

An Energy-Efficient Tracking Algorithm to Trace a Radioactive Mobile Target in a Wireless Sensor Network

Natarajan Meghanathan¹, Tiffani Gardner¹ & Justin Lewis¹

¹ Department of Computer Science, Jackson State University, Jackson, MS, USA

Correspondence: Natarajan Meghanathan, Department of Computer Science, Mailbox 18839, Jackson State University, Jackson, MS 39217, USA. Tel: 1-601-979-3661. E-mail: natarajan.meghanathan@jsums.edu

Received: January 25, 2015

Accepted: February 10, 2015

Online Published: March 25, 2015

doi:10.5539/cis.v8n2p15

URL: <http://dx.doi.org/10.5539/cis.v8n2p15>

The research is financed by the Massie Chair Grant (#: DE-NA0000654) at Jackson State University

Abstract

In this paper, we propose an energy-efficient tracking algorithm for predicting the trajectory of a mobile radioactive target in a wireless sensor network. The sensor nodes are assumed to be capable of detecting the strength of the background radioactive radiations as well as the signals emanating from a mobile radiological dispersal device (RDD). As the individual RDD signals may not be easily distinguishable enough from the background radiation, we propose that each sensor node sum up the strength of the signals sensed in its neighborhood for a sampling time period, and then (at the end of this time period) forward the value of the sum of the signals sensed in the neighborhood to a control center (sink). The sink identifies the sensor nodes (suspect nodes) that report relatively larger values for the sum of the signal strengths that is different from those of others; the arithmetic mean of the X and Y coordinates of the suspect sensor nodes is predicted as the location of the RDD at a time instant corresponding to the middle of the sampling time period. We evaluate the difference between the predicted and exact locations of the RDD trajectory over time as a function of the different operating parameters (such as the RDD velocity, transmission and sensing range of the sensor nodes and the duration of the sampling time period) as well as evaluate the network lifetime and node lifetime incurred due to exhaustion of the energy levels of the sensor nodes.

Keywords: tracking, mobile target, radioactive dispersal device, sensor networks, simulations

1. Introduction

Wireless sensor networks (WSNs) have been used for several environmental monitoring applications. These include applications wherein a mobile target is traced based on some characteristic signals (quantified as data) of the target sensed in the neighborhood and forwarded by the sensor nodes; a control center (sink) queries the sensor nodes either in random or in a particular order and the sensor nodes forward the sensed data (without much processing) to the sink. The sink aggregates all of the queried data and processes them to evolve a trajectory for the target. It would be more energy-consuming to let each sensor node directly report to the sink. Several energy-efficient data gathering algorithms (e.g., Meghanathan, 2012) have been proposed in the literature; the data sensed at each sensor node is forwarded along a communication topology (like a tree rooted at a leader node) in tandem with the data sensed at the other sensor nodes.

In this paper, we envision an application of WSNs to trace the trajectory of a mobile radioactive dispersal device (RDD) that randomly moves around in the network. The RDD emits low-level radiation as it moves along. We deploy a network of sensor nodes (randomly distributed throughout the network) with each sensor node capable of sensing the radioactive signals in its neighborhood (defined by a sensing range, which is half of the transmission range of a sensor node). It is quantitatively very difficult (and would be even erroneous) to distinguish the strength of a RDD signal sensed at a sensor node from the strength of the background radiations in the vicinity. At the same time, it would be very energy-consuming to report the sensed data for every time unit. We propose a tracking algorithm wherein each sensor node simply sums up the strengths of the radioactive signals sensed in the neighborhood (that includes the RDD signals if the device is moving around within the sensing neighborhood of the sensor node) for a time period. At the end of the sampling time period, the sensor

nodes forward their summed signal strength data as a single value to the sink along a data gathering tree. We construct a minimum-distance spanning tree based data gathering tree in a distributed fashion, as proposed in a recent work (Meghanathan, 2014) and use the DG tree as the basis for aggregating the radio signal data towards the sink. The sink picks the top x sensor nodes (in our simulations $x = 5$ sensor nodes) that report sum values (for the radioactive signals in their neighborhood), which are the largest among those learnt from the sensor nodes. We refer to these sensor nodes as the suspect sensor nodes in whose neighborhood the RDD is likely to be moving for the duration of the particular sampling time period. The sink computes the average of the X and Y coordinates of the suspect sensor nodes and predicts the calculated average value as the location of the RDD at the middle of the sampling time period. We continue the above procedure for the entire simulation period or when the network gets disconnected due to the failure of one or more sensor nodes. We determine the distance between the predicted and exact RDD locations (referred to as the distance error and determine its median value) for the entire simulation time or network lifetime, collected over the sampling time periods.

As the sensor nodes operate with limited battery charge, they are likely to run out of energy during the course of data aggregation and mobile target tracking. A sensor node is considered to have died if it runs out of its initial battery charge. The accuracy of the prediction gets affected due to the failure of several sensor nodes (especially in the neighborhood wherein the RDD moves). However, like the median distance error, the node lifetime (time of first node failure) and network lifetime (time of network disconnection due to multiple node failures) are also dependent variables. We identify the independent variables of the network and the operating environment to be parameters such as the velocity of the RDD, the transmission range (and in turn the sensing range) of the sensor nodes, and the sampling time period. Through extensive simulations, we evaluate the impact of the above three operating parameters on the median distance error, the node lifetime and network lifetime.

The rest of the paper is organized as follows: Section 2 describes the system model and the algorithm to construct a data gathering tree in a distributed fashion. Section 3 presents the procedure employed at the individual sensor nodes and the sink with regards to tracking a mobile target. Section 4 presents the simulation models for generating the mobility profile of the RDD and presents detailed simulation results for various operating conditions as well as discusses the observations. Section 5 presents related work on tracking mobile targets using wireless sensor networks. Section 6 concludes the paper.

2. System Model and Data Gathering Algorithm

2.1 System Model

We model the wireless sensor network as a unit-disk graph; the sensor nodes are the vertices and there exists an edge between two vertices if the Euclidean distance between the corresponding sensor nodes is within the transmission range of the nodes. All the sensor nodes are assumed to operate with an identical transmission range (TR). Each sensor node maintains a list of neighbor nodes that are within the transmission range and that have positive residual energy. The sensing range of a node (SR) is the distance (radius) within whose neighborhood the sensor node could sense any data. To ensure that coverage implies connectivity, it has been proved that the transmission range has to be at least twice the sensing range of the nodes (Zhou & Hou, 2005); hence, we set the sensing range of the nodes to be half the transmission range in this paper. The network boundary is assumed to be $[0...XMAX][0...YMAX]$. The sensor nodes are assumed to be static. All nodes are peers of each other; so, any node could end up being the leader node (root node of the tree) or the intermediate nodes or the leaf nodes. Even though we use a data gathering tree (which is hierarchical in nature), it does not mean the whole network is hierarchical. The data gathering tree is refreshed periodically and the roles of leader node, intermediate node and leaf nodes could go to any tree. Topology changes are triggered due to the failure of one or more sensor nodes. When a sensor node fails due to the exhaustion of battery charge, all its associated edges are removed from the network graph and the neighborhood of the nodes changes accordingly. The sensor nodes are assumed to be able to sense the radiation data (of the RDD and the background radiation) for every time unit.

Include in these subsections the information essential to comprehend and replicate the study. Insufficient detail leaves the reader with questions; too much detail burdens the reader with irrelevant information. Consider using appendices and/or a supplemental website for more detailed information.

2.2 Data Gathering Algorithm

The data gathering algorithm is executed at the time of network startup to establish a data gathering tree (DG tree) as the underlying communication topology for the sensor nodes to report their sensed data and their current location to the sink. The DG tree is also then refreshed for every $t_{Refresh}$ seconds and whenever a sensor node fails and a new DG tree needs to be constructed involving only the live sensor nodes (those that have a positive residual energy). Sensor nodes lose energy to establish or refresh/reconstruct a DG tree as well as to transmit and

receive data on the DG tree.

The sink node randomly picks a sensor node to be the root node (a.k.a. leader node) of the DG tree and sends a LEADER NODE NOMINATION message. If the chosen sensor node is live (has a positive residual energy), then it sends a confirmation; otherwise not. If the sink does not receive a confirmation within a certain time, it randomly chooses another sensor node and sends a LEADER NODE NOMINATION message; the above process is repeated until a sensor node accepts to serve as the leader node. The leader node then initiates a TREE CONSTRUCT message by recording its own location in the message and broadcasting it in its neighborhood. The TREE CONSTRUCT message contains a forwarding list that is updated (with the ID and X, Y coordinates of the forwarding node) each time a node forwards it. A node receiving the TREE CONSTRUCT message estimates the distance to the node from which the TREE CONSTRUCT message was received. Initially, each node is assumed to be of an *estimated tree-join distance* of ∞ . When a sensor node receives a TREE CONSTRUCT message from a forwarding node the distance to which is less than the most recently estimated value of the tree-join distance, then the sensor node decides to connect to the tree through the forwarding node and updates its tree-join distance to be the distance to the forwarding node; the sensor node further updates the forwarding list with its own ID and location and broadcasts further in its neighborhood. If a sensor node sees its ID already recorded in the forwarding list of a TREE CONSTRUCT message, then the sensor node drops the message (to avoid looping). After the broadcast has stopped in the neighborhood, each sensor node sends an UPSTREAM NODE SELECTION message to the upstream forwarding node to which it has the minimum tree-join distance. This way, the DG tree is setup with the leader node and the other sensor nodes at various levels of the tree. At the time of startup, each sensor node updates its location information to the sink using the QUERY DATA message propagated up and down the DG tree (see Section 3).

3. Procedure for Mobile Target Tracking

Each sensor node is assumed to be capable of detecting the strength of the radioactive signals in the background (assumed to be present all the time) as well as capable of detecting the strength of the radioactive signals emanating from a RDD moving within the neighborhood of its sensing range. For simplicity, we assume that the cumulative strength of the radioactive signals detected by a sensor node to be a discrete value (say, 2) if an RDD is moving within its sensing range and the cumulative strength to be a relatively lower discrete value (say, 1) if no RDD is within the sensing range and the signals sensed account mainly for the background radiation. Each sensor node maintains a sum of the sensed signal strength values since the last time period the sensor node reported the data to the sink; the sum is reset to 0 after the data is reported and is again incremented for the subsequent time units.

When it is time to query the network for data, the sink sends a QUERY DATA message to the current leader node of the DG tree. The leader node in turn forwards the QUERY DATA message downstream along the edges of the DG tree. The QUERY DATA message contains entries for each sensor node to insert the locally maintained sum of the sensed signal strength value at the node. Each intermediate sensor node receives the QUERY DATA message once from its upstream node and sends to all its immediate downstream child nodes; the intermediate sensor node waits to receive an updated QUERY DATA message from all its immediate downstream child nodes, inserts its own data and forwards to its upstream node in the DG tree. The leader node receives the updated QUERY DATA message from the sub tree rooted at all its immediate downstream child nodes. The leader node merely updates its own entry in the QUERY DATA message and forwards the message to the sink node.

Upon receiving the QUERY DATA message from the leader node, the sink determines the top x sensor nodes (in our case, we pick 5 sensor nodes) that reported the largest sum value; these sensor nodes are branded as the suspect sensor nodes within whose neighborhood the RDD could be potentially moving around. The sink computes the average of the X and Y coordinates of the suspect sensor nodes and the average value is considered the predicted X and Y coordinate location of the RDD at the time unit corresponding to the middle of the sampling time period. For quantitative evaluation purposes, we compute the distance between the predicted X and Y coordinates of the RDD and its exact X and Y coordinates at the corresponding time unit, and refer to this distance as the *distance error*.

As we use a minimum distance spanning tree-based data gathering tree, we anticipate the proposed tracking method to be energy-efficient as the transmissions on links in the DG tree occur over links whose end nodes are separated by a relatively smaller distance compared to the links of DG trees formed using other criterion like the predicted expiration time of a link (Meghanathan, 2014).

4. Simulations

4.1 RDD Mobility Profile

The RDD moves randomly within the network with a velocity that is uniform-randomly selected from the range $[0...v_{RDD}]$. To generate the mobility profile of the RDD for a particular v_{RDD} , we let the RDD to start moving from a location that is uniform-randomly selected within the network boundary $[0...XMAX][0...YMAX]$ to another uniform-randomly chosen location from within the network boundary. The RDD travels in a straight line from the starting location to the chosen destination location. After reaching the selected location, the RDD repeats the above procedure by choosing a random location and a random velocity to move. As part of the mobility profile of the RDD, we store the X and Y coordinates of the locations where the RDD changes direction and moves with a different velocity as well as store the time instants the RDD was at these locations. Using these time instants and locations, we can determine the X and Y coordinates of the RDD at any time instant using the interpolation formula shown below in Figure 1. Figure 1 illustrates the methodology used to determine the exact location of the RDD at any time instant t wherein $t_i \leq t \leq t_j$, i.e., the RDD changes directions at time instant t_i - location (X_i, Y_i) and moves in a straight line to location (X_j, Y_j) at time instant t_j . Since the RDD travels in a straight line from time instant t_i to t_j , we can determine the fraction f with which the time instant t divides the line, and use this fraction f to determine the X and Y co-ordinates of the RDD at time instant t according to the interpolation formula shown in Figure 1. Figure 2 shows a sample screenshot of the predicted and exact locations of the RDD during a simulation run for a particular set of operating conditions (the coordinates of the X and Y axes range from 0 to 100).

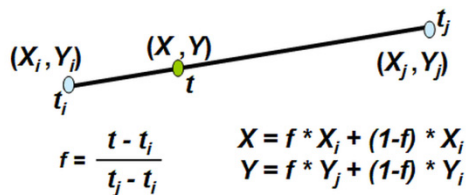


Figure 1. Procedure to determine the exact location of the RDD at any time instant

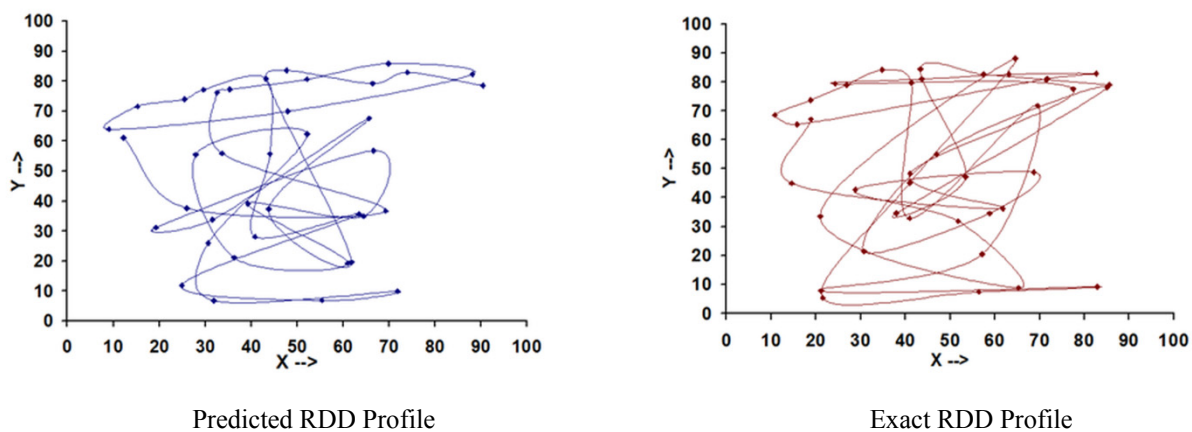


Figure 2. A sample screenshot of the predicted vs. exact locations of the RDD during a simulation run

4.2 Energy Consumption Model

The energy consumption model used is a first-order radio model (Rappaport, 2002) that has been used in several of the previous work in the literature (e.g., Lindsey et. al., 2002; Han et. al., 2011). According to this model, the energy consumed at a sensor node to transmit a k -bit message over a distance d is given by: $E_{TX}(k, d) = E_{elec} * k + \epsilon_{amp} * k * d^2$, where $E_{elec} = 50$ nJ/bit is the energy lost to run the radio transmitter or receiver circuitry and $\epsilon_{amp} = 100$ pJ/bit/m² is the energy lost to run the transmitter amplifier. The energy lost at a sensor node to broadcast a k -bit message to all its neighbors within the transmission range TR is simply given by $E_{TX}(k, TR)$. The energy consumed at a sensor node to receive a k -bit message is $E_{RX}(k) = E_{elec} * k$. The total energy consumed at a sensor node to receive k -bit broadcast messages transmitted by all of its n -neighbors is simply given by $n * E_{RX}(k)$. The energy lost at a sensor node is the sum of the energy lost due to the broadcasting and reception of the TREE-CONSTRUCT messages during the DG tree construction and refresh phases as well as the energy lost to

transmit and receive the QUERY DATA messages for every data reporting time period. A sensor node is assumed to lose 5 nJ/bit to sense the signals in its neighborhood and quantify them to data.

4.3 Network and Data Aggregation Parameters

The simulations are conducted in a discrete-event simulator developed by the authors in Java. The MAC layer is assumed to be ideal and devoid of any collisions. The sensor nodes are assumed to be both TDMA (Time Division Multiple Access) and CDMA (Code Division Multiple Access)-enabled. During the data gathering process, the upstream node schedules a time for the downstream nodes to transmit the completed QUERY DATA message and each downstream node adheres to the schedule. As each intermediate node could communicate with its downstream nodes using a unique CDMA code, the intermediate nodes at a particular level could communicate in parallel with their downstream nodes at the subsequent level.

The number of nodes in the network is set to be 100; the network area is 100m x 100m; the density of the network is varied by operating the nodes at transmission ranges of 25m (moderate density) and 40m (high density). As the connectivity of the network is likely to be less than 100% for transmission ranges less than 25m, we do not employ transmission range lower than 25m. The maximum velocity of the RDD is varied from 2 m/s (low) to 5 m/s (moderate) and 10 m/s (high speed). The sampling time period for the leader node to transmit the QUERY DATA across the DG tree and for the sensor nodes to respond with their sum of the signal strength values is varied from 1 seconds to 200 seconds in this order: 1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 120, 140, 170 and 200 seconds. The DG tree is refreshed for every 300 seconds to avoid frequent energy loss due to broadcasting and reception of the control messages. The number of suspect sensor nodes (nodes that report relatively larger values for the sum of the strength of the signals sensed) is fixed at 5. The values reported in the performance figures 3 and 4 are those averaged from 20 runs of the simulations for each of the scenarios representing the various combinations of values corresponding to the maximum RDD velocity, sampling time period and the transmission range of the sensor nodes. The initial energy of all the sensor nodes is set to 0.25J. For comparison with a benchmark, we also run the simulations (for the above combinations of simulation parameters) considering all the sensor nodes to have unlimited energy (no energy constraints); such simulations would help us obtain the minimum possible median distance error values obtainable when the sensor nodes have unlimited energy for the particular operating scenario.

4.4 Performance Metrics

The following performance metrics are evaluated for the simulations: (1) *Median Distance Error* - We measure the distance between the predicted X and Y coordinates of the RDD at time instants corresponding to the middle of the sampling periods and the exact X and Y coordinates of the RDD at these time instants, and refer to it as the distance error. We store the distance error values calculated across all the time instants of the different simulation runs for a particular scenario and find the median of these values. (2) *Node Lifetime* - The time of first node failure, averaged across all the simulation runs for a particular scenario. (3) *Network Lifetime* - The time of network disconnection due to the failure of one or more sensor nodes (otherwise, the network would have stayed connected if all the nodes have had positive residual energy).

The median distance error is the primary metric used to evaluate the effectiveness of the proposed mobile target tracking algorithm. The smaller the median distance error, the more effective is the algorithm. Based on the simulation results presented in Figure 3, we observe the median distance error to be dependent on the following key operating parameters - the RDD velocity, the density of the network (controlled directly by varying the transmission range of the sensor nodes and indirectly by operating the network with limited energy per node triggering node failures) and the sampling time period. For a fixed value of three of these parameters, as we increase the transmission range (and hence the sensing range) of the sensor nodes, we observe the median distance error observed when operating the network with energy constraints to be relatively close to that of the minimum possible median distance error for the particular operating scenario. On the other hand, for a fixed network density and sampling time period, as we increase the RDD velocity, the median distance error increases for both the energy-constrained and energy-unconstrained scenarios; of course, for larger RDD velocity values, the median distance error for the energy-constrained scenario increases significantly when sufficient sensor nodes are not available to track the fast moving RDD (occurs when the sampling time period is too large, resulting in fewer samples being collected; but these are not sufficient enough to accurately track the RDD). Thus, to detect a fast moving RDD, it would be wise to operate with a smaller sampling time period (collect frequent samples) even if it comes at the cost of running out the sensor nodes of the available energy (smaller node lifetime and network lifetime).

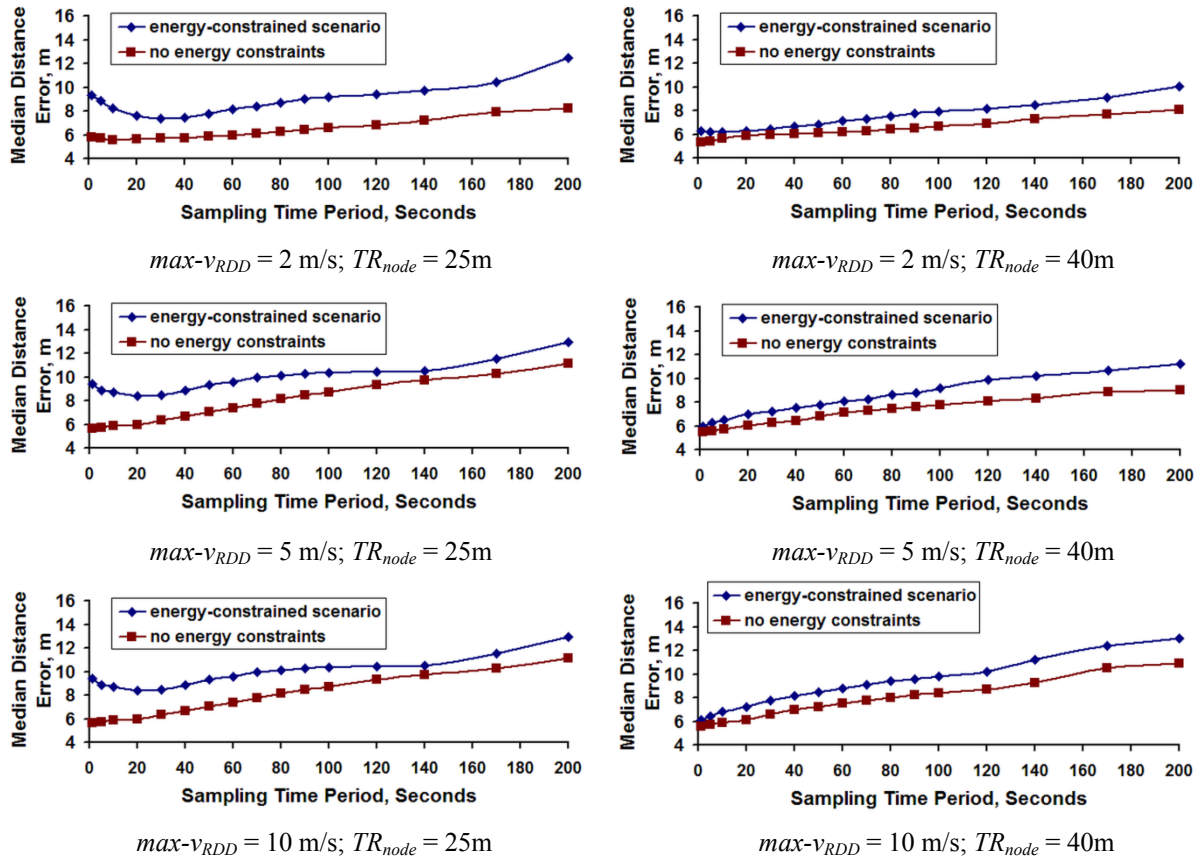


Figure 3. Median Distance Error Values

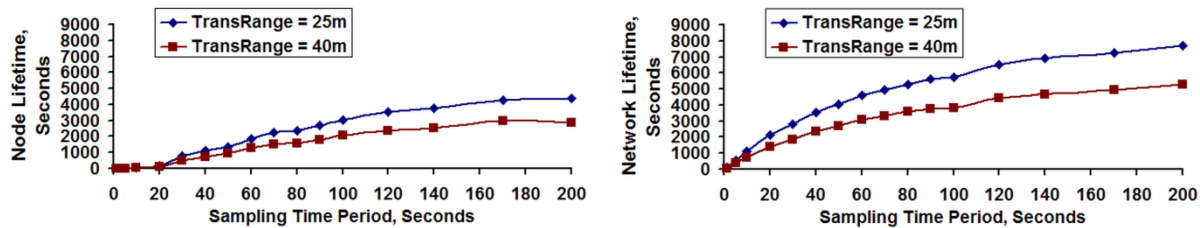


Figure 4. Node Lifetime and Network Lifetime

When the network is operated without any energy constraints, the smaller the sampling time period, the smaller the median distance error as the location of the RDD could be more accurately tracked with a larger number of samples collected from the network. However, when operated under energy constraints, we observe that sampling at a high frequency (sampling time period of 1 and 5 seconds) could sometimes lead to slightly high median distance error values, especially when the RDD moves relatively slowly ($max-v_{RDD} = 2$ and 5 m/s); this could be attributed to the nodes quickly running out of battery charge and the RDD has to be tracked with a reduced number of sensor nodes after a while (with a reduced network density, the predictions tend to become less accurate, as only fewer sensor nodes are likely to be in the vicinity of the RDD). Thus, when the RDD moves relatively slower, the network needs to be operated at slightly higher sampling time periods of 10-40 seconds to ward off unnecessary node failures.

As can be observed in Figure 3, for fixed value of the operating parameters, the larger the transmission range per node, the lower the median distance error under energy constrained scenarios and closer it is to the minimum possible median distance error for the particular operating condition; however the tradeoff is a lower node lifetime and network lifetime (Figure 4). The difference in the magnitude of the network lifetime and node lifetime increases with increase in the sampling time period - frequent sampling is an overkill for the sensor nodes and quickly drains their battery charge. However, moderate sampling time period values of 10-40 seconds

contribute to a lower median distance error as well as a relatively larger network lifetime. Also, by operating the network at a larger transmission range per node, we could effectively track the moving RDD (the reduction in the median distance error under energy constrained scenarios is more significant compared to the reduction in the minimum median distance error when the network is operated without any energy constraint under identical conditions) - but, this comes at a higher penalty of lower node lifetime and network lifetime (for a fixed sampling time period, the difference between the node lifetime and network lifetime is higher when operated at a lower transmission range per node).

An interesting and significant observation is that for low-moderate mobility of the RDD ($max-v_{RDD} = 2$ and 5 m/s), the median distance error values (especially those for energy constrained scenarios) predominantly show a convex pattern of increase with increase in the sampling time period; on the other hand, for high mobility of the RDD ($max-v_{RDD} = 10$ m/s), the median distance error values show a concave nature of increase with increase in the transmission range per node. This is a positive characteristic of the proposed mobile target tracking algorithm indicating that the median distance error values do not tend to increase at a faster rate with increase in RDD velocity (the rate of increase actually slows down). Looking at Figure 4 from a similar point of view, we observe both the node lifetime and network lifetime to display a concave pattern of increase with increase in the sampling time period, illustrating that the node lifetime and network lifetime cannot be significantly increased with increase in the sampling time period (because the DG tree has to be still regularly refreshed to rotate the role of the leader node) that corresponds to an increase in the median distance error too. Thus, overall, it would be more wiser to operate the network with moderate values of the sampling time period and sustain a lower median distance error for an appreciable duration of the lifetime of the network.

5. Related Work and Our Contributions

Mobile target tracking using sensor networks has been actively studied in the past few years. Most of the work (e.g., Chin et al., 2010) on target tracking require target localization using techniques such as triangulation and tri-lateration (Han et al., 2011), in addition to requiring the sensor nodes to be able to determine their own location. Also, some algorithms (e.g., Amundson et al., 2007) required the sensor nodes and the target to be synchronized with respect to time, especially for heterogeneous sensor networks. The advantage with our proposed algorithm is that it does not require target localization, and the target and the sensor node do not need to be tightly synchronized with respect to time. Moreover, our algorithm can work for both homogeneous networks (all nodes operating under identical transmission/sensing range) and heterogeneous networks (the transmission/sensing range could be different across nodes). We adopt an effective approach (novel contribution to the literature) of accumulating the strength of the sensed signals locally at the sensor nodes and then averaging out (at the sink) the X and Y coordinates of the sensor nodes that report relatively larger values for the sum of the signal strengths during the sampling time period. Also, most of the work (e.g., Zhang & Cao, 2004) require appreciable coordination between the sensor nodes with respect to data fusion as well as to track the mobile target in a neighborhood. Our algorithm does not require much coordination among the sensor nodes other than just forwarding the QUERY DATA message along the data gathering tree in both directions; the intermediate sensor nodes do not require to run any complicated algorithm for data fusion. Some of the tracking algorithms require the formation of clusters and the election of cluster head as a pre-requisite for tracking (e.g., Goshorn et al., 2007). Some other algorithms model the sensor network to be hierarchical in nature (e.g., Wang et al., 2008). Our algorithm can run for both flat and hierarchical sensor networks and it only requires a leader node for the data gathering tree; the leader node is simply selected by the sink and the data gathering tree is also regularly refreshed to avoid any particular sensor node from losing too much of energy for serving as the leader node of the DG tree.

Liu et al. (2007) propose a Bayesian statistics based method for tracking a nuclear threat using a network of static and mobile sensor nodes; however, for effective tracking - this approach requires prior knowledge about the distribution of the background radiation (to distinguish it from the radiations detected by a particular sensor node) as well as requires strategic placement of the sensor nodes. Our proposed tracking method does not require any prior knowledge of the distribution of the background radiation as the sum of the signal strength values are collected from each sensor node (through a QUERY DATA message) for every sampling time period. Also, the sensor nodes could be simply uniform-randomly distributed throughout the network (i.e., a sensor node could be located anywhere within the network with equal chance). Likewise, there are approaches (e.g., Rassam et al., 2013) that use anomaly detection algorithms to distinguish the signal data emanating from a particular sensor node from that of others; on the other hand, due to the use of the sum of the signal strengths over the sampling time period, our premise is that the differences in the signal strengths sensed by the sensor nodes (due to the presence or absence of the RDD in their respective neighborhood) will magnify itself when we sum the signal

strengths over a period of time. Chandy et al. (2013) present an analytical model for tracking one or more stationary radioactive sources that are tracked using a network of static but smart radioactive sensor nodes that can distinguish the radiations from background noise. However, the method cannot be used to trace the trajectory of a mobile RDD.

Ours is the first such algorithm designed to track and trace the trajectory for a mobile radioactive device without using any complicated algorithm for decoding the signals sensed in the environment. The idea of summing up the strengths of the signals sensed over a sampling time period helps to effectively differentiate sensor nodes that are in the vicinity of a mobile RDD from those that are not. Though designed to track mobile targets, our algorithm can even more easily and effectively track a static target. Above all, the algorithm runs in real-time and it does not require a huge volume of data to be collected prior to making any prediction. The algorithm simply works based on the data collected during the sampling time period and could predict the location of the RDD when it might be still in the location's vicinity.

6. Conclusions

The high-level contribution of this paper is the design and development of an efficient algorithm to target mobile radioactive devices (RDD) using a wireless sensor network. The tracking algorithm is data-centric and lets each sensor node to simply sum the strength of the radioactive signals sensed in the neighborhood (without any need to distinguish the background signals and the signals emanating from a radioactive target moving in the neighborhood) and report the summed value over a sampling time period to the sink via a data gathering tree. The hypothesis is that the sensor nodes that report relatively larger value for the sum of the sensed signals (referred to as the suspect sensor nodes) are likely to be in the vicinity of the mobile target during the sampling time period - accordingly, we use the average of the X and Y coordinates of the suspect sensor nodes as the predicted locations of the RDD.

We run extensive simulations to validate our hypothesis and we observe our hypothesis to be valid under all operating conditions. The median of the distance error values reported for both energy-unconstrained and energy-constrained scenarios are well within the sensing range of the sensor nodes; this implies that if at least one sensor node is in the vicinity of a mobile target in a particular region, that sensor node will be able to effectively track the presence of the target over a period of time (as the sum of the signal strength values is likely to be higher than those reported by the peer sensor nodes). We also extensively evaluate the impact of the different operating parameters (RDD velocity, sampling time period, transmission range/sensing range of the sensor nodes and the energy available at the sensor nodes) on the median distance error. Simulation results indicate that it is imperative to operate the network at low-moderate values of the sampling time period to obtain a lower median distance error for an appreciably longer network lifetime, even if the RDD velocity gets higher. In this paper, we have restricted our simulations to a heterogeneous network of sensor nodes; as part of future work, we will conduct simulations of the proposed RDD tracking algorithm for a network of heterogeneous sensor nodes.

Acknowledgments

This research is funded by the Massie Chair grant (#: DE-NA0000654) at Jackson State University.

References

- Amundson, I., Kushwaha, M., Kusy, B., Volgyesi, P., Simon, G., Koutsoukos, X., & Ledeczi, A. (2007). *Time Synchronization for Multi-Modal Target Tracking in Heterogeneous Sensor Networks*. Paper presented at the Workshop on Networked Distributed Systems for Intelligent Sensing and Control, Kalamata, Greece.
- Chandy, M., Pilotto, C., & McLean, R. (2008). *Networked Sensing Systems for Detecting People Carrying Radioactive Material*. Paper presented at the 5th International Conference on Networked Sensing Systems, Kanazawa, Japan. <http://dx.doi.org/10.1109/INSS.2008.4610916>
- Chin, J. C., Rao, N. S. V., Yau, D. K. Y., Shankar, M., Yang, Y., Hou, J. C., Srivathsan, S., & Iyengar, S. (2010). Identification of Low-Level Point Radioactive Sources using a Sensor Network. *ACM Transactions on Sensor Networks*, 7(3), Article 21. <http://dx.doi.org/10.1145/1807048.1807050>
- Goshorn, R., Goshorn, J., Goshorn, D., & Aghajan, H. (2007). *Architecture for Cluster-based Automated Surveillance Network for Detecting and Tracking Multiple Persons*. Paper presented at the ACM/IEEE International Conference on Distributed Cameras, Vienna, Austria. <http://dx.doi.org/10.1109/ICDSC.2007.4357527>
- Han, G., Xu, H., Duong, T. Q., Jiang, J., & Hara, T. (2011). Localization Algorithms of Wireless Sensor

- Networks: A Survey. *Telecommunication Systems*, 52(4), 2419-2436. <http://dx.doi.org/10.1007/s11235-011-9564-7>
- Heinzelman, W., Chandrakasan, A., & Balakrishnan, H. (2004). *Energy-Efficient Communication Protocols for Wireless Microsensor Networks*. Paper presented at the Hawaiian International Conference on Systems Science, USA. <http://dx.doi.org/10.1109/HICSS.2000.926982>
- Lindsey, S., Raghavendra, C., & Sivalingam, K. M. (2002). Data Gathering Algorithms in Sensor Networks using Energy Metrics. *IEEE Transactions on Parallel and Distributed Systems*, 13(9), 924-935. <http://dx.doi.org/10.1109/TPDS.2002.1036066>
- Liu, A. H., Bunn, J. J., & Chandy, K. M. (2011). *Sensor Networks for the Detection and Tracking of Radiation and Other Threats in Cities*. Paper presented at the 10th International Conference on Information Processing in Sensor Networks, Chicago, IL, USA.
- Meghanathan, N. (2012). A Comprehensive Review and Performance Analysis of Data Gathering Algorithms for Wireless Sensor Networks. *International Journal of Interdisciplinary Telecommunications and Networking*, 4(2), 1-29. <http://dx.doi.org/10.4018/jitn.2012040101>
- Meghanathan, N. (2014). Stability-based and Energy-Efficient Distributed Data Gathering Algorithms for Mobile Sensor Networks. *Ad hoc Networks*, 19, 111-131. <http://dx.doi.org/10.1016/j.adhoc.2014.02.007>
- Rappaport, T. S. (2002). *Wireless Communications: Principles and Practice* (2nd ed.) Prentice Hall.
- Rassam, M. A., Zainal, A., & Maarof, M. A. (2013). Advancements of Data Anomaly Detection Research in Wireless Sensor Networks: A Survey and Open Issues. *Sensors*, 13, 10088-10122.
- Wang, Z., Li, H., Shen, X., Sun, X., & Wang, Z. (2008). *Tracking and Predicting Moving Targets in Hierarchical Sensor Networks*. Paper presented at the IEEE International Conference on Networking, Sensing and Control, Sanya, China. <http://dx.doi.org/10.1109/ICNSC.2008.4525393>
- Zhang, H., & Hou, J. C. (2005). Maintaining Sensing Coverage and Connectivity in Large Sensor Networks. *Wireless Ad hoc and Sensor Networks: An International Journal*, 1(1-2), 89-123.
- Zhang, W., & Cao, G. (2004). DCTC: Dynamic Convoy Tree-based Collaboration for Target Tracking in Sensor Networks. *IEEE Transactions on Wireless Communications*, 3(5), 1689-1701. <http://dx.doi.org/10.1109/TWC.2004.833443>

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).