

Connected Point Coverage in Wireless Sensor Networks using Robust Spanning Trees

Pouya Ostovari

Department of Computer
and Information Sciences
Temple University
Philadelphia, Pennsylvania, USA
Email: ostovari@temple.edu

Mehdi Dehghan

Department of Computer Engineering
and Information Technology
Amirkabir University of Technology
Tehran, Iran
Email: dehghan@aut.ac.ir

Jie Wu

Department of Computer
and Information Sciences
Temple University
Philadelphia, Pennsylvania, USA
Email: jjewu@temple.edu

Abstract—Energy limitation is one of the most critical challenges in the area of sensor networks. Sleep scheduling mechanisms can reduce the energy consumption. Coverage mechanisms attempt to cover the area with the minimum possible number of sensors. There are many area coverage approaches which also consider the connectivity problem. However, in the area of point coverage, there are limited mechanisms that maintain connectivity. In this paper, we propose a point coverage mechanism and two connectivity mechanisms. We compare these mechanisms to one of the best methods that consider both point coverage and connectivity. In the point coverage mechanism, we present a method for computing the waiting time, which reduces the number of the required sensors. For preserving the connectivity, virtual robust spanning tree (VRST) and modified virtual robust spanning tree (MVRST) are proposed. These mechanisms are based on making a virtual spanning tree and converting this tree to a physical tree. In order to spread out sensed data to the sink from different paths and decrease the loss probability, instead of using a minimum spanning tree (MST) to connect nodes to the sink, we use a combination of distance of nodes and number of hops to select edges and construct the tree. The simulation results show that the proposed coverage method reduces energy consumption by up to 7% compared to the Cardei method. The VRST and MVRST use more energy than the Cardei method, but the average data loss decreases by up to 40%. Moreover, VRST and MVRST have less depth and data latency.

Index Terms—connectivity, energy efficiency, point coverage, sensor network, sleep scheduling, virtual tree.

I. INTRODUCTION

Coverage is one of the most important challenges in the area of sensor networks. Since the energy of sensors are limited, it is vital to cover the area with fewer sensors. Generally, coverage in sensor networks is divided into area coverage, point coverage, and boundary coverage subareas. Coverage does not ensure connectivity of nodes. However, many approaches have addressed both area coverage and connectivity, but limited number of approaches have covered both.

Cardei proposed one of the best approaches in the field of connected point coverage [1]. It contains two steps. Using the first step, the sensing nodes are selected. In the second step, some relay nodes are selected to make a MST between sensing nodes and the sink, based on Prim's algorithm, to ensure sink-connectivity. However, MST has a problem. Usually, MST is relatively deep. In addition, the sink does not have many

branches. Therefore, in the case of failure, we lose a significant amount of data.

In this paper, we modify MST and we introduce a balanced tree, which solves the problem of using MST to ensure connectivity. In Prim's algorithm, the cost of edges is the distance of nodes. In contrast, our proposed method uses a combination of distance and hop count as the cost of edges. In the first step, we use a distributed algorithm to compute the priority of nodes. This priority is computed based on the residual energy of nodes and the number of targets in their sensing area. Nodes with more residual energy and more targets in their sensing area have more priority than other nodes. In the second phase, we use another algorithm to select some relay nodes to construct a balanced tree between sensing nodes and the sink. To construct the tree, we first construct a virtual tree with targets and the sink, which is used as a skeleton to built an actual tree. After making a virtual tree, we convert it to a physical tree between nodes that are selected in the covering phase as a sensing node and the sink.

In comparison to the Cardei method, our proposed method uses less sensors to cover all targets. In addition, our proposed method is more balanced than the Cardei method; therefore, in the case of node failure, we will lose less data, as all branches are more balanced. However, our proposed trees uses more relay nodes to deliver sensed data to the sink, so the probability of a node failure in our tree is more of that the Cardei method.

The contributions of this paper are the following:

- We change one of the existing point coverage methods to reduce energy consumption of sensor nodes.
- To maintain connectivity, we propose a method to construct a balanced tree which is more robust against failure.
- We consider and avoid cycle formation which might happen during converting virtual trees to physical trees.

The remainder of the paper is organized as follows: Section 2 presents related works on connectivity and coverage. In Section 3, we present our proposed methods. Simulation results are shown in Section 4.

II. RELATED WORKS

A survey on coverage problems in wireless sensor networks is presented in [2]. The coverage problem is divided into three

subcategories: area coverage, point coverage and boundary coverage.

Most of the research in the field of sensor networks consists of area coverage. The goal of area coverage is to cover as much area as possible with the minimum number of sensors. Most of these researches use probabilistic or geometric approaches. It is shown in [3] that the problem of selecting a subset of sensors that cover the whole area is NP-Complete, and the authors proposed an efficient approximation algorithm to address area coverage. [4] proves that if the radio range is at least twice the sensing range, complete coverage of a area implies connectivity among the working set of nodes. In ACOS [5], each node computes the area which can be only covered by it. If this area is smaller than a threshold, then this node goes to active mode.

The goal of point coverage is to cover some specific points of the network. [6] proposes an efficient method to extend the sensor network lifetime by organizing the sensors into a maximal number of disjoint sets that are activated successively. It is shown in [7] that selecting disjoint sets do not necessarily result in a larger lifetime of the network than non-disjoint sets. In this approach, the nodes are organized into maximal numbers of set covers instead of disjoint-sets. [8] addresses the target coverage issue in wireless sensor networks that have sensors with adjustable sensing range. To solve this issue, linear programming, a localized greedy algorithm, and a distributed greedy algorithm, are proposed.

In the boundary coverage problem, the goal is to cover the network so as to minimize the probability that a mobile object which cross the barrier of network remains undetected. It is illustrated in [10] that the best path for an intruder from a source to a destination is in the Voronoi diagram. [11] supposes that the view field of sensor nodes is limited and sensors have a directional camera. An optimal polynomial time algorithm was presented for computing the worst-case breach coverage. Breach is the maximal distance that any hostile target cannot be detected by the sensors while traveling through a region. [12] studies the trade-off between the number of sensors and the breach detection probability.

The studies that are more relevant to our approach are [1] and [9]. [9] presented a distributed approach for the connected point coverage problem of wireless sensor networks. This approach selects a dominate set of sensors to serve as a backbone to ensure connectivity. In addition, it uses a distributed algorithm similar to [8] to cover all targets. [1] uses a distributed algorithm to select sensing nodes. During the second phase, it selects relay nodes. For this purpose, it makes a virtual tree between targets and the sink and then converts it to a physical tree of sensing nodes and the sink.

Our work is an improvement of [1]. We changed the coverage method to reduce the number of active sensors that are needed to cover all of the targets. Instead of using MST to maintain sink-connectivity, we use a balanced tree to decrease the amount of data loss in sensor networks.

III. OUR APPROACH

In this section we propose our distributed method for connected point coverage, which is based on the Cardei method. We want to cover some targets in a homogeneous sensor network, using an efficient number of sensors while still preserving sink-connectivity. Nodes and targets are stationary and we suppose that each node knows the location of all targets and the sink.

The algorithm runs in rounds. Each round begins with an initialization phase in which every sensor decides whether to be active or inactive for the rest of the round. This phase is divided into two steps: In the first step, *sensing nodes selection*, the sensor nodes are selected such that the union of them covers all the targets. In the second step, *relay nodes selection*, additional relay nodes are chosen so as to guarantee sink-connectivity. These steps are explained in the following sections.

A. Sensing Nodes Selection

In this phase a set of efficient sensing nodes is selected to cover all the targets in the field. Since this problem is NP-complete [1] we use a distributed greedy heuristic to address it. We start from selecting sensors with more targets in their sensing area and more residual energy as sensing nodes and we keep doing it until covering all the targets. To make this heuristic distributed, each sensor node computes a waiting time based on its residual energy and the number of uncovered targets it can cover. We use these waiting times to select sensing nodes in increasing order of their priority. The sensor that covers more uncovered targets and has more residual energy will get less waiting time, and thus more priority than other sensors. The waiting time of a sensor s_u is computed by the equation (1):

$$T_u = (1 - \alpha * \frac{E'_u}{E} - \beta * \frac{|TargetS_u|}{M}) * W_1 - T'_u \quad (1)$$

where:

- E'_u : the residual energy of sensor s_u
- E : the initial energy of sensors
- M : the number of targets in the network
- W_1 : the maximum waiting time
- α, β : weights which are assigned to the residual energy and the number of uncovered targets
- $TargetS_u$: Targets which are in sensing range of node s_u and have not been covered by any sensor node yet
- T'_u : the waiting time which sensor s_u has passed

When the waiting time of a sensor s_u is finished, it is selected as a sensing node and it acts as the supervisor of all the targets in the set $TargetS_u$. The supervisor of a target is the first sensor that passed its waiting time and covers it. Next, this sensor broadcasts a notification message to inform its neighbors about the targets covered by it.

When a sensor passing its waiting time receives a notification message from its neighbors, it updates its waiting time

and the set $TargetS_u$ if it has at least one target within its sensing range in common with the targets mentioned in the message. Assuming that R_S is the sensors' sensing range and R_c is the sensors' communication range, the maximum distance between two sensors with a common target in their sensing range is $2R_S$, if $R_S < 2R_C$. Therefore, it is adequate to broadcast the notification message in two hops.

The basic difference between our proposed method for sensing nodes selection and the Cardei method is that in the former the elapsed waiting time of a sensor is not considered in re-computation of the waiting time when it receives a notification message. As a result, sensors with a high number of targets in common with other sensors must update their waiting time many times, so their priority decreases and their waiting time increases each time, regardless of the time that passed up until now.

B. Relay Nodes Selection

After selecting sensing nodes and covering all the targets, relay nodes must be selected to provide sink-connectivity. We suppose here that each sensor knows the location of the sink and all the targets.

Achieving sink-connectivity, each selected sensing node s_u makes a virtual tree based on the set of targets and the sink. Unlike the Cardei method, which uses Prim's algorithm to construct a virtual minimum spanning tree, we use a modified version of *robust spanning tree* [13] in order to make our approach robust against failure of nodes. In this spanning tree, which we call a *virtual robust spanning tree* (VRST), the root is the sink and other vertices are targets. This tree serves as a virtual skeleton for the considered network. In order to convert this virtual tree to a physical tree of sensors, each sensing node broadcasts a message to find the supervisor of the target which is the parent of its covered target.

The VRST algorithm is similar to Prim's algorithm, but rather than using the edge's length (edge's cost) for choosing an edge, it uses a combination of the edge's length and the vertex's depth (hop count). Here, we assume that the edge's length is Euclidean distance between connecting points. In this algorithm, cost is computed as follows:

$$Cost = \lambda * hop\ count + (1 - \lambda) * (weight\ of\ path_i), \quad (2)$$

which λ is a function of the depth of a vertex:

$$\lambda_i = 1 - \frac{h_i}{\epsilon_1} \quad (3)$$

$$weight\ of\ path_i = weight\ of\ path_j + z_{i,j} \quad (4)$$

Descriptions of notations are as follows:

- hop count: number of hops to the sink
- $weight\ of\ path_i$: sum of the edges' cost which connect vertex i to the sink
- ϵ_1 : depth of the MST (maximum hop count)
- h_i : depth of vertex i

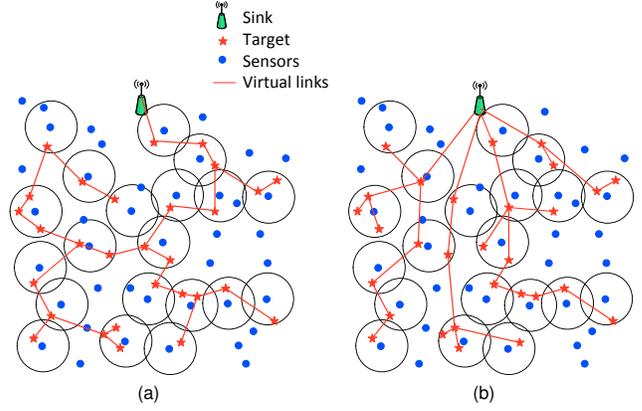


Fig. 1. Examples of virtual trees in (a) MST and (b) VRST

- $z_{i,j}$: length of edge (i, j)

In relay node selections, we first construct a MST based on the set of targets and the sink, using the Prim's algorithm, and we set ϵ_1 to the depth of this tree. Subsequently, cost is computed for each vertex and for the edges which connect it to a vertex of the tree according to equation (2). The vertex connected to the edge with the minimum cost is selected and added along with this edge to the tree.

The closer a vertex is to the sink, the greater λ it will have. Therefore, the edge's length has a greater effect on computing the edge's cost. As a result, vertices nearer to the sink will directly connect to it yielding a fat tree around the sink. On the other hand, vertices farther from the sink will have more choices to connect. As a result, the depth of the tree will decrease. Farther vertices have smaller λ , so the path weight has more of an effect on their edges' cost. Thus, these vertices will connect to the tree along edges which connect them to the path with the minimum cost.

Figures 1(a) and 1(b) show a MST built with Prim's algorithm and a VRST, respectively. Obviously, the depth of the VRST is far less than the MST.

As said before the supervisor of a target is the first sensor that passed its waiting time and covers it. After constructing the virtual tree, supervisors of targets must connect together via physical paths. Therefore, we have to select some relay nodes to connect the supervisors of targets. For each target t_i , the sensing node s_u starts the selection of relay nodes along the virtual link $(t_i, \pi(t_i))$, where $\pi(t_i)$ is the parent of target t_i in the virtual tree.

Every sensing node s_u broadcasts a control message *RELAY_REQ* for each target under its supervision. This message contains the location of sensing node s_u , destination target $\pi(t_i)$, and the maximum distance from sensing node s_u to the supervisor of target $\pi(t_i)$. The maximum distance of two supervisors can be calculated using equation (5). Each sensor that receives this message computes its distance from the source sensor. If this distance is less than the maximum distance mentioned in the message *RELAY_REQ*, then it forwards the message.

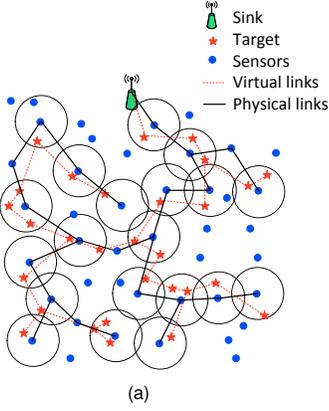


Fig. 2. Examples of virtual and physical trees in (a) Cardei and (b) VRST.

$$\text{Max distance} = \text{distance}(t_i, \pi(t_i)) + 2R_s \quad (5)$$

When the supervisor of target $\pi(t_i)$ receives the message *RELAY_REQ*, it will reply to this message. It will send this reply message to the sensing node s_u , through one of the paths in which it received the message *RELAY_REQ*. The supervisor of target $\pi(t_i)$ can select one of the relay sensors who delivered the message *RELAY_REQ*, based on one of the following criteria:

- Relay sensor closest to the node s_u
- The first relay node which delivered the message
- Relay node with the most residual energy

This procedure will continue until the reply message is delivered to the source sensing node. Consequently, a physical path is built between sensing nodes and the sink. All nodes participating in delivering the reply message are chosen as relay nodes.

In Figure 2, virtual trees made by the Cardei method, the VRST method, and physical trees built based on them are shown. It can be seen in this figure that the physical tree made based on the VRST method, has less depth and more paths to the sink compared to the physical tree made based on MST. Therefore, in our proposed method, we will have less data loss when a sensor node fails.

In equation (4), the cost of connecting vertex (target) i to vertex j as a member of the tree is equal to the sum of the length of edge (i, j) and the path's weight from vertex j to the sink. Hence, in addition to the direct effect of depth on equation (2) as hop count, it indirectly affects the path weight from vertex j to the sink; thus, it has a double effect on this equation. In other words, not only is the vertex's depth important to vertices near the sink, but also important to the vertices far from it. Its immediate result is that the vertex's depth has a more important role than the edge's length in selecting edges. For this reason, in the second proposed method (MVRST), for relay nodes selection, we ignore the weight of path from vertex j to the sink and we consider the

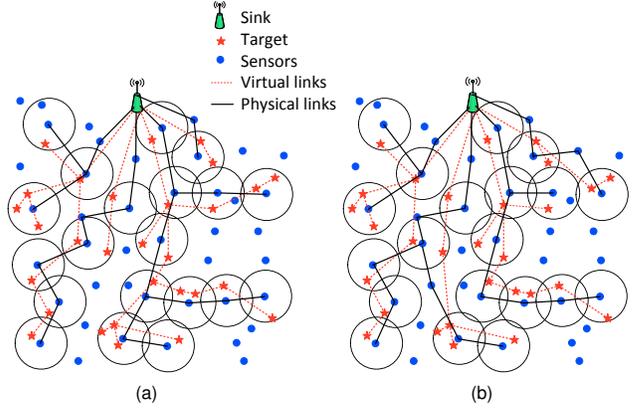


Fig. 3. Examples of virtual and physical trees in (a) MVRST and (b) VRST.

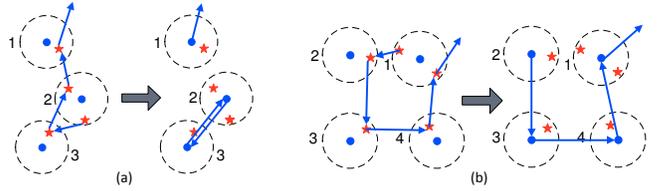


Fig. 4. Forming cycles during converting virtual trees to physical trees.

weight of edge (i, j) as the weight of the path from vertex i to the sink:

$$\text{weight of path}_i = z_{(i,j)} \quad (6)$$

As shown in Figure 3, virtual and physical trees made by the MVRST method have less depth than trees made by the VRST method.

C. Avoiding Cycle Formation

Since some of the sensing nodes cover more than one target and are supervisors of some or all of them, it is possible that while we are converting a virtual tree to a physical tree, cycles are formed. This happens when a sensing node is a supervisor of more than one target and these targets have different parents. These cycles could have a length greater than or equal to two.

In Figure 4(a), a cycle with a length of two is shown. Sensor 2 covers two targets whose parents are different. If this sensor chooses sensor 3 as its parent during the conversion from virtual to physical tree, a cycle will be formed between sensors 2 and 3.

In order to avoid cycles with length of two, when a sensing node s_u , which is supervisor of more than one target, wants to select sensing node s_j as its parent, it must check if it has received any *RELAY_REQ* message from node s_j that its destination is node s_u . Receiving this message means that it is possible that sensing node s_j selects sensing node s_u as its parent, and in this case, a cycle with a length of two will be formed. For cases with a length more than two, we could use two approaches:

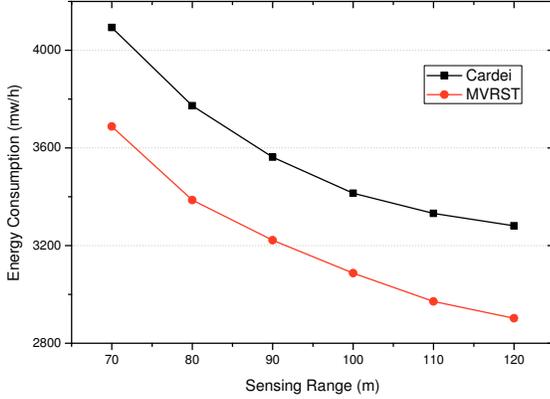


Fig. 5. Effect of sensing range on energy consumption: $R_c = 120m$

- Using cycle detection algorithms: Sensors covering more than one target run a cycle detection algorithm. If they detect a cycle, they choose another sensor as their parent.
- Packet examination: Sensors send their packets in one of the existing paths. A returned packet to its source indicates that there is a cycle in the network, and this sensor should change its parent.

/bf We used packet examination approach in our proposed methods, since it is more efficient than cycle detection approach. The reason is that the computational cost of cycle detection algorithms is high, which is $O(n^3)$, and the probability of formation of cycles with length more than two is low (as observed in simulations, less than 5 cycles in 1000 run times).

IV. SIMULATION RESULTS

In this section we setup a set of simulation experiments to compare the performance of our proposed methods with that of the Cardei method. For this purpose, we implemented a simulator in the environment of MATLAB. We considered a stationary network in which sensor nodes and targets are scattered randomly in a square field $500m \times 500m$. Other simulation parameters are as follows:

- 500 sensors.
- 50 targets.
- Sensing energy consumption in a range of 50m is 20mW/s.
- Communication energy consumption in a range of 80m is 60mW/s.

For sensing and communication ranges greater than the specified ranges, we compute energy consumption as a square power. All sensors are homogeneous, so they have equal sensing and communication range. For each scenario, we executed simulations 200 times and we showed the average outputs in charts.

In the first experiment, we compared our proposed coverage method with the Cardei coverage method. In both methods we used the same relay nodes selection scheme, proposed by

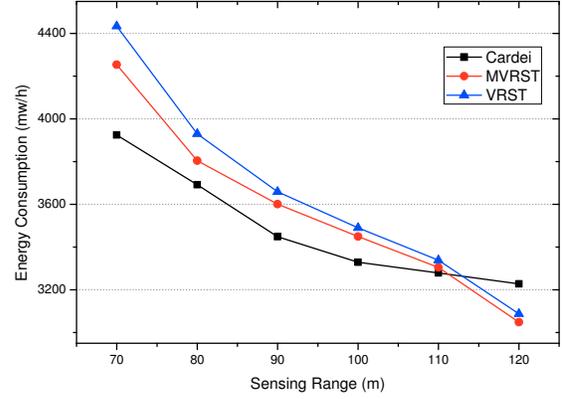


Fig. 6. Effect of sensing range on energy consumption: $R_c = 120m$

Cardei. Energy consumption of all the nodes, as shown in Figure 5, is the sum of energy consumption of sensing nodes and relay nodes. In this figure, the communication range is 120m. It can be seen that energy consumption of our proposed coverage method is less than that of the Cardei coverage method. The difference between these methods increases by increasing the sensing range. The reason is that by increasing the sensing range, the number of common targets covered by distinct sensors increases.

In Figure 6, we compared the VRST and MVRST methods against the Cardei method. In this figure we combined our proposed coverage method with the VRST and MVRST methods. Since the VRST method gives more priority to the depth of the virtual tree and tries to decrease it, the number of relay nodes and consequently energy consumption of this method is more than that of others. The Cardei method has the least energy consumption, since it uses a MST for relay nodes selection. Finally, as the MVRST method gives more importance to energy consumption during the construction of the virtual tree, whereas the VRST method, its energy consumption is less. Notice in Figure 6 that the energy consumptions of the VRST and MVRST methods are decreased more than the Cardei method by increasing the sensing range. As mentioned before, the reason is that by increasing the sensing range, the effect of our proposed coverage method increases.

In Figures 7 and 8, we compared the maximum and average data loss of our proposed methods with that of the Cardei method in a scenario of a single node failure. As expected, the maximum and average amount of data loss in the Cardei method is significantly more than the VRST and MVRST methods, since the Cardei method does not consider the depth of the tree and the number of transmission paths to the sink. It can be observe in Figure 7 that there are some critical nodes in the Cardei method, which their failure will result in a loss of more than 95% of sensed data.

Figures 9 and 10 illustrate the effect of communication range on the maximum and average data loss. In these figures the slope of all charts is almost zero. This is because that

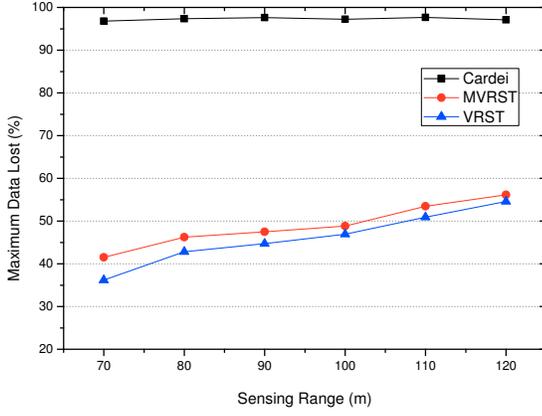


Fig. 7. Effect of sensing range on the maximum data loss: $R_c = 120m$

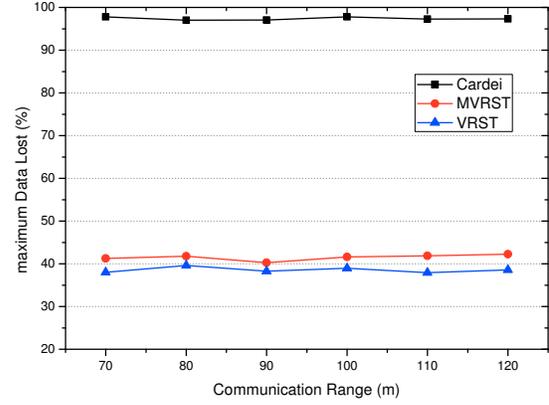


Fig. 9. Effect of communication range on the maximum data loss: $R_s = 70m$

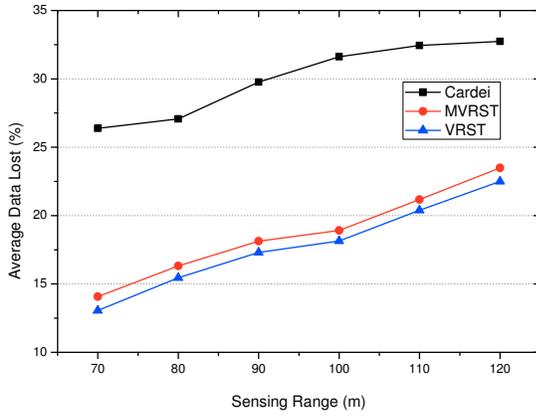


Fig. 8. Effect of sensing range on the average data loss: $R_c = 120m$

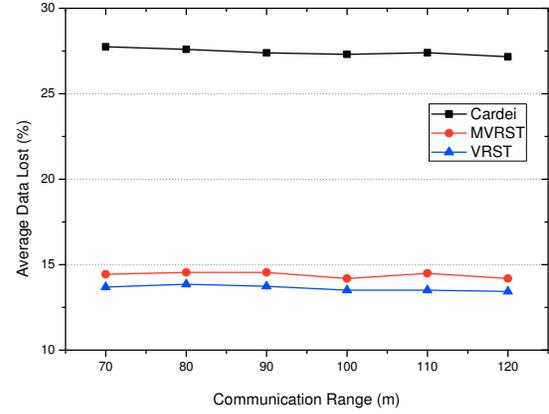


Fig. 10. Effect of communication range on the average data loss: $R_s = 70m$

communication range only affects the number of relay nodes and it has no effect on the construction of virtual trees and number of transmission paths to the sink. It is clear from these figures that the amount of data loss only depends on the structure of the virtual trees.

In the next experiment, we divided data loss into 10% slots and we computed the probability of failure which could result in a given percent of data loss. The probability of failure in our proposed methods is more than that of the Cardei method, since the number of relay nodes in VRST and MVRST are greater than that method. Therefore, in the VRST and MVRST methods, the probability of a failure which causes a low percentage of data loss is more than that of the Cardei method, as shown in Figure 11. However, in our methods, the probability of a failure which causes more than 40% data loss is less than that of the Cardei method. In other words, our methods reduce the probability of a high percentage of data loss, but they increase the probability of a low percentage of data loss.

The most important parameter in transmission delay is the depth of sensing nodes, which can be used to estimate the transmission delay. Consequently, we compared the average

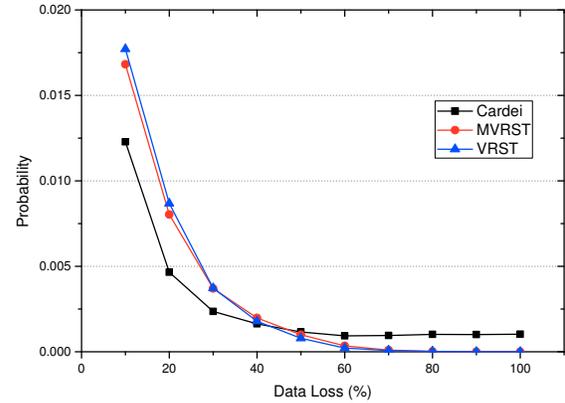


Fig. 11. Probability of data loss: $R_s = 70m$ and $R_c = 100m$.

sensing nodes' depth of our proposed methods with that of the Cardei method in Figure 12, instead of comparing transmission delay. Contrary to the Cardei method, the depth of the nodes is considered in the VRST and MVRST methods, and as

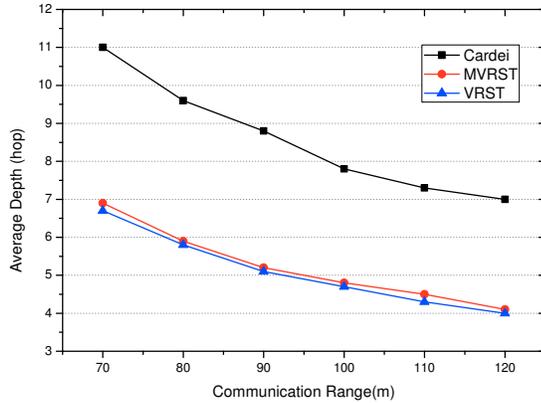


Fig. 12. Average depth: $R_s = 70m$

expected, our proposed methods have less depth and less transmission delay than those of the Cardei method. Since the nodes' depth has a greater role in the selection of the nodes in the VRST method, MVRST has more depth than VRST. The number of relay nodes decreases by increasing the communication range, and the depth of the tree in all methods decreases.

In both our proposed coverage method and Cardei coverage method, each node uses a linear equation to compute the waiting time. Therefore, complexities of both coverage methods are $O(c)$. On the other hand, in Cardei connectivity method, each node runs Prim's algorithm to construct a MST. Thus, the complexity of Cardei connectivity method is the same as Prim's algorithm, which is $O(n^3)$. In the VRST and MVRST methods, we use Prim's algorithm to compute the maximum depth of MST, and then, we run an algorithm similar to Prim's algorithm to construct a spanning tree. The only difference between Prim's algorithm and the algorithm we used is the way we assign weights to the edges. Therefore, the complexity of the VRST and MVRST methods are $O(n^3) + O(n^3)$, which is equal to $O(n^3)$.

V. CONCLUSION

In this paper, we proposed a point coverage mechanism in addition to two connectivity mechanisms. In comparison with existing approaches, our point coverage mechanism needs less sensing nodes to cover all the targets, and as a result, less energy consumption. We presented the VRST and MVRST methods to preserve connectivity. In these mechanisms, in order to spread out sensed data to the sink from different paths and decreasing the loss probability, we used a combination of distance of nodes and number of hops to select edges and construct the tree. Our simulations show that the proposed coverage method reduces the energy consumption by up to 7% compared to the Cardei method. The VRST and MVRST use more energy than the previous method, but the average data loss decreases by up to 40%. Moreover, our approach has less data latency.

REFERENCES

- [1] I. Cardei and M. Cardei, "Energy-efficient connected-coverage in wireless sensor networks," *International Journal of Sensor Networks*, Vol. 3, No. 3, pp. 201-210, May 2008.
- [2] M. Cardei and J. Wu, "Energy-Efficient Coverage Problems in Wireless Ad Hoc Sensor Networks," *Computer Communications*, Vol. 29, No. 4, pp. 413-420, February, 2006.
- [3] M. Hefeeda and M. Bagheri, "Randomized k-coverage algorithms for dense sensor networks," in Proc. of *IEEE INFOCOM: 26th IEEE International Conference on Computer Communications*, Anchorage, AK, United States, pp. 2376-2380, 2007.
- [4] H. Zhang and J. C. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks," *Ad Hoc and Sensor Wireless Networks*, Vol. 1, No. 1-2, pp. 89-124, March 2005.
- [5] Y. Cai, M. Li, and M. Y. Wu, "An area-based collaborative sleeping protocol for wireless sensor networks," in Proc. of *8th Asia-Pacific Web Conference*, Harbin, China, pp. 449-460, 2006.
- [6] M. Cardei and D. Du, "Improving wireless sensor network lifetime through power aware organization," *Wireless Networks*, Vol. 11, No. 3, pp. 333-340, May 2005.
- [7] M. Cardei, M. T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *IEEE INFOCOM 2005*, Miami, FL, United States, pp. 1976-1984, 2005.
- [8] M. Cardei, J. Wu and M. Lu "Maximum network lifetime in wireless sensor networks with adjustable sensing ranges," in Proc. of *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, (WiMob), pp. 438-445, 2005.
- [9] M. Lu, J. Wu, M. Cardei, and M. Li, "Energy-efficient connected coverage of discrete targets in wireless sensor networks," *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 4, No. 3-4, pp. 137-147, 2009.
- [10] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M.B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in Proc. of *IEEE INFOCOM*, 2001.
- [11] J. Adriaens, S. Megerian, and M. Potkonjak, "Optimal worst-case coverage of directional field-of-view sensor networks," in Proc. of *3rd Annual IEEE Communications Society on Sensor and Ad hoc Communications and Networks*, (SECON), Reston, VA, United States, pp. 336-345, 2007.
- [12] E. Onur, C. Ersoy, and H. Delic, "How many sensors for an acceptable breach detection probability?," *Computer Communications*, Vol. 29, No. 2, pp. 173-182, January 2006.
- [13] D. England, B. Veeravalli, and J. B. Weissman, "A Robust Spanning Tree Topology for Data Collection and Dissemination in Distributed Environments," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 18, No. 5, pp. 608-620, May 2007.