Constructing A Shortest Path Overhearing Tree With Maximum Lifetime In WSNs

Wael Y. Alghamdi, Hui Wu, Wenguang Zheng, Salil S. Kanhere

School of Computer Science and Engineering
The University of New South Wales
Sydney, Australia

Email: {waela, huiw, wenguangz, salilk}@cse.unsw.edu.au

Abstract—Secure data collection is an important problem in wireless sensor networks. Different approaches have been proposed. One of them is overhearing. We investigate the problem of constructing a shortest path overhearing tree with the maximum lifetime. We propose three approaches. The first one is a polynomial-time heuristic. The second one uses ILP (Integer Linear Programming) to iteratively find a monitoring node and a parent for each sensor node. The last one optimally solves the problem by using MINLP (Mixed-Integer Non-Linear Programming). We have implemented the three approaches using MIDACO solver and MATLAB Intlinprog, and performed extensive simulations using NS2.35. The simulation results show that the average lifetime of all the network instances achieved by the heuristic approach is 85.69% of that achieved by the ILP-based approach and 81.05% of that obtained by the MINLP-based approach, and the performance of the ILP-based approach is almost equivalent to that of the MINLP-based approach.

Keywords—Overhearing; routing tree; network lifetime; secure data collection; integer linear programming; mixed-integer non-linear programming

I. INTRODUCTION

A WSN (Wireless Sensor Network) consists of a large number of autonomous sensors nodes and a single or multiple base stations. Each sensor node delivers the data sensed from the physical environment to its designated base station. Typically, sensor nodes are battery powered. Most of the energy of a sensor node is consumed by communication. In order to save energy, the transmit power of each sensor node is kept low, leading to a short transmission range. Thus, data collection is performed in a multi-hop way.

In WSNs, various attacks may exist. Among them are selective forwarding attacks and modification attacks. In the selective forwarding attacks, a malicious sensor node may deliberately drop some packets received from other sensor nodes, resulting in packet loss. In the modification attacks, a malicious sensor node may modify some packets received from other sensor nodes and forward the incorrect packets to the base station. In order to ensure that the data sensed by each sensor are delivered to the base station correctly, the protocols for secure routing are required.

The secure and reliable data collection problems in WSNs have been extensively investigated. Many approaches are based on the overhearing technique [1]. When a sensor node receives a packet and forwards it to another sensor node, a third sensor node will overhear the packet reception and transmission. Therefore, the overhearing technique can be used to detect the modification attacks and the selective forwarding attacks.

In this paper, we investigate the problem of constructing a routing and overhearing topology with the maximum network lifetime in a WSN with a single base station. Specifically, we construct a shortest path tree for routing and select a monitoring sensor node for each sensor node such that the network lifetime is maximized. For each sensor node $v_i$, its monitoring sensor node overhears the reception and transmission of each packet that the parent of $v_i$ receives from $v_i$. The network lifetime is defined as the time when the first sensor node depletes its energy. Even without considering overhearing, the problem of constructing a shortest path tree with the maximum lifetime is NP-Complete [2].

We make the following major contributions.

- We propose a polynomial-time heuristic approach, an ILP-based approach, and a MINLP-based approach to the problem of constructing a shortest path overhearing tree with the maximum network lifetime. To the best of our knowledge, our work is the first attempt to construct such an overhearing topology with the maximum lifetime for security purposes.
- We have implemented our approaches using MIDACO solver and MATLAB Intlinprog, and performed extensive simulations on 150 network instances with three different distributions, namely, uniform, grid, and random distributions. The simulation results show that the average lifetime of all the network instances achieved by the heuristic approach is 85.69% of that achieved by the ILP-based approach and 81.05% of that obtained by the MINLP-based approach, and the average network lifetime achieved by the ILP-based approach is 96.2% of that obtained by the MINLP-based approach.

The rest of the paper is organized as follows. Section II gives a brief survey of the related work. Section III provides the network model and definitions. Section IV describes our heuristic approach in detail. Section V presents our MINLP-based approach. Section VI proposes our ILP-based approach. Section VII shows the simulation results and analyses. Lastly, Section VIII concludes this paper.
II. RELATED WORK

The overhearing technique has been widely used to improve the security and the reliability of data collection in WSNs. The problem of constructing a routing tree with the maximum network lifetime has also been extensively investigated. Next, we give a survey of the major work related to secure and reliable data collection by using overhearing and the lifetime-aware routing tree construction problem in WSNs.

A. Overhearing-Based Secure and Reliable Data Collection

[1] uses an overhearing technique to detect modification attacks in WSNs. By the overhearing technique, a committee structure is constructed for each sensor node. The committee structure includes several committee sets, and each committee set is designed for a specific communication link. Due to the microwave nature of the wireless channel, neighbouring sensor nodes within a sender’s radio range can overhear the packet the sender is transmitting. Therefore, each packet can be examined by the sensor nodes of the committee set during forwarding. If a packet is modified by a malicious node, the committee will detect the anomaly. [3] provides several drawbacks of the security mechanism used by [1]. Firstly, there is no mechanism implemented for neighbour nodes authentication within the construction phase. Hence, the malicious node may penetrate into the network to send fake information about their neighbours and contribute in voting. Secondly, in the last phase, the proposed mechanism does not isolate the malicious nodes from the network.

[4] proposes an efficient overhearing-based reliable transfer protocol for WSNs. The monitoring sensor node overhears a packet being transmitted by the monitored sensor node to determine whether the monitored node is malicious or not. To make such a decision, a reputation value is calculated for each sensor node in the designated area. The sensor node with a reputation value below the threshold is considered as a malicious sensor node.

[5] proposes an efficient, reliable transfer protocol for WSNs by introducing implicit and selective acknowledgement. The selective acknowledgement mechanism is executed by comparing the current path reliability and the base reliability. Overhearing is also used in the implicit acknowledgement mechanism.

Several approaches [6]–[9] use the overhearing technique to avoid redundant information to be transmitted towards the base station, improving the energy efficiency.

[6] proposes an energy-efficient data transmission reduction approach for periodic data gathering in WSNs by using the overhearing technique. In this approach, each sensor node in a WSN autonomously determines whether its own reading is redundant or not by using the overheard packets transmitted by its neighbors. If the reading is determined as redundant, the sensor node stops transmitting it. [8] extends this approach by proposing an overhearing-based data aggregation method using spatial and temporal interpolations.

[7] presents OBMAC, an enhancement for MAC protocol based on the overhearing technique in WSNs. The objective of the proposed protocol is to reduce the number of redundant packets using the overhearing technique. In OBMAC, every sensor node verifies each overheard packet and compares it to its own in order to avoid transmitting the same information to the base station. The notion of influential range is used to improve the efficiency of OBMAC.

[9] presents a method for evaluating the approach proposed in [6] by using a practical model of lossy links. The evaluation results show that the proposed approach suppresses data transmissions and reduces total energy consumption even in a lossy environment.

[10] investigates the problem of improving the data persistence. It introduces a distributed scheme based on LT (Luby Transform)-codes and an overhearing technique. In the proposed scheme, each sensor node uses overhearing to check if a packet has been transmitted by one of its neighbors. When a sensor node needs to transmit a packet, it randomly chooses one of its neighbors that does not transmit the packet as the receiver. Each sensor node computes a key parameter of LT codes by using some properties of the packet transmission mechanism, and then stores the data accordingly. After the process of storage is finished, a collector will recover all the data by visiting a small subset of sensor nodes.

B. Lifetime-Aware Routing Trees

A routing tree of a WSN with a single base station is a spanning tree rooted at the base station. Each sensor node sends its own data and the data received from its children to its parent. Since each sensor node is typically battery-powered, it is important to construct a routing tree such that the network lifetime is maximized. Many approaches to the lifetime-aware routing tree construction problem have been proposed.

[11] shows that the problem of constructing a spanning tree with the maximum lifetime is NP-complete and proposes a polynomial-time approximation algorithm. The approximation algorithm starts with an arbitrary tree and iteratively reduces the loads of bottleneck nodes. [12] studies an on-line data gathering problem, and proves that the problem is NP-complete. It presents a generic cost model of energy consumption for data gathering queries and several heuristics.

[13] and [14] take into account the remaining energy and the load of each node, and propose top-level load balancing algorithms with dynamic modifications. [15] constructs a multi-tree topology to allow more choices for the next hop when routing messages. [16] proposes a distributed probabilistic load-balancing converge cast tree algorithm to
address the heterogeneity issues in terms of nodal traffic burden and residual energy by dynamically forming converge cast routing trees.

[17] proposes a new weighted path cost function improved from the shortest path tree approach. In this approach, links are assigned weights according to their path lengths to the root, and those close to the root have larger weights. By balancing loads according to the link weights, this approach increases the network lifetime compared with those randomly constructed shortest path trees.

[2] investigates the problem of finding a shortest path aggregation tree with the maximum lifetime in a WSN. The proposed algorithm first builds a fat tree which contains all the shortest path trees. Then, it converts the problem into a sequence of semi-matching problems each considering two adjacent levels of the fat tree, and solves each semi-matching problem by using the min-cost max-flow approach in polynomial time. [2] also proposes a distributed algorithm for constructing a maximum lifetime shortest path aggregation tree, and proves that if no data aggregation is performed, it is NP-complete to construct a shortest path tree with the maximum lifetime.

[18] investigates the lifetime-aware data collection problem without data aggregation, and proposes an approximation algorithm for constructing a routing tree with the maximum network lifetime. The approximation algorithm iteratively transfers some of the descendants of the node with the largest weight to a node with a smaller weight, and stops when no more descendants of a bottleneck node can be transferred.

[19] studies the load balance problem in a grid topology. It focuses on the energy consumptions of the nodes which can communicate with the base station directly. Firstly, the algorithm selects the most lightly loaded and most confined branches for growth. Secondly, it selects the heaviest nodes with the maximum growth space. After establishing a loosely balanced tree, the algorithm re-balances the tree by moving nodes from the heaviest loaded branches to more lightly loaded neighbouring branches. The simulation results show that the routing trees constructed by their algorithm are more balanced than the shortest path tree constructed by Dijkstra’s algorithm.

[20] investigates the problem of network lifetime maximization of WSNs in the context of data collection trees. It proposes an efficient algorithm, called Randomized Switching for Maximizing Lifetime (RaSMaLai) that aims at maximizing the lifetime of WSNs through load balancing with a low time complexity, and a distributed version of the algorithm.

[21] investigates the problem of lifetime and latency-aware data collection in WSNs with one base station. It proposes a new routing structure, namely $k$-tree, and a distributed algorithm for constructing a lifetime-aware $k$-tree. A unique feature of the $k$-tree is that it provides the maximum latency guarantee for data collection.

III. NETWORK MODEL AND DEFINITIONS

The target WSN consists of a set $V = \{v_1, v_2, \cdots , v_n\}$ of $n$ static sensors. There is only one base station. Each sensor generates one packet of data per unit time and sends the packet to the base station without performing any data aggregation. All the sensor nodes are identical with the same transmission range and the same initial energy level.

We use an undirected graph, named as connectivity graph, to represents the connectivity between sensor nodes in the WSN. The connectivity graph $G$ is defined as follows: $G = (V \cup \{ BS \}, E)$, where, $BS$ is the base station, and $E = \{(v_i, v_j) : v_i, v_j \in V \cup \{ BS \} \text{ and } v_i \text{ and } v_j \text{ can communicate with each other directly}\}$. A communication link between two sensor nodes indicates that the two sensor nodes can communicate with each other. We assume that the connectivity graph is connected. The base station can collect the connectivity graph from all the sensor nodes.

Some sensor nodes may be compromised. A compromised sensor node is called a malicious sensor node. A malicious sensor node may drop, or modify the packets it receives from other sensor nodes.

Given the connectivity graph $G = (V,E)$ of a target WSN, the problem we investigate is to construct a shortest path routing tree rooted at the base station and assign a monitoring sensor node to each sensor node such that the network lifetime is maximized. Hereinafter, a monitoring sensor node is called a monitor. A monitor $v_i$ of a sensor node $v_k$ is used to detect if the data sent by $v_k$ to its parent $v_j$ will be forwarded correctly by $v_j$ to its parent in the shortest path tree. Therefore, the monitor $v_i$ needs to overhear the transmission of each packet $v_k$ sends to its parent $v_j$ and the forwarding of each packet $v_j$ receives from $v_k$. If $v_j$ drops or modifies any packet received from $v_k$, $v_i$ will detect it and report it to the base station.

For each sensor node, its parent and monitor must satisfy the following requirements:

1) There is a communication link between the parent and the monitor.
2) There is a communication link between the monitor and the sensor node.
3) There is a communication link between the parent and the sensor node.
4) The parent and the monitor have the same depth in the routing tree.

When constructing a shortest path tree, we mainly consider the energy consumption of data receptions and transmissions for each sensor node. The energy consumption of listening and communication session setup is ignored. For each sensor node, $\alpha$ is the energy consumed to receive one packet, and $\beta$ is the energy consumed to transmit one packet. Given a set $S$, $|S|$ denotes the size of $S$. 5860
IV. HEURISTIC APPROACH

Our heuristic approach consists of three phases, namely, partitioning phase, initial monitor and parent selection phase, and energy balancing phase.

In the partitioning phase, our heuristic approach partitions all the sensor nodes into $m$ disjoint groups $C_i(i = 1, 2, \ldots, m)$ such that the shortest path length between each sensor node in the group $C_i$ to the base station in the connectivity graph is equal to $i$. In the initial monitor and parent selection phase, for each group $C_i(i = m, m - 1, \ldots, 2)$, our heuristic approach assigns a parent and a monitor from the group $C_{i-1}$ to each sensor node in the group $C_i$, aiming at minimizing the maximum energy consumption of all the individual sensor nodes in the group $C_{i-1}$. In the energy balancing phase, our heuristic approach performs the monitor and parent adjustment for each group $C_i(i = m, m - 1, \ldots, 2)$ such that all the sensor nodes in each group almost consume the same amount of energy per unit time.

Next, we describe how to assign a monitor and a parent from $C_{t-1}$ to each sensor node in $C_t$ for $t = 1, 2, \ldots, n$. Recall that for each sensor node, its parent and its monitor must satisfy the four requirements stated in the previous section.

For each sensor node $v_i \in C_t$, we use a 3-tuple $(v_i, v_j, v_k)(j < k)$ to uniquely represent a triangle formed by $v_i$, $v_j$ and $v_k$ in the connectivity graph, where $v_j$ and $v_k$ are in $C_{t-1}$.

For each sensor node $v_i \in C_t$, we introduce the following notations:

- $S_i$: a set of all the triangles in the connectivity graph each of which contains $v_i$ and two sensor nodes in $C_{t-1}$. Notice that if $S_i$ is empty, no shortest path overhearing tree exists.
- $P_i$: a set of all the sensor nodes in $C_{t-1}$ each of which is adjacent to $v_i$ in the connectivity graph and is not selected as the monitor or the parent of $v_i$.
- $M_i$: a set of all the sensor nodes in $C_{t+1}$ for which $v_i$ is the monitor. Initially, $M_i$ is $\emptyset$. Each time when $v_i$ is selected as the monitor of a sensor node $v_j$ in $C_{t+1}$, $M_i$ is updated as follows: $M_i = M_i \cup \{v_j\}$.
- $CH_i$: a set of all the children of $v_i$ in the partial shortest path tree rooted at $v_i$ currently constructed.
- $\text{subtree} - \text{size}(v_i)$: the number of sensor nodes in the subtree rooted at the sensor node $v_i$ constructed so far.
- $e_i$: the energy consumed by $v_i$ per unit time under the current assignment of monitors and parents. Initially, $e_i$ is equal to $\beta$. Each time when $v_i$ is selected as the monitor of a sensor node $v_j$, $e_i$ is updated as follows: $e_i = e_i + 2(\text{tree} - \text{size}(v_j))\alpha$. Each time when $v_i$ is selected as the parent of a sensor node $v_j$, $e_i$ is updated as follows: $e_i = e_i + \text{tree} - \text{size}(v_j)\alpha + \text{tree} - \text{size}(v_j)\beta$.

We define two types of priorities: a priority for each sensor node in $C_t$ and a priority for each sensor node in $C_{t-1}$. The priority $P_1(v_i)$ of each sensor node $v_i$ in $C_t$ is a 3-tuple defined as follows:

$$P_1(v_i) = (|S_i|, |P_i|, 1/\text{tree} - \text{size}(v_i))$$  \hspace{1cm} (1)

The priority $P_2(v_j)$ of each sensor node $v_j$ in $C_{t-1}$ is a 2-tuple defined as follows:

$$P_2(v_j) = (e_j, |M_j \cup CH_j|)$$  \hspace{1cm} (2)

For both types of priorities, a smaller tuple implies a higher priority. Note that the priority of each sensor node may be changed dynamically during the initial monitor and parent selection phase.

For each group $C_i(i = m, m - 1, \ldots, 2)$, the initial monitor and parent selection algorithm works as follows:

1) For each sensor node $v_i \in C_t$, compute $S_i$ and $P_i$.
2) For each sensor node $v_j \in C_{t-1}$, do the following.
   a) Set $CH_j$ and $M_j$ to $\emptyset$.
   b) Set $e_j$ to $\beta$.
3) For each sensor node $v_i \in C_t$, compute the priority $P_1(v_i)$.
4) For each sensor node $v_j \in C_{t-1}$, compute the priority $P_2(v_j)$.
5) $A = C_t$.
6) Repeat the following until $A$ is $\emptyset$.
   a) Select a sensor node $v_s$ with the highest priority from $A$.
   b) If $S_s$ is equal to $\emptyset$, no parent and monitor exist for $v_s$, and the algorithm terminates.
   c) Assign a 2-tuple rank $R(X)$ to each triangle $X \in S_s$ as follows:
      i) Let $v_j$ and $v_k$ be the two sensor nodes other than $v_s$ in $X$ such that $P_2(v_j) \leq P_2(v_k)$ holds.
      ii) $R(X) = (P_2(v_j), P_2(v_k))$.
   d) Find a triangle $X_{\text{min}}$ in $S_s$ with the smallest rank.
   e) Let $R(X_{\text{min}}) = (P_2(v_s), P_2(v_t))$.
   f) If $e_s < e_t$ holds, do the following.
      i) Select $v_j$ and $v_t$ as the parent and the monitor of $v_s$, respectively.
      ii) $M_s = M_s \cup \{v_j\}$.
      iii) $CH_s = CH_s \cup \{v_j\}$.
      iv) $e_s = e_s + \text{tree} - \text{size}(v_j)\alpha + \text{tree} - \text{size}(v_t)\beta$.
      v) $e_t = e_j + 2(\text{tree} - \text{size}(v_j))\alpha$.
      vi) Re-compute $P_2(v_s)$ and $P_2(v_t)$.
Otherwise, do the following.
   i) Select $v_j$ and $v_s$ as the parent and the monitor of $v_t$, respectively.
   ii) $M_t = M_t \cup \{v_j\}$.
   iii) $CH_t = CH_t \cup \{v_j\}$.
iv) \(e_t = e_t + \text{tree}-\text{size}(v_l)\alpha + \text{tree}-\text{size}(v_i)\beta\).

v) \(e_s = e_s + 2\text{tree} - \text{size}(v_i)\alpha\).

vi) Re-compute \(P_2(v_s)\) and \(P_2(v_t)\).

g) \(A = A - \{v_i\}\).

h) For each sensor node \(v_j \in A\) that is adjacent to \(v_i\) in the connectivity graph, set \(P_j\) to \(P_j - \{v_j\}\).

i) For each sensor node \(v_j \in A\) that is adjacent to \(v_i\) in the connectivity graph, set \(P_j\) to \(P_j - \{v_j\}\).

j) For each sensor node \(v_j \in A\) that is adjacent to \(v_i\) in the connectivity graph, re-compute \(P_1(v_j)\).

After the initial monitor and parent selection phase, the energy balancing phase starts. The energy balancing phase works from the group \(C_{m-1}\) to the group \(C_1\). For each group \(C_l(l = m - 1, m - 2, \cdots, 1)\), the energy balancing algorithm selects a sensor node \(v_i\) with the maximum energy consumption per unit time, and shifts the role of \(v_i\) as the parent or the monitor of a sensor node \(v_j\) to another sensor node with lower energy consumption. In order to find such a sensor node \(v_j\), we define a new rank for each sensor node \(v_r\) in \(M_i \cup CH_i\) as follows:

\[
W(v_r) = (W_1(v_r), W_2(v_r), \cdots, W_N(v_r)) \quad (3)
\]

where \(N\) is equal to \(|C_l|\), and \(W_1(v_r), W_2(v_r), \cdots, W_N(v_r)\) are the energy consumptions per time unit of all the sensor nodes in \(C_l\) sorted in non-increasing order after \(v_r\)'s role as \(v_r\)'s monitor or parent is switched to a candidate sensor node \(v_p\) in \(C_l\).

Let \(v_r\) be a sensor node in \(M_i \cup CH_i\) such that the role of \(v_r\) as the monitor or the parent of \(v_r\) will be replaced by a candidate sensor node \(v_p\). \(v_p\) is found as follows:

1) Let \(B\) be a set of all the triangles of the form \((v_r, v_i, v_s)(i < s)\) or \((v_r, v_i, v_s)(s < i)\).

2) Let \(E\) be a set of all the sensor nodes in \(B\) that are different from \(v_r\) and \(v_i\).

3) \(v_p\) is the sensor node in \(E\) that has the smallest energy consumption per time unit.

If such a sensor node \(v_p\) is not found, \(W(v_r)\) is set to \((+\infty, +\infty, \cdots, +\infty)\). A role switch is allowed only if the switch results in better energy balancing.

The energy balancing algorithm works for each group \(C_l(l = m - 1, m - 2, \cdots, 1)\) as follows.

1) Mark each sensor node in \(C_l\) as switchable.

2) Repeat the following until no sensor node in \(C_l\) is switchable.

a) Let \(e_1, e_2, \cdots, e_N\) be the energy consumptions per time unit of all the sensor nodes in \(C_l\) sorted in non-increasing order.

b) \(W = (e_1, e_2, \cdots, e_N)\).

c) Pick a sensor node \(v_i\) in \(C_l\) that is not marked as unswitchable and has the maximum energy consumption per unit time.

d) For each sensor node \(v_s \in M_i \cup CH_i\), compute \(W(v_s)\).

e) \(T = \min\{|W(v_s) : v_s \in M_i \cup CH_i\}\).

f) If \(T \geq W\) holds, mark \(v_i\) as unswitchable.

g) Otherwise, let \(v_r\) be a sensor node in \(M_i \cup CH_i\) with the smallest rank \(W(v_r)\), and \(v_p\) the sensor node selected to replace \(v_r\)'s role as the parent or the monitor of \(v_r\) when computing \(W(v_r)\). Do the following.

i) Switch \(v_r\)'s role as \(v_r\)'s monitor or parent to \(v_p\).

ii) Re-compute \(e_1, e_2, \cdots, e_N\).

3) If \(l > 1\) holds, for each sensor node \(v_j\) in \(C_{l-1}\), re-calculate \(e_j\) based on the current partial shortest path overhearing tree.

Next, we analyze the time complexity of our heuristic approach. The time complexity is broken down into the following three parts.

1) The partitioning phase. We can use breadth-first search to compute the shortest path length from each sensor node to the base station. Therefore, the time complexity of this phase is \(O(e)\), where \(e\) is the number of edges in the connectivity graph.

2) The initial monitor and parent selection phase. First, we assume that for each sensor node, the maximum number of sensor nodes it can communicate directly is a constant. Under this assumption, for each sensor node \(v_i\), \(O(|S_i|) = O(|P_i|) = O(|M_i \cup CH_i|) = O(1)\) holds. The breakdown of the time complexity of this phase as follows.

a) Steps 1-5. Notice that the connectivity graph is connected. Therefore, the time complexity of steps 1-5 for all the groups is \(O(e)\).

b) Step 6. The time complexity of this step for all the groups is \(O(ne)\), where \(n\) is the number of sensor nodes in the WSN.

As a result, the time complexity of this phase is \(O(ne)\).

3) The energy balancing phase. Notice that each time when a role switch occurs, \(W\) will decrease monotonically. For each sensor node \(v_i\) in \(C_l\), it is processed at most \(|CH_i|\) times by the energy balancing algorithm. Therefore, the energy balancing algorithm terminates in \(\sum_{i \in C_l} |CH_i| = O(|C_l|)\) steps for each group \(C_l\). For each group \(C_l\), each step takes \(O(|C_l| \log |C_l|)\). As a result, the time complexity of this phase is \(O(n^2 \log n)\).

As discussed above, the time complexity of our heuristic approach is \(O(e) + O(ne) + O(n^2 \log n) = O(ne + n^2 \log n)\).

V. MINLP-BASED APPROACH

The objective of MINLP-based approach is to construct a shortest path tree for routing and assign a monitor for each
sensor node such that the network lifetime is maximized while the four requirements described in Section III are met.

The MINLP-based approach consists of two phases. In the first phase, it partitions all the sensor nodes into \( m \) disjoint groups \( C_l (l = 1, 2, \cdots, m) \) such that the shortest path length between each sensor node in the group \( C_l \) to the base station in the connectivity graph is equal to \( l \). In the second phase, for each sensor node \( v_i \in C_l (l = 2, 3, \cdots, m) \), it assigns a monitor and a parent in \( C_{l-1} \) to \( v_i \) such that the maximum energy consumption per unit time of all the individual sensor nodes is minimized by using MINLP.

As in our heuristic approach, we use \( (v_i, v_j, v_k)(j < k) \) to uniquely represent a triangle formed by \( v_i, v_j \) and \( v_k \) in the connectivity graph, where \( v_i \) is in \( C_l \), \( v_j \) and \( v_k \) are in \( C_{l-1} \). For each sensor node \( v_i \in C_l \), we also use \( S_l \) to denote a set of all the triangles in the connectivity graph each of which contains \( v_i \) and two sensor nodes in \( C_{l-1} \) as in Section IV.

For each triangle \( (v_i, v_j, v_k) \), we introduce two binary decision variables \( x(i,j,k) \) and \( y(i,j,k) \) as follows:

\[
x(i,j,k) = \begin{cases} 1 & \text{if } v_j \text{ is the monitor } & v_k \text{ is the parent of } v_i \\
0 & \text{otherwise}
\end{cases}
\]

\[
y(i,j,k) = \begin{cases} 1 & \text{if } v_k \text{ is the monitor } & v_j \text{ is the parent of } v_i \\
0 & \text{otherwise}
\end{cases}
\]

Therefore, for each sensor node \( v_i \in C_l (l = 1, 2, \cdots, m) \), we have the following monitor and parent selection constraint:

\[
\sum_{(v_i, v_j, v_k) \in S_l} x(i,j,k) + y(i,j,k) = 1
\]

The above constraint implies that among all the sensor nodes in \( C_{l-1} \) with which \( v_i \) can communicate directly, only two sensor nodes are selected as the monitor and the parent of \( v_i \), respectively, and the monitor and the parent can communicate with each other.

Next, we derive the energy constraint and other related constraints for each sensor node. For each sensor node \( v_i \in C_l (l = 1, 2, \cdots, m) \), we further introduce the following notations:

1) \( l_i \): the number of packets \( v_i \) receives from all its children in the shortest path tree per unit time.

2) \( m_i \): the total energy consumption per unit time by \( v_i \) being a monitor.

3) \( p_i \): the total energy consumption per time unit by \( v_i \) being a parent.

4) \( e_i \): the total energy consumption per unit time by \( v_i \).

For each sensor node \( v_i \), if it is in \( C_m \), it is a leaf node in the shortest path tree. Therefore, we have the following constraint on \( l_i \):

\[
l_i = \begin{cases} 0 & v_i \text{ is in } C_m \\
\sum_{v_k \in CH_i (v_k, v_i, v_j) \in S_k} y(k,i,j) * l_k + \sum_{v_k \in CH_i (v_k, v_j, v_i) \in S_k} x(k,j,i) * l_k & \text{otherwise}
\end{cases}
\]

(7)

If a sensor node \( v_i \) is the monitor of a sensor node \( v_k \), it needs to overhear the packet transmission when \( v_k \) sends its packet to its parent \( v_j \) and when \( v_j \) forwards the packet to its own parent. Therefore, we have the following constraint on \( m_i \):

\[
m_i = \sum_{v_k \in M_i (v_k, v_i, v_j) \in S_k} y(k,i,j) * 2(l_k + 1) + \sum_{v_k \in M_i (v_k, v_j, v_i) \in S_k} x(k,j,i) * 2(l_k + 1)\alpha
\]

(8)

If a sensor node \( v_i \) is a parent of a sensor node \( v_k \), it needs to not only receive the data from \( v_k \), but also forward the data to its parent. Therefore, we have the following constraint on \( p_i \):

\[
p_i = \sum_{v_k \in CH_i (v_k, v_i, v_j) \in S_k} y(k,i,j) * l_k (\alpha + \beta) + \sum_{v_k \in CH_i (v_k, v_j, v_i) \in S_k} x(k,j,i) * l_k (\alpha + \beta)
\]

(9)

For each sensor node \( v_i \), its energy consumption per time unit consists of three parts: \( m_i \), \( p_i \), and the energy for transmitting its own packet. Therefore, we have the following constraint on \( e_i \):

\[
e_i = m_i + p_i + \beta
\]

(10)

Our optimization objective function is as follows:

\[
\min \max_{v_i \in V} \{e_i\}
\]

(11)

VI. ILP-BASED APPROACH

The ILP-based approach consists of two phases. In the first phase, it partitions all the sensor nodes into \( m \) disjoint groups \( C_l (l = 1, 2, \cdots, m) \) such that the shortest path length between each sensor node in the group \( C_l \) to the base station in the connectivity graph is equal to \( l \). In the second phase, for each group \( C_l (l = m, m-1, \cdots, 2) \), it assigns a monitor and a parent in \( C_{l-1} \) to each sensor in \( C_l \) such that the maximum energy consumption per time unit of all the individual sensor nodes in \( C_{l-1} \) is minimized by using ILP.
Next, we show how to use ILP to find a locally optimal assignment of a monitor and a parent for each sensor node in $C_l (l = m, m - 1, \cdots, 2)$.

Similar to the MINLP approach, for each sensor node $v_i \in C_1$, we have the following monitor and parent selection constraint:

$$\sum_{(v_i,v_j,v_k) \in S_i} x_{(i,j,k)} + y_{(i,j,k)} = 1$$

(12)

The binary decision variables $x_{(i,j,k)}$ and $y_{(i,j,k)}$ are defined in the same way as in the MINLP approach.

For each sensor node $v_i \in C_{l-1}$, we have the following energy constraint on $m_i$:

$$m_i = \sum_{v_k \in M_i (v_k,v_i,v_j) \in S_k} x_{(k,i,j)} * 2(l_k + 1)\alpha +$$

$$\sum_{v_k \in M_i (v_k,v_i,v_j) \in S_k} y_{(k,i,j)} * 2(l_k + 1)\alpha$$

(13)

If $v_k$ is a sensor node in $C_m$, $l_k$ is equal to 0. Notice that each $l_k$ is a constant as our ILP-based approach has finished the monitor and parent assignment for each group $C_j (j > l)$.

For each sensor node $v_i \in C_{l-1}$, we have the following constraint on $p_i$:

$$p_i = \sum_{v_k \in CH_i (v_k,v_i,v_j) \in S_k} y_{(k,i,j)} * l_k(\alpha + \beta) +$$

$$\sum_{v_k \in CH_i (v_k,v_i,v_j) \in S_k} x_{(k,i,j)} * l_k(\alpha + \beta)$$

(14)

For each sensor node $v_i \in C_{l-1}$, we have the following constraint on $e_i$:

$$e_i = m_i + p_i + \beta$$

(15)

Our optimization objective function is as follows:

$$\min_{v_i \in C_{l-1}} \max \{ e_i \}$$

(16)

After selecting the parent and the monitor of each sensor node in $C_l$, our ILP-based approach computes $l_k$ for each sensor node $v_k$ in $C_{l-1}$.

VII. SIMULATION RESULTS

A. Setup

In order to evaluate our approaches, we use NS 2.35 to generate 150 instances with three different distributions, namely, uniform, grid, and random distributions. We vary the number of sensor nodes from 100 to 300 with an increment of 50. For each scenario with a fixed number of sensor nodes and a particular distribution, we generate 10 instances. For each instance, sensor nodes are deployed in a 300 m x 400 m rectangular area, and the base station is deployed at
the middle of the upper boundary of the rectangular area. The initial energy of every sensor node is 0.5 KJ. The transmission range is fixed to 70 m. The energy consumption for receiving one packet of data is $\alpha = 0.001$ KJ while the energy consumed to transmit one packet of data is $\beta = 0.002$ KJ. MIDACO Solver is used to solve the MINLP problems and the Intlinprog Solver of MATLAB is used for the ILP problems.

The hardware platform is Intel Core i5-3470 with a clock frequency of 3.20 Ghz, a memory size of 8 GB and a cache size of 8134 MB.

B. Simulation Results

Figure 1. shows the network lifetimes of the shortest path overhearing trees constructed by the three approaches for all the instances in the three different distributions.

The simulation results show that the heuristic approach obtains comparable network lifetimes compared with the ILP-based approach and the MINLP-based approach. For example, for the instance having 150 sensor nodes with uniform distribution as shown in Figure 1. (b), the network lifetime obtained by the heuristic approach is 3.4, comparable to the network lifetime 4.42 obtained by the ILP-based approach, and the network lifetime 4.5 achieved by the MINLP-based approach.

The relative maximum network lifetime obtained by the heuristic approach occurs in the instance of 100 randomly distributed sensor nodes, as shown in Figure 1.(c), and the network lifetime is 7.29. The relative minimum network lifetime obtained by the heuristic approach is 3.4, which occurs in the instance of 150 uniformly distributed nodes, as shown in Figure 1. (b).

Comparing the heuristic approach with the MINLP-based approach, the minimum ratio between the network lifetime achieved by the heuristic approach and the network lifetime obtained by the MINLP-based approach is 76.6%, which occurs in the instance of 150 nodes as shown in Figure 1.(b) while the maximum ratio is 85.62% as shown in Figure 1.(b). The average ratios for the grid distribution, the uniform distribution and the random distribution are 83.68%, 81.13%, and 78.25%, respectively. The average ratio for all the instances with the three different distributions is 81.05%.

In comparison between the ILP-based approach and the MINLP-based approach, the minimum ratio between the network lifetime achieved by the ILP-based approach and the network lifetime obtained by the MINLP-based approach is 94.43%. The maximum ratio is 98.24%. The average ratios for the grid distribution, the uniform distribution and the random distribution are 94.43%, 98.23%, and 93.7%, respectively. The average ratio for all the instances with the three different distributions is 96.2%.

In comparison between the heuristic approach and the ILP-based approach, the minimum ratio between the network lifetime achieved by the heuristic approach and the
network lifetime obtained by the ILP-based approach is 78.04% while the maximum ratio is 93.1%. The average ratios for the grid distribution, the uniform distribution and the random distribution are 85.6%, 83.94%, and 87.5%, respectively. The average ratio for all the instances with the three different distributions is 85.69%.

Figure 2 shows the running times of all the three approaches for all the instances with the three different distributions.

The simulation results show that the heuristic approach always constructs a shortest path overhearing tree in a reasonable amount of time for all the instances while both the MINLP-based approach and the ILP-based approach do not scale. For example, the ILP-based approach fails to construct a shortest path overhearing tree for the instances with more than 200 sensor nodes in 3 hours while the MINLP-based approach fails to construct a shortest path overhearing tree for the instances with more than 100 sensor nodes in 3 days.

VIII. Conclusion

We investigate the problem of constructing a shortest path overhearing tree with maximum network lifetime, and propose three approaches, a polynomial-time heuristic approach, a MINLP-based approach and an ILP-based approach. We have implemented our approaches using MiDACO solver and MATLAB Intlinprog, and performed extensive simulations on the 150 network instances with three different distributions, namely, uniform, grid, and random distributions. The simulation results show that the heuristic approach performs very well. The average lifetime of all the network instances achieved by the heuristic approach is 85.69% of that achieved by the ILP-based approach and 81.05% of that achieved by the MINLP-based approach.

All the approaches proposed in this paper are centralized ones. A centralized approach requires that the base station collect the connectivity graph from all the sensor nodes, and send the shortest path overhearing tree constructed to all the sensor nodes. We can construct a shortest path spanning tree with a special naming scheme as proposed in [22] for collecting the connectivity graph and sending the shortest path overhearing tree. A distributed heuristic can avoid those costs. Nevertheless, a distributed heuristic introduces additional message complexity. It is interesting to know if there is a distributed heuristic for the problem with better overall performance than our centralized heuristic.

One limitation with our shortest path overhearing tree is that if a monitor colludes with the parent of the monitored sensor node, the attacks by the parent of the monitored sensor node may not be detected. Our future work includes constructing a more secure shortest path overhearing tree where there are multiple monitors for each sensor node.

REFERENCES


