randomly selects coordinates while making sure that no two nodes get the same coordinates. Table 1 below gives the other parameters used in the article.

Table 1. Parameters and description

Parameter	Description of the Parameter
X_{offset} & Y_{offset}	Offset coordinates so that the nodes are not drawn too close to the edges of the monitoring area.
N	The number of nodes in the simulation environment.
R_s	The range of each node.
X_{max}	Max x coordinate, equivalent to Length L of network.
Y_{max}	Max y coordinate, equivalent to Width W of network.
$Numpaths$	The variable that represents the number of barriers found.
S	Source node.
t	Sink node.
$No\ overlap$	To check if there is any overlap in the range of two sensors.
$Scattering$	Keep it true for random deployment and false for sequential deployment.
$Numhave$	It represents the number of nodes the current node is having information about.
$current$	Node under consideration.
n	The node which is passed to the current node.

We consider a rectangular region of length L and width W. A total of N nodes are randomly scattered across the complete area of $L \times W$. Let us consider a set $V = \{v_1, v_2, \ldots \ldots \ldots v_n\}$ as set of N sensors nodes that are randomly deployed. Consider $E = \{p_1, p_2, \ldots \ldots \ldots p_n\}$ as the number edges t ofhat can be formed after initial random deployment of sensor nodes. Here the value of E corresponds to a number of barriers that are formed in the network i.e., the value of k in k- barrier covered network. Now we can construct a connectivity graph $G = \{V, E\}$ where V is the sensor node set and E is the edge disjoint set. If we consider R_c as the communication range of each sensor then we can a relation between V and E as follows:

Lemma 1: $E = \{V_i, V_j\}$ represents a path or a barrier from node V_i to V_j where $i{\neq}j$, if and only if $d(V_i, V_j) \leq R_c$, where $d(V_i, V_j)$ is the straight line distance between V_i and V_j.

If we add two virtual nodes *s* and *t* as described in Figure 2, then the set V gets modified to $\{s, v_1, v_2, \ldots \ldots \ldots v_n, t\}$ and lemma 1 has to be re-written as follows:

Lemma 2: $E = \{S, V_i\}$ represents a path or a barrier from virtual node S to V_i if and only if $d(S, V_i) \leq R_s$, where $d(S, V_i)$ is the straight line distance between S and V_i and R_s is the communication range of virtual node *S*.

Similarly, $E = \{V_j, t\}$ represents a path or a barrier from the virtual node V_j to t if and only if $d(V_j, t) \leq R_t$, where $d(V_j, t)$ is the straight line distance between V_j and t and R_t is the communication range of virtual node *t*.

Lemma 3: Using Lemma 1 and 2 we can write equation 1 which states the requirement of formation of one barrier path in the network.

$$E= \{(S,V_i) \wedge d(S,V_i) \leq R_s \} \cup \{V_i,V_j \wedge d(V_i,V_j) \leq R_c\} \cup \{(V_j,t) \wedge d(V_j,t) \leq R_t\} \forall \ V$$

$$(1)$$

In general $R_s = R_t$.

More the number of paths satisfying the equation 1 more is the number of barriers formed.

After the initial random deployment, our main aim is to form barrier coverage with minimum energy requirement. *Tell Neighbor Algorithm* (*TNA*) uses the properties of adjacency matrix created by the nodes to exchange minimum information among the neighboring nodes. This ensures that the overall lifetime of the network is enhanced.

4. Tell Neighbor Algorithm (TNA)

Once the nodes are randomly deployed in the monitoring area, after the initial deployment, every node starts searching for their neighbors. The scattering function given in table 1 is kept true for random deployment and false for deterministic sequential deployment. The philosophy of algorithm is the random deployment strategy for the nodes where every node chooses a random coordinate in the monitoring area. Every node will have assigned a unique id and a variable called numhave. Id of every node is for its own identification and variable numhave for information of neighbors. If any node comes in its communication range, it treats it as its neighbors and updates its numhave value. Therefore numhave is assigned an incremental value which indicates the actual number of neighbors associated with current node. In this algorithm node under consideration is designated as variable '*current*' and neighbor node as variable *'n'*. The *TNA* algorithm has 3 steps which are given below:

Step1- Set the node coordinates: This step has a function that allows random deployment of nodes. For this the function named *Scattering* is kept true. Once this function is called *N* nodes are scattered randomly with some constraints. As nodes are scattered randomly, two constraints should be addressed which are

i)   There is every chance that one or more nodes may get deployed outside the monitoring area. To answer this problem the algorithm has X_{max} and Y_{max}. These coordinates are the maximum limits that any node can have while randomly scattered X_{max} and Y_{max}. This means that (x,y) coordinates of each nodes will be in the range of 1 to X_{max} and 1 to Y_{max}. The same is given in the equation 2 and 3 below:

$$v_n(x_i) \begin{cases} x & 1 \leq i < X_{max} \\ Null & Otherwise \end{cases} \forall \ V \tag{2}$$

$$v_n(y_i) \begin{cases} y & 1 \leq i < Y_{max} \\ Null & Otherwise \end{cases} \forall \ V \tag{3}$$

where $X_{max} < L$ *and* $Y_{max} < W$.

ii) Nodes should not fall on the edge of the monitoring area neither they should be deployed close to the edges. Parameters X_{offset} & Y_{offset} solves this problem. The same is given in equation 4 and 5.

$$v_n(x_i) \begin{cases} x & 1 \leq i < X_{offset} \\ Null & Otherwise \end{cases} \forall V \tag{4}$$

$$v_n(y_i) \begin{cases} y & 1 \leq i < Y_{offset} \\ Null & Otherwise \end{cases} \forall V \tag{5}$$

where $X_{offset} < X_{max}$ and $Y_{offset} < Y_{max}$.

Step-2: Find Neighbor: After successful completion of step1, every node tries to find its neighbors. This is done by sending a *'hello'* packet to all the nodes that are in its range. Two nodes V_i and V_j exchange packets only if the distance between the two is less than the communication range R_c. To reduce the number of packets in the network a node is restricted from sending packet to it-self. The complete step 2 is depicted in equation 6.

$$v_n(sendhello) = \begin{cases} true & i \neq j \\ false & i = j \end{cases} \forall V \tag{6}$$

Step-3: Constructing and updating Adjacency Matrix: Adjacency matrix will be constructed for all the nodes in the network and in-turn for the overall network. It is a square matrix with the order $N \times N$, Where N is the number of nodes in the network. Each row of the matrix corresponds to information of the node in sequential order, i.e., row 1 for node 1, row 2 for node 2 and so on. The entries of the row corresponding to information of that particular node. A network with 8 nodes and its adjacency matrix A is given in figure 3 and figure 4 respectively, for better understanding.
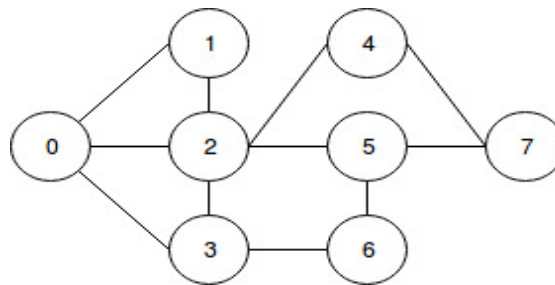

Figure 3. Network with 8 nodes.

$$A_{mn} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Figure 4. Adjacency matrix.

In the adjacency matrix given, m is the number of rows and n is number of columns. Positions of rows and columns indicate the node number plus the response it receives from the neighboring node. For example A_{00} corresponds to node number 0 and A_{01} corresponds to node 1's response when node 0 sends a hello packet to it. Entries in the adjacency matrix are filled with respect to equation number 7. As given in the equation every diagonal element in the matrix is zero which means that a node will not send packet to itself. If a node receives response to its hello packet from the neighboring node then it enters a 1 in the corresponding position and it will enter 0 in positions of nodes which did not respond to the hello packet. For example node 0 and node 1 are in range then A_{01} will have 1 and due to symmetry property of adjacency matrix A_{10} will also be 1. The diagonal elements of Amn are all zeros indicating that a node will not send a hello packet to itself. Equation 7 and 8 given below sums up the formation of adjacency matrix A_{mn}.

$$A_{mn} = \begin{cases} 1 & m \neq n \wedge \text{ if response to hello packet is } 1 \\ 0 & m = n \wedge \text{ if response to hello packet is } 0 \end{cases} \quad \forall \; V \tag{7}$$

$$numhave = \begin{cases} numhave + 1 & A_{mn} = 1 \\ numhave & A_{mn} = 0 \end{cases} \quad \forall \; V \tag{8}$$

Equation 8 indicates that variable numhave will be incremented for every node whenever a node finds a neighbour within its range.

5. Result

To evaluate the performance of proposed *TNA* algorithm, several simulations are carried out using an open simulation and graphics software called Processing. The results obtained by the proposed algorithms are compared with *ICABC*, *TMFA* and *DPath* algorithms. ICABC algorithms objective was to improve the overall lifetime of the deployed network by using the imperialist competitive algorithm. ICABC works for both random and deterministic deployments. The two-round maximum flow (TMFA) algorithm maximizes the number of barriers with a complexity of $O\,(n^5 m^5)$ TMFA first will calculate the number of potential paths in the network and then eliminates the conflicting paths. DPath algorithm studies strong barrier coverage with directional sensors. First, it introduces the coverage graph model for barrier coverage and then provides an optimal solution through integer linear programming.

The simulation of our algorithm is carried on an i5 CPU with 8GB of RAM. For calculation of lifetime of the network in weeks, we assume that one barrier has a lifetime of one week as mentioned in the *ICABC* algorithm. Therefore a number of barriers formed will be equal the lifetime of the network in the number of weeks. The width of the network is kept at 130meters and range of each node is 13meters which is less than half of the range assumed in the *ICABC* algorithm. We prove that our results are better than the existing algorithms even with half of the range of sensor nodes. Increase in range directly affects the power consumed by the sensor nodes which in turn effects the lifetime of the overall network.

Figure 5 gives the comparison of the result obtained by *TNA* algorithm with respect to number of nodes deployed. *TNA* algorithm has a superior performance for network size greater than 100 nodes when compared to *ICABC*, *TMFA* and *DPath* algorithms. One obvious reason is the novel approach adopted by *TNA* where number of packet exchange rate among the nodes is decreased. *TNA* provides an increase in lifetime by 144%, 90% and 211% when compared to *ICABC* and *TMFA* and *DPath* algorithms.

Figure 6 gives the performance of *TNA* algorithm in comparison with the other three. Here also *TNA* outperforms *ICABC*, *TMFA* and *DPath* algorithms by a great margin. Here the network width is varied from 100 to 180 meters with an incremental value of 20m. The range of each sensor is 30m as given in *ICABC*. The lifetime of *TNA* reduces with the increasing width of the network but the lifetime provided by *ICABC* is almost constant for network widths 120 to 180m.
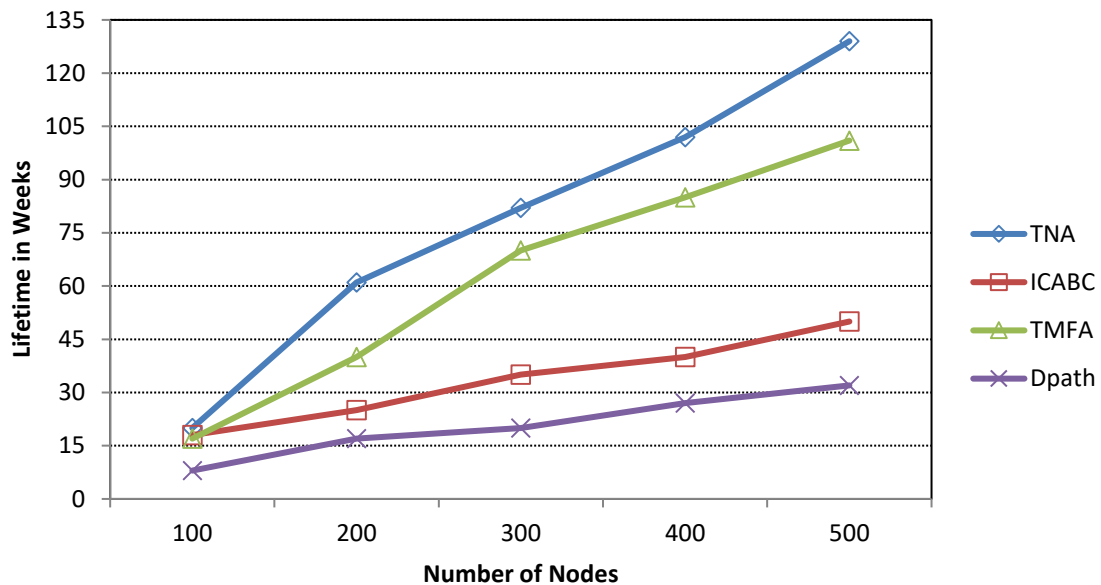

Figure 5. Comparison of network lifetime versus number of nodes

Figure 7 compares the performance of algorithms for lifetime versus range in meters. Here for *TNA* algorithm, the network width is kept constant at 180m. The range of the sensor is varied from 30 to 60m with an incremental value of 5m. Here *TNA* outperforms the other three algorithms until the range is 40m. After 40m *TNA* performance reduces when compared to *ICABC* but is still superior to *TMFA* and *DPath*. Here for *TNA,* the numbers of nodes are fixed at 500 and length of the area is 180m. One observation made is that when the range of the sensor is varied by keeping sensor number and length fixed, lifetime achieved is almost constant with a change of 10 weeks when the range is increased from 30 to 60m.
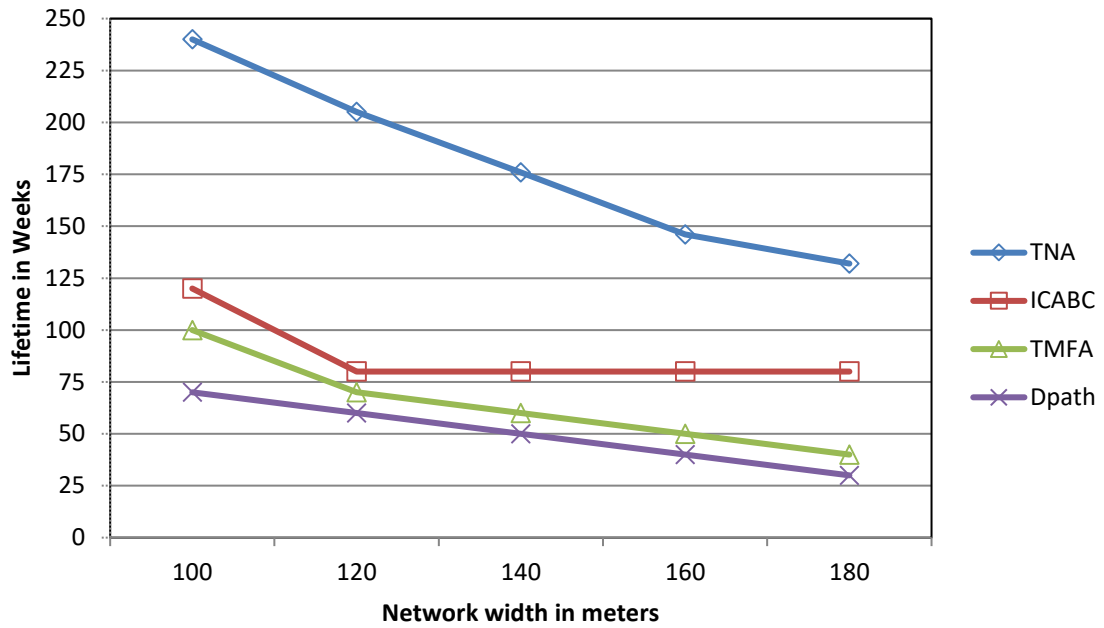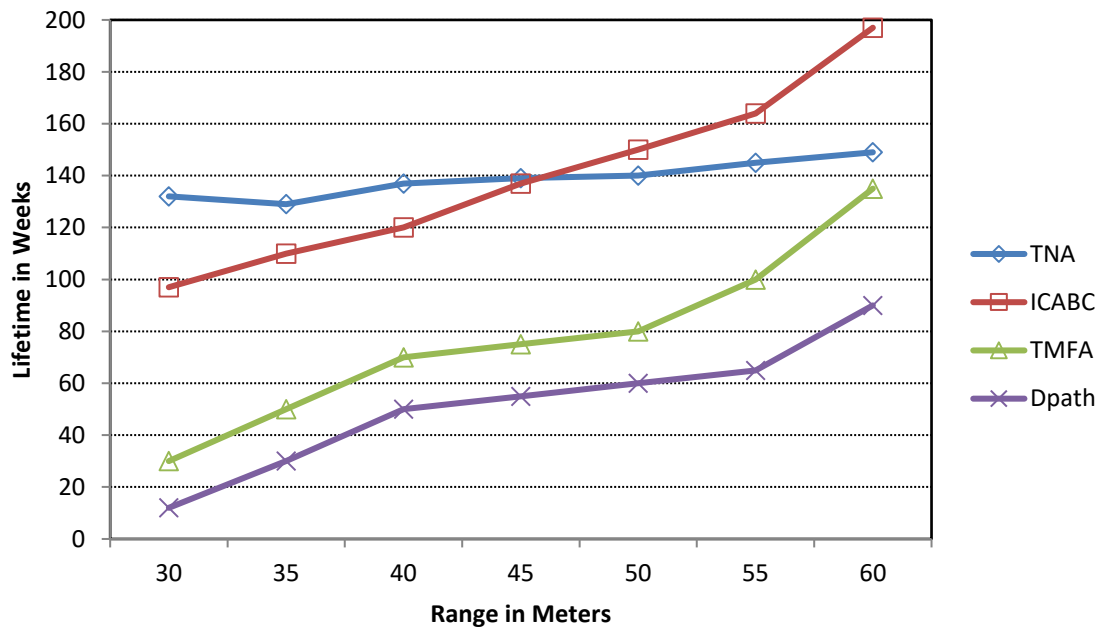
Figure 6. Lifetime versus network width

Figure 7. Lifetime versus sensor range

Results obtained for *Tell-Neighbor Algorithm* shows its superiority when compared to other state-of-art works in enhancing the lifetime of the network for barrier coverage. TNA has shown superior performance in terms of the lifetime of the network in weeks versus number of nodes randomly deployed in the network, width of the network and range of nodes in the network. TNA increases not only the overall operating lifetime of the network but also the number of barriers formed. Results show that the quality of service provides by TNA is better than the other existing algorithms. Authors of this article encourage readers to work on the convergence time of the network i.e., the time is taken by the nodes to get randomly deployed should be as small as possible while the quality of service is maintained.

6. Conclusion

We have presented a novel *Tell Neighbor Algorithm (TNA)* that increases the lifetime of the network while maintaining the barrier coverage. The simulation results show that the proposed *TNA* algorithm outperforms its counterparts *ICABC*, *TMFA* and *DPath* algorithms by a great margin. The graphs comparison proved the superiority of the proposed algorithm. Authors of this article would encourage the readers to apply optimization techniques such as gravitational search algorithm described in Bhunia et al. (2017). Also, the scope of applying neural and fuzzy as described in Joshi (2016) and, Begum and Dohi (2018) can be referred. System reliability analysis can be carried out when the number of sensor node increases beyond 1000, as the system may become unstable during simulation analysis. The amount of data exchanged between the sensors for barrier formation is huge, therefore data security is highly recommended. Cryptographic techniques mentioned in (Ahmad & Rushdi, 2018) can be explored for the same.

Conflict of Interest
Authors declare that there is no conflict of interest.

References

Ahmad, W., & Rushdi, A.M.A. (2018). A new cryptographic scheme utilizing the difficulty of big Boolean satisfiability. *International Journal of Mathematical, Engineering and Management Sciences, 3*(1), 47-61.

Ban, D., Yang, W., Jiang, J., Wen, J., & Dou, W. (2010). Energy-efficient algorithms for k-barrier coverage in mobile sensor networks. *International Journal of Computers Communications & Control, 5*(5), 616-624.

Begum, M., & Dohi, T. (2018). Optimal release time estimation of software system using box-cox transformation and neural network. *International Journal of Mathematical, Engineering and Management Sciences, 3*(2), 177-194

Bhunia, A.K., Duary, A., & Sahoo, L. (2017). A genetic algorithm based hybrid approach for reliability-redundancy optimization problem of a series system with multiple-choice. *International Journal of Mathematical, Engineering and Management Sciences, 2*(3), 185-212.

Chen, A., Kumar S., & Lai, T.H. (2007, September). Designing localized algorithms for barrier coverage In *Proceedings of the 13th annual ACM International Conference on Mobile Computing and Networking, 1990, ACM.*

Du, J., Wang, K., Liu, H., & Guo, D. (2013). Maximizing the lifetime of k-discrete barrier coverage using mobile sensors. *IEEE Sensors Journal, 13*(12), 4690-4701.

Han, R., Zhang, L., & Yang, W. (2016, September). Maximizing strong barriers in lifetime-heterogeneous directional sensor network *In Wireless Communication Systems (ISWCS), 2016, International Symposium on, IEEE.*

He, S., Gong, X., Zhang, J., Chen, J., & Sun, Y. (2014). Curve-based deployment for barrier coverage in wireless sensor networks. *IEEE Transactions on Wireless Communications, 13*(2), 724-735.

Huang, R.-M., X.-S., Qiu, & Ye, L. (2010, October). Probability-based fault detection in wireless sensor networks In *Network and Service Management (CNSM), 2010, International Conference on, IEEE.*

Joshi, R. (2016). Artificial neural network (ANN) based empirical interpolation of precipitation. *International Journal of Mathematical, Engineering and Management Sciences, 1*(3), 93-106.

Kim, K.-S., & Jin, G.-W. (2011, February). Maximizing the lifetime of a sensor network with barrier coverage In *Green and Smart Technology with Sensor Applications, 2012, Communications in Computer and Information Science,* Springer, pp. 347-354.

Kumar, S., Lai, T.H., & Arora, A. (2005, August). Barrier coverage with wireless sensors In *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking, 2005, ACM.*

Mostafaei, H. (2015). Stochastic barrier coverage in wireless sensor networks based on distributed learning automata. *Journal of Computer Communications, 55(C),* 51-61.

Mostafaei, H., Shojafar, M., Zaher, B., & Singhal, M. (2017). Barrier coverage of WSNs with the imperialist competitive algorithm. *The Journal of Supercomputing, 73*(11), 4957-4980.

Saipulla, A., Liu, B., Xing, G., Fu, X., & Wang, J. (2010, September). Barrier coverage with sensors of limited mobility In *Proceedings of the eleventh ACM International Symposium on Mobile ad hoc Networking and Computing, 2010, ACM.*

Wang, Z., Liao, J., Cao, Q., Qi, H., & Wang, Z. (2014). Achieving k-barrier coverage in hybrid directional sensor networks. *IEEE Transactions on Mobile Computing, 13*(7), 1443-1455.

Zhang, L., Tang, J., & Zhang, W. (2009, November). Strong barrier coverage with directional sensors. in *Global Telecommunications Conference, 2009, GLOBECOM IEEE.*