Computing Localized Power-Efficient Data Aggregation Trees for Sensor Networks

Hüseyin Ozgür Tan, *Member*, *IEEE*, Ibrahim Korpeoglu, *Member*, *IEEE*, and Ivan Stojmenovic, *Fellow*, *IEEE*

Abstract—We propose localized, self organizing, robust, and energy-efficient data aggregation tree approaches for sensor networks, which we call Localized Power-Efficient Data Aggregation Protocols (L-PEDAPs). They are based on topologies, such as LMST and RNG, that can approximate minimum spanning tree and can be efficiently computed using only position or distance information of one-hop neighbors. The actual routing tree is constructed over these topologies. We also consider different parent selection strategies while constructing a routing tree. We compare each topology and parent selection strategy and conclude that the best among them is the shortest path strategy over LMST structure. Our solution also involves route maintenance procedures that will be executed when a sensor node fails or a new node is added to the network. The proposed solution is also adapted to consider the remaining power levels of nodes in order to increase the network lifetime. Our simulation results show that by using our power-aware localized approach, we can almost have the same performance of a centralized solution in terms of network lifetime, and close to 90 percent of an upper bound derived here.

Index Terms—Wireless networks, sensor networks, data gathering, minimum energy control, distributed algorithms.

1 Introduction

The design of wireless sensor networks depends on the application requirements. Environmental monitoring is an application where a region is sensed by numerous sensor nodes and the sensed data are gathered at a base station (a sink) where further processing can be performed. The sensor nodes for such applications are usually designed to work in conditions where it may not be possible to recharge or replace the batteries of the nodes. This means that energy is a very precious resource for sensor nodes, and communication overhead is to be minimized. These constraints make the design of data communication protocols a challenging task [1].

A common scenario of sensor networks involves deployment of hundreds or thousands of low-cost, low-power sensor nodes to a region from where information will be collected periodically. Hence, sensor nodes will periodically sense their nearby environment and send the information to a sink which is not energy limited. The collected information can be further processed at the sink for end-user queries. In order to reduce the communication overhead and energy consumption of sensors while gathering, the received data can be combined to reduce message size. A simple way of doing that is aggregating the data. A different way is data fusion (aggregation) which can be defined as producing a more accurate signal by combining several unreliable data measurements. In this paper, we focus on scenarios where perfect aggregation is used while gathering data, meaning that all forwarded messages are of the same size.

An important problem studied here is finding an energy-efficient routing scheme for gathering all data at the sink periodically so that the lifetime of the network is prolonged as much a possible. The lifetime can be expressed in terms of rounds where a round is the time period between two sensing activities of sensor nodes.

There are several requirements for a routing scheme to be designed for this scenario. First, the algorithm should be distributed since it is extremely energy consuming to calculate the optimum paths in a dynamic network and inform others about the computed paths in a centralized manner. The algorithm must also be scalable. The message and time complexity of computing the routing paths must scale well with increasing number of nodes. Another desirable property is robustness, which means that the routing scheme should be resilient to node and link failures. The scheme should also support new node additions to the network, since not all nodes fail at the same time, and some nodes may need to be replaced. In other words, the routing scheme should be self-healing. The final and possibly the most important requirement for a routing scheme for wireless sensor networks is energy efficiency.

A previous study [2] showed that the minimum spanning tree (MST)-based routing provides good performance in terms of lifetime when the data are gathered using aggregation. In that work, the authors proposed a new centralized protocol called PEDAP. The idea in PEDAP is to use the minimum energy cost tree for data gathering. This tree can be efficiently computed in centralized manner using Prim's minimum spanning tree algorithm [3]. In PEDAP-PA, the authors changed the cost of the links so that the remaining energy of the sender is also taken into consideration. Since the link costs vary over time, the authors proposed recomputing the routing tree from time to time using a power-aware cost function. By changing the routing tree over time, the load on the nodes is balanced and a longer lifetime compared to the static version is

- *H.O. Tan and I. Korpeoglu are with the Department of Computer Engineering, Bilkent University, 06800 Ankara, Turkey. E-mail: {hozgur, korpe}@cs.bilkent.edu.tr.*
- *I. Stojmenovic is with SITE, University of Ottawa, 800 King Edward, Ottawa, ON K1N 6N5, Canada. E-mail: ivan@site.uottawa.ca.*

Manuscript received 13 Nov. 2008; revised 4 Oct. 2009; accepted 13 Oct. 2009; published online 31 Mar. 2010.
Recommended for acceptance by S. Olariu.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2008-11-0454.
Digital Object Identifier no. 10.1109/TPDS.2010.68.

achieved. In this way, the lifetime in terms of the first node failure is almost doubled.

The most important disadvantage of these two protocols, however, is their centralized nature. In this paper, we propose a localized version of PEDAP, which tries to combine the desired features of MST and shortest weighted path-based gathering algorithms. We also expand the idea and propose a new family of localized protocols for the power-efficient data aggregation problem. Our main concern in this work is the lifetime of the network. We name our new approach *localized power-efficient data aggregation protocol* (L-PEDAP).

Our proposed routing approach consists of two phases and satisfies the requirements stated above. In the first phase, it computes a sparse topology over the visibility graph of the network in a localized manner. In the second phase, it computes a data gathering tree over the edges of the computed sparse topology. The topology needs to be efficiently computed by using only the one-hop neighborhood information.

For the first phase, we test two different sparse topologies in a distributed manner, namely, local minimum spanning tree (LMST) [4] and relative neighborhood graph (RNG) [5]. These structures are supersets of MST and can be efficiently computed in a localized manner. For the second phase, we propose three different methods and provide performance results of them. All of the methods are based on flooding a special packet using only the edges of the computed structure. According to the decisions made during this flooding process, the tree is yielded. These three methods that can be executed at a node for choosing the parent node toward the sink are to choose: 1) the first node from which the special packet is received, 2) the node that minimizes the number of hops to the sink, and 3) the node that minimizes the total energy consumed over the path to the sink.

Our solution can also handle new node arrivals and departures of existing nodes. Hence, it is adaptive. The routing path is maintained when those dynamic conditions occur. We also propose power-aware versions of our protocols that consider the dynamic changes in the remaining energy levels of nodes while constructing the sparse topologies and routing trees.

We also derive a new theoretical upper bound for the lifetime in terms of the first node failure. We used this upper bound to see how close our protocols are to the theoretical limit. The simulation results showed that our protocols can achieve up to 90 percent of the upper bound.

The rest of the paper is organized as follows: Section 2 defines our system model and describes the problem we solve. Section 3 briefly discusses the related work. In Section 4, we give a theoretical analysis about the lifetime. In Section 5, we describe our solution in detail. In Section 6, we present our comprehensive set of simulation results. Section 7 concludes the paper and discusses some future directions of research. Preliminary conference version of this paper appeared in [6].

2 System Model and Problem Statement

The following are our assumptions about the features of sensor networks and application scenarios we consider in this paper. The sensor nodes are homogeneous and energy constrained. Sensor nodes and sink are stationary and located randomly. Every node knows the geographic location of itself by means of a GPS device or using some other localization techniques [7], [8]. Every node senses periodically its nearby environment and has data to send to the sink in each round. The nodes have a maximum transmission range denoted by R. Sensor nodes are thus normally not in direct communication range of each other. Therefore, applying centralized approaches will have a high communication cost for gathering network information at a node. Data aggregation is used to reduce the data volume. We assume a perfect aggregation or correlation of data which means that combining n packets, each packet being of size k, results in only one packet of size k. We also assume that the sensing period (the duration of a round) is much larger than the time required for transmitting all the information from all nodes to the sink.

2.1 Problem Statement

We model the reachability in a sensor network using a visibility graph $G = (V, E)$, where V is the set of sensor nodes and the base station, and E is the set of edges e_{ij}, where the distance between node i and j is smaller than the maximum transmission distance R. In the application scenario, we consider that for this network model (called the unit disk graph model), sensor nodes periodically sense the environment and generate data in each round of communication. Given a routing plan, each sensor node receives the data from its children, aggregates them into one single packet, and sends the packet to the next node on its way to the sink. Instances of such an application can be event (fire and intrusion) detection systems (starting from all the sensors located near an event) or average data (temperature and humidity) extraction systems (where all active sensors can participate).

The problem is to find an energy-efficient routing plan which maximizes the network lifetime. The routing plan determines for each node the incoming and outgoing neighbors for data forwarding and aggregation. In other words, a tree spanning all the nodes must be found as the routing plan. The routing scheme should also include mechanisms to handle node failures and support new node arrivals.

In the context of sensor networks, the network lifetime can be defined in various ways. One of the definitions can be the time elapsed in terms of rounds until the first node depletes all of its energy, as investigated in Section 2.2. This metric is appropriate to measure the load balancing feature of a routing algorithm. So, if the energy is balanced well among the nodes, the time until the first node drains out its energy will be maximized. Another alternative definition can be the time elapsed until the network is partitioned so that some of the alive sensor nodes cannot transmit their data to the base station. With this metric, one could measure how the bottleneck nodes are handled. If the network becomes partitioned quickly, it means that the energy expenditure of the bottleneck nodes is not managed well. It is desirable that a routing scheme considers several lifetime definitions and provide reasonably good results for them. In our work, we provide results related to both of the definitions mentioned above.

2.2 Energy Consumption Model

There are different models proposed for modeling energy consumption in sensor nodes. Here, we use the first order radio model proposed in [9]. In this model, the energy consumed to transmit a k bit packet to a distance d (denoted as E_{tx}) and the energy consumed to receive a k bit packet (denoted as E_{rx}) are given as follows: $E_{tx}(k, d) = ak + bkd^n, E_{rx}(k) = ck$. In this model, a and c are the energy consumption constants of the transmit and receiver electronics, respectively, and b is the energy consumption constant for the transmit amplifier. There are various studies in the literature based on this model. Heinzelman et al. [9] propose and use this model, assuming that $a = c = 50$, $b = 0.1$, and $n = 2$. On the other hand, in [10], $(a + c) = 2 \times 10^8$, $b = 1$, and $n = 4$.

According to our model, if we express the total energy cost of transmitting a k bit packet from a node i to a neighboring node j as $C_{ij}(k)$, then $C_{ij}(k)$ is given as follows:

$$\begin{aligned} C_{ij}(k) &= ak + bkd_{ij}^n, \text{ if } j \text{ is sink}, && (1)\\ &= (a + c)k + bkd_{ij}^n, \text{ otherwise}. && (2) \end{aligned}$$

The costs of transmission of one packet to another node and to the sink are different, since the sink has no energy constraints and its cost for receiving messages is ignored.

We are assuming that we are routing the data packets on a tree rooted at the sink. Hence, a sensor node receives data from several children and transmits to a single parent after aggregation. Therefore, the total energy consumed at a node i ($\omega_i(k)$) for receiving a k bit packet from its children and for sending the packet to its parent is given as: $\omega_i(k) = \delta_i^+ ck + ak + bkd_{ij}^n$, where δ_i^+ denotes the in-degree of node i in the given routing tree. Thus, there are two parameters that affect the energy consumption of a node: the in-degree of the node and the distance of the node to its parent. Nodes with high in-degrees could quickly drain their energies. Since distance has a power of n, the increase in load is exponential when the distance is increased. Therefore, to obtain a routing tree that is maximizing the lifetime, we have to balance the energy load among the nodes, and we have to try to minimize the degree for a node while minimizing the distance the node will transmit.

If we use a static routing tree, the lifetime of a node i (L_i) and the lifetime of the network (L) can be given in terms of rounds as follows: $L_i = E_0/\omega_i, L = \min(L_i)$, where E_0 is the initial energy of the nodes (assuming that all nodes have the same initial energy). So, in order to maximize L, we have to choose the routing tree which maximizes the minimum L_i value.

3 Related Work

3.1 Routing Protocols

There exist several routing protocols for data gathering without aggregation. The majority of them use the shortest weighted path approach using several combinations of transmission power, reluctance, hop count, and energy consumption metrics [11], [12], [13], [14]. The classical routing algorithms such as AODV [15] or Directed Diffusion [16] can be considered also for this case.

There are also a number of protocols for data gathering with aggregation. Most of them are centralized approaches and assume that all the sensor nodes are in direct communication range of each other and the sink. Kalpakis et al. [17] propose a linear programming solution to maximize the lifetime. The solution provides near-optimal results. However, their approach has high computational cost and must be applied in a central location. Heinzelman et al. [9] propose a distributed two-level hierarchy called LEACH. In this protocol, sensors randomly decide whether or not to become clusterheads. If not, they join the nearest clusterhead and transmit sensed data to it. Clusterheads aggregate collected data and transmit directly to the sink. Since LEACH protocol relies on randomization, it is far from being optimal. Lindsey and Raghavendra [18] proposed PEGASIS protocol in which the sensors are organized into a chain by a centralized algorithm. They transmit to each other along the chain, aggregate received data, and last sensor in the chain transmits to the sink. This approach is also not very efficient, since the transmission distances can be quite long and finding a minimum distance chain is NP-complete (traveling salesman problem). Also, the delay is another problem for PEGASIS.

There are also algorithms in the literature that take the data growth factor into consideration, where data may not be perfectly aggregated. The purpose of these papers is to provide an optimal routing solution which is adaptive to the data growth factor. Hua and Yum [19] described an algorithm for joint optimization of routing and data aggregation. Row data are sent to downstream neighbors. The receiving neighbor encodes the data using local information, with certain compression rate. Transit data (already compressed by upstream neighbors) are directly forwarded to the next-hop neighbors. Therefore, data aggregation is done only by neighbors of measuring sensors, and the size of aggregated data varies. This problem statement and the model are different from the ones used in this paper. Upadhyayula and Gupta [20] proposed a combination of single-source shortest path spanning tree and minimal spanning tree algorithms to construct optimal data aggregation tree which controls latency by limiting the number of children of each node while optimizing energy consumption. Constant data growth factor spans aggregation level from no aggregation to full aggregation at each intermediate node. Although the problem statement is more general than the one in this paper, their algorithm is centralized. One important point is that the authors consider MST as optimal solution in perfect correlation case. Park and Sivakumar [21] optimized number of messages sent while aggregating data originating from k of the n sensors, with various data growth factors. Their solution aggregates correlated data from neighboring sources at nodes of minimum dominating set (MDS). It then creates shortest path of MDS nodes tree by basic flooding. We consider perfect correlation with $k = n$. For this case, the work in [21] reduces to a constant number of messages (one per each sensor) and does not consider energy optimization.

In [2], Tan and Korpeoglu showed that their protocol PEDAP, which routes the packets on the edges of an MST, improves the system lifetime dramatically compared to its

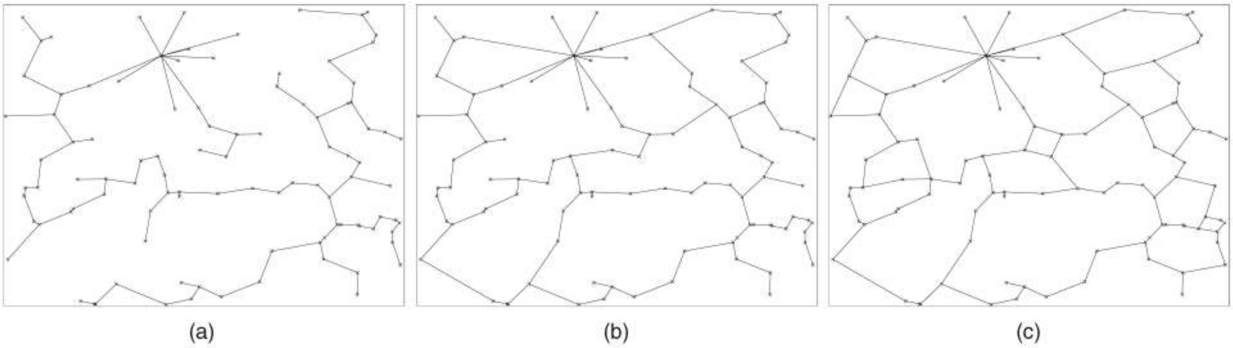


Fig. 1. Comparison of different topologies. (a) MST. (b) LMST. (c) RNG.

alternatives. PEDAP protocol uses the link costs given in (1) and (2) and computes the minimum energy cost tree by using Prim's MST algorithm. PEDAP protocol differs from the euclidean MST with only the degree of the sink. Fig. 1a shows this difference. Fortunately, for the nodes, the properties of the euclidean MST are conserved. For example, the degree of the nodes (except the sink) is at most 6. Also, as stated in [22], the longest edge in the euclidean MST is the minimum common transmission range for network to be connected. So, the transmission distances are also optimal for the nodes routing using PEDAP. As shown in Section 2, the energy load of a node (ω_i) is directly related with its degree and the distance to its parent, and PEDAP balances these parameters well. Also, PEDAP consumes the minimum amount of energy in a single round. Moreover, the authors also propose a power-aware version of their protocol, which they call PEDAP-PA. This protocol provides near-optimal lifetime for the first node failure by sacrificing the lifetime for the last alive node. The idea behind PEDAP-PA is to use a power-aware cost function for a link that considers the remaining energy of the sender. Specifically, the cost function is: $C^*_{ij} = C_{ij}/r_i$, where r_i is the normalized remaining energy of node i. This cost metric is not symmetric. It is used by a node j when looking for candidate neighbor i on route toward sink. The PEDAP-PA algorithm simply finds the minimum spanning tree with these link costs. In order to balance the load, it recomputes the routing tree after a predefined number of rounds.

Hussain and Islam [23] proposed energy-efficient spanning tree approach (EESR) which is similar to PEDAP-PA but has some advantages over PEDAP-PA algorithm. For instance, edge weight assignment used in EESR considers both transmitters and receivers remaining energy levels. With the edge weights they use, the algorithm prevents transmitters and receivers from being overloaded. Another advantage of it is dynamic determination of the duration of recomputation period. The algorithm is, however, centralized. MST-based structure is suitable for environments where all the nodes have data to send and data can be aggregated (fused) in the relay nodes. The drawback of PEDAP and EESR protocols is the centralized nature of MST and the lack of quick response to node failures.

Wu et al. [24] study the construction of a data gathering tree to maximize the network lifetime, which is defined as the time until the first node depletes its energy. Nodes do not adjust their transmission radius to the distance to neighbors (different from our model). The problem is shown to be NP-complete. They design a centralized algorithm which aims at finding a spanning tree whose maximal degree is the minimum among all spanning trees, since energy consumption at each node only depends on the number of messages received from children nodes, that is, on the number of children. Such tree then reduces the load on bottleneck nodes.

3.2 Power-Efficient Topologies

There are many topologies proposed in the literature which can be efficiently computed using the location information of one-hop neighbors.

Rodoplu and Meng [10] proposed a localized topology for estimating the shortest weighted path tree which they called enclosure graph. An edge e_{ij} is in the enclosure graph if and only if the direct transmission between node i and node j consumes less energy than the total energy of all links of any path between them. The enclosure graph includes the shortest weighted path tree, and thus, provides a good performance when it is used in routing without aggregation.

The topologies that we focus in this work are supersets of euclidean MST. One of them is the RNG [5]. An edge e_{ij} is included in the euclidean RNG graph if there are no nodes closer to both nodes i and j than the distance between nodes i and j. That is, an edge e_{ij} remains in RNG if it does not have the largest cost in any triangle $\overset{\triangle}{ikj}$, for all common neighbors k. The euclidean MST of a graph is a subgraph of its RNG.

Li et al. [4] propose a neighborhood structure called local MST (LMST) as an alternative topology. LMST is computed as follows: First, each node determines its one-hop neighbors and computes an MST for that set of nodes, based on the distance between nodes as the weight of the edges. After computing the MST of the neighbors, each node i selects the edges (e_{ij}) where node j is a direct neighbor of node i in its MST. The resulting structure is a directed graph. The structure can be converted to an undirected one in two ways [4]. First way is to include edge (e_{ij}) only when both nodes i and j include that edge (LMST^-). The second way is to include that edge when either node i or node j includes it (LMST^+). In this work, we choose to use LMST^- in our simulations, but our algorithm can support both.

There are some desirable properties of the LMST structure which make using the structure in the context of sensor networks advantageous. MST of a graph is a subgraph of its LMST and the LMST is a subgraph of its RNG [25]. The maximum degree of a node is bounded by 6. This is a desirable property since the load of a node is directly related to the degree of the node, as shown in Section 2. In [4], the authors compare their LMST structure with the enclosure graph and find out that the enclosure graph performs better in terms of energy consumption. However, the comparison did not consider the effect of data aggregation.

Although the RNG and LMST structures are defined based on euclidean distances, they can be used with other link cost functions as long as the functions are symmetric [26], [27]. We can use, for instance, the cost function given in (2), while computing the structures. Figs. 1b and 1c show this case. For the rest of the paper, if we mention MST, LMST, and RNG, we mean the structures that are computed using the link costs given in (1) and (2). They resemble the original MST, LMST, and RNG structures, except replacing some links by direct links to sink (the effect of adding (1)). However, the structure may become considerably different in the whole network, if a cost function that depends on nodes' remaining energies is used to define them.

An important advantage of using structures like RNG and LMST is that they can be constructed very efficiently in a localized manner. Node deletions and additions do not globally change the structure. Only local changes in the structure are required and they can be efficiently computed when a node fails or when a new node is introduced to the network.

4 Lifetime Analysis

In this section, we derive a theoretical upper bound for the lifetime of the first failing node in a sensor network using tree-based routing. This upper bound will be used to test the performance of our protocols against a theoretical limit.

Theorem 4.1. *The lifetime of a sensor network (L) in terms of first failing node is bounded by $\bar{L} = nE_0/|T_{min}|$, where $|T_{min}|$ is the minimum possible total energy consumption for a round that can be achieved.*

Proof. Let T_{min} be the tree that gives the minimum total energy consumption for any round. That is, it is a fixed tree which minimizes the total energy consumption for the network. This tree can be derived from the minimum spanning tree algorithm [3] by using the cost functions given in (1) and (2).

Let $|T_{min}|$ be the total energy expenditure in using T_{min} as the routing tree. We can state that in any round, the total energy consumption is $\geq |T_{min}|$.

Although the routing trees may change in each round, the total energy consumption in L rounds is always $\geq L|T_{min}|$. This implies that there exists at least one node whose energy consumption in L rounds is $\geq L|T_{min}|/n$. Since energy of each node is limited by E_0, $L|T_{min}|/n \leq E_0$. Consequently, $L \leq nE_0/|T_{min}|$. □

Thus, we can easily compute the upper bound $\bar{L}$ for any sensor network, where we know the locations of the nodes, by just computing T_{min} which is the minimum spanning tree with edge weights as in (1) and (2). The total cost of the minimum spanning tree gives us $|T_{min}|$, and since we know n and E_0, we can find the upper bound $\bar{L}$.

For static routing tree approach, it seems to be very difficult to achieve this upper bound, since the load on the nodes cannot be balanced in a static tree. As we will see in our simulations, the lifetime of a static method will be far from being optimal.

In this paper, instead of using a static routing tree, we propose dynamically changing the routing tree repeatedly in order to balance the load among the nodes over the time. Although optimal load balance ($|T_{min}|/n$) cannot be achieved for a single round, if we can use a good randomization scheme, we expect maximum average value for ω_i to become closer to $|T_{min}|/n$, and consequently, lifetime becomes closer to $\bar{L}$.

As mentioned in Section 3, PEDAP-PA algorithm recomputes the routing tree in every 100 rounds by using an asymmetric cost function, by applying Prim's minimum spanning tree algorithm [3]. However, our algorithms need a symmetric cost function to compute LMST and RNG. Consequently, we changed the power-aware cost function to the one given below for our dynamic case:

$$C^*_{ij} = C_{ij}/(r_i \times r_j). \tag{3}$$

For the rest of the paper, whenever we refer to PEDAP-D or LMST-D, we mean the structures that are computed using the link costs given in (3). Our simulation results with these cost functions showed that our dynamic approach is a good randomization scheme.

5 Proposed Solution

Our aim is to combine the energy-efficient features of the MST with the distributed nature of the shortest weighted path-based routing schemes, in order to efficiently and locally compute the routing paths that can also provide a superior network lifetime.

5.1 Our Approach

Our approach for solving the aggregation and routing problem consists of two phases: topology construction and aggregation/routing tree computation.

5.1.1 Topology Construction

In this phase, we aim to construct a sparse and efficient topology over the visibility graph of the network in a distributed manner. We have different alternatives for sparse topologies that can be efficient for energy-aware routing. In this work, we choose to investigate the use of RNG and LMST and compare their relative performance. We expect that LMST performs better than RNG because it is sparser. However, there are some aspects that make RNG and LMST comparable. First, the computation of RNG is more efficient than LMST. RNG needs only the location information of one-hop neighbors, whereas LMST needs a second message for informing about the LMST neighbors. This second message contains the local MST neighbors of the nodes, and hence, it is larger in size compared to the first message which contains only the location information.

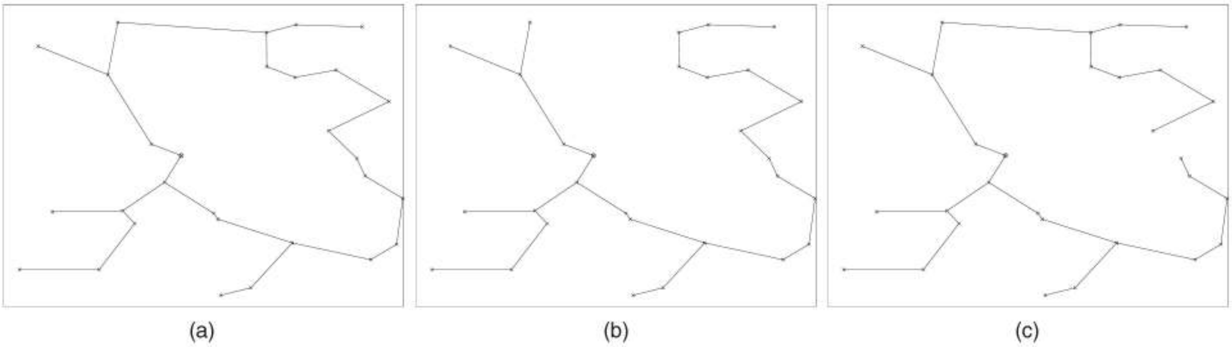


Fig. 2. Comparison of different route computation techniques. (a) LMST. (b) Shortest path on LMST. (c) Minimum hop path on LMST.

On the other hand, one advantage of LMST is that it can approximate MST well especially when the density is high.

In both topologies, we can also use the power-aware cost functions, and consequently, we can efficiently approximate PEDAP-PA. Figs. 1a, 1b, and 1c show MST, LMST, and RNG structures. As seen in the figure, LMST is sparser than the RNG structure. Both can use the same cost functions ((1) and (2)) used in PEDAP.

5.1.2 Routing Tree Computation

There are several methods for obtaining a tree structure (spanning all the nodes) given a graph. In this work, we use a flooding-based tree construction algorithm. A special route discovery packet is broadcasted by the sink and when a node receives that packet, it decides its parent according to the information in the packet. After selecting the parent, it rebroadcasts the packet. The details will be given in Section 5.2. Here, we investigate the efficiency of three different methods: first parent path method (FP), nearest minimum hop path method (MH), and shortest weighted path (i.e., least cost) method (SWP).

The FP method is the simplest among the three. In this method, a node will set its parent as the first neighboring node (among neighbors in selected sparse structure) from which the special route discovery packet was received. In the MH method, the node chooses its nearest neighbor among those with minimum hops to reach to the sink. So, the node updates its parent only if the sender node has a smaller hop count or has the same hop count as the current parent, but it is closer than the current parent (among neighbors in selected sparse structure). Otherwise, the packet is ignored. The SWP method tries to yield a tree that minimizes the cost of reaching the sink for each node. The details will be given in the next section.

At first glance, we expect these three algorithms to give almost the same performance for approximating the minimum spanning tree, since the topologies are sparse enough. However, this is not always the case. Since we use a cost function that uses a power of the distance, minimum hop method cannot give always the most efficient tree. Intermediate nodes at closer distances can make the packet transmission more efficient [12]. Consider the LMST of a sample network given in Fig. 2a. Note that the sink is at the center. Only one edge removal yields a tree. As shown in Fig. 2c, the longest edge is kept by the minimum hop method since choosing that edge reduces the hop count of children nodes toward the sink. However, the shortest path algorithm yields the same tree as MST since having closer nodes reduces the total transmission cost especially when the power of the distance is high, and consequently, longer paths in terms of hop count can be more efficient than the shorter ones.

5.2 Algorithm Details

In our proposed routing scheme, at any time, each sensor node has to know its all one-hop neighbors and their locations, the neighbors on the computed topology, the parent node that it will send the data to in order to reach the sink, and the child nodes that it will receive the data from before it sends the fused or aggregated packet to its parent node. Our solution consists of three parts: *Route Computation*, *Data Gathering*, and *Route Maintenance*.

5.2.1 Topology and Route Computation

The main goal in this phase is to find a sparse topology and set up the routes over it, which means determining the children and parent nodes for each node. At the end of this phase, a data aggregation tree rooted at sink is constructed. The pseudocode for this phase is given in Algorithm 1.

Algorithm 1. Topology and Route Computation

```
Send HELLO message
Collect HELLO messages for $t_{hello}$
Reset Parent ($\pi \leftarrow null$)
Compute neighbors on the sparse topology
**while** ROUTE-DISCOVERY packet $RD$ received
  in $t_{discovery}$ **do**
    **if** update required for $RD$ **then**
        Update parent ($\pi \leftarrow source(RD)$)
        Broadcast ROUTE-DISCOVERY
    **end if**
**end while**
Inform $\pi$ to construct its child-list
```

Initially, the nodes and the sink are not aware about the environment. In the setup phase, all nodes and the sink broadcast HELLO messages, which include their location and remaining energy, using their maximum allowed transmit power. The remaining energy level is advertised only when dynamic (power-aware) protocols are used. We give a time threshold t_{hello} for waiting advertisements, which must be long enough to hear all possible advertisements. After receiving HELLO messages, all nodes are informed about their one-hop neighbors and their locations and energy levels. Each node can then locally compute its neighbors in the desired sparse topology (static and dynamic versions of RNG and LMST). After finding its neighbors in the sparse topology, a node can join the distributed route computation process in order to find its parent and children on the aggregation tree.

The route computation is done via a broadcasting process which starts at the sink node. The sink initiates a ROUTE-DISCOVERY packet in order to find and set up the routes from all sensor nodes toward itself. When a sensor node receives a ROUTE-DISCOVERY packet, it broadcasts the packet to all its neighbors on the computed topology if it updates its routing table. By this way, the routing tree rooted at the sink is established over the sparse topology. An important energy conserving feature of our algorithm is that the packet is sent with a power just enough for reaching all the neighbors on the sparse topology instead of using the maximum power.

Each ROUTE-DISCOVERY packet has three fields: a *sequence ID* which is increased when a new discovery is initiated by the sink, an optional *distance* field which shows the cost of reaching the sink, and an optional *neighbor list* field which is the list of the neighbors of the sending node in the chosen topology. The *distance* field is not required if FP algorithm is chosen. It holds the minimum number of hops or minimum energy cost to reach the sink, respectively, if MH or SWP algorithm is chosen. The *neighbor list* field must only be used if the LMST topology is chosen. So, if we use the FP on RNG topology, we can decrease the message overhead. On the other hand, if we use SWP on LMST, which gives the best performance in all cases according to our experiments, we have to have some overhead. But an important point to mention is that in our approach, since the LMST computation is combined with the route computation, no extra messages are used for negotiation among LMST neighbors. Only overhead is the size of the ROUTE-DISCOVERY packet.

Upon receiving a new ROUTE-DISCOVERY packet, the sensor node ignores the packet if it is not coming from a direct neighbor, in order to ensure using only the edges in the computed topology.

After that, according to the routing strategy chosen, the node decides whether or not to update its parent. If FP strategy is used, the node updates the parent information only if it has not a parent yet. In MH strategy, the node compares its current parent with the sending node and chooses the sender as its new parent if it has a smaller hop count to the sink or has the same number of hops but is closer to the node. And finally, if the SWP is chosen, the node updates its parent only if the path using the sender node is advantageous in terms of total energy consumption. Regardless of the chosen strategy if the packet has a higher sequence ID, the node directly updates its parent. If the node decides to update its parent, it rebroadcasts the ROUTE-DISCOVERY packet with updated fields. If in the time threshold $t_{discovery}$, no other route discovery packets are received, we can conclude that the route setup converged.

At this step, each node can inform its parent, in order to construct the children list which will be used in data gathering phase. After this final step, the data aggregation tree is set up and stabilized. This means that each node knows from which neighbors it will receive data and to which node it will send the received data after aggregation.

5.2.2 Data Gathering

After the parent and children nodes for an individual sensor node are determined, the node can join the data gathering process. In data gathering phase, each sensor node periodically senses its nearby environment and generates the data to be sent to the sink. However, before sending it directly to the parent node, it will wait all the data from its child nodes and aggregate the data coming from them together with its own data, and then, send the aggregated data to the parent node. Thus, at the beginning of data gathering step, only leaf nodes can transmit their data to their corresponding parent nodes. At each step, the data are gathered upward in the tree and reaches the sink after h steps, where h is the height of the aggregation tree. The reason for waiting to receive data from child nodes is to use the advantage of the aggregation. In this way, each sensor only transmits once in a round, and as a result, saves its energy.

5.2.3 Route Maintenance

After setting up the routes, three events can cause a change in the routing plan: route recomputation, node failure, and node addition. We will discuss them separately.

Recomputation of the aggregation tree is required when power-aware (dynamic) cost functions are used. In power-aware methods, the tree must be recomputed at specified intervals. Since the computation depends on the remaining energy of nodes, each time the computation takes place and a different and more power-efficient plan is yielded. In our case, we handle this requirement by broadcasting a new ROUTE-DISCOVERY packet with a new sequence ID. Apparently, in order to utilize the power-aware methods, each node must know the remaining energy levels of its neighbors. In order to exchange the remaining energy levels, we use HELLO messages. So, at the beginning of each recomputation phase, the nodes advertise their remaining energy levels. After that, ROUTE-DISCOVERY packet with a new sequence ID can be broadcasted by the sink. It is worth to mention that in order to achieve recomputation, each node must know the predefined time (in terms of rounds) to send HELLO messages.

Node failures can be due to various reasons. However, the most critical reason is depletion of energy of a node. Previous approaches (e.g., [9], [2], [18]) did not discuss the node failure problem. In these approaches, however, a node failure in communication phase will cause a routing problem in which the descendants of the failed node cannot send their data until next setup phase. In order to prevent this, failures must be handled as soon as possible. In our solution, we handle the case where failures are due to energy depletion. However, the idea behind the solution can be applied to other failure causes as well.

Failure of a node due to energy depletion can be handled gracefully, since the node can predict that it will die soon due to energy limitation. Algorithm 2 presents the route recovery algorithm. In our solution, when a node's energy reduces below a threshold value, the node broadcasts a BYE message using maximum allowed transmit power. All nodes receiving the BYE message will immediately update their local structure. This message is not required to be retransmitted since the node failures do not affect the structure globally. However, in this case, the nodes that cannot reach the sink because of the energy depletion of their ancestor must find a new cost-efficient path to send their packets. In our solution, this is handled in a localized manner as follows: The child nodes of the failed node that receive the BYE message reset their routing tables and enter the parent-discovery phase by broadcasting a special message PARENT-DISCOVERY to its neighbors on the structure. According to the receiver of that special message, if the sender is its own parent on the way to the sink, the receiver also resets its routing table and broadcasts the packet to its neighbors. In this way, all the nodes that should enter the parent-discovery phase will be reached. If the PARENT-DISCOVERY packet is received by a neighboring node of the sender and if it has a valid parent, the receiver constructs a new ROUTE-DISCOVERY packet as mentioned above and broadcasts it to the sender. This ROUTE-DISCOVERY packet is handled as mentioned in Section 5.2.1. It is worth to mention that the sequence ID in this new packet is not incremented; therefore, the update of the routing table takes place only when the newly received cost is smaller. After the route discovery phase converges, the new routes are set up and data gathering can continue.

Algorithm 2. Route Recovery

```
$\pi_{old} \leftarrow \pi$
if BYE message $B$ received then
    remove $source(B)$ from neighbor list
    compute the sparse topology
    if $source(B) = \pi$ then
        Reset parent ($\pi \leftarrow null$)
        Reset child list
        Broadcast PARENT-DISCOVERY message
        Enter route discovery phase
    end if
end if
if PARENT-DISCOVERY message $PD$ received then
    if $source(PD) = \pi$ then
        Reset parent ($\pi \leftarrow null$)
        Reset child list
        Broadcast PARENT-DISCOVERY message
        Enter route discovery phase
    else
        if $\pi \neq null$ then
            Send ROUTE-DISCOVERY
        end if
    end if
end if
if $\pi \neq \pi_{old}$ then
    Inform $\pi_{old}$ and $\pi$ to construct their child-list
end if
```

Consider now the case of node additions. When a new node is deployed, it broadcasts a HELLO message. Its neighbors update their local structure upon receiving this message and also inform the new node about their existence and locations by replying a HELLO message so that the newly deployed node can also determine its neighbors. Nodes that update their local structure send back a ROUTE-DISCOVERY packet including their costs to the newly deployed node. The new node selects the most efficient node as its parent and broadcasts this information by a new ROUTE-DISCOVERY packet. Since the sequence ID is again not incremented, the new packet is broadcasted throughout the network only when using the new node is advantageous. This final step can be avoided if FP method is used. So, the newly added node just chooses its closest neighbor as its parent and starts sending data.

TABLE 1
Comparison of Algorithms—Normalized Lifetime
N:100, R:20, l:100, d:10

	RNG			LMST		
n	SWP	MH	FP	SWP	MH	FP
2	0.917	0.910	0.910	0.991	0.990	0.989
4	0.808	0.401	0.384	0.907	0.744	0.737

6 Simulation Results

In this section, we will first try to choose the best parent selection strategy, and then, continue the experiments with that strategy, since running the experiments with all topologies and strategies will become too complicated.

For our scenario, there are three parameters that we can change to see their effect: number of nodes N, maximum transmission radius R, and side length of the square area l. One other parameter that depends on these three parameters and that gives direct intuition about the scenario is the density d, which is defined as the average number of neighbors per node. For the sake of completeness, we will give the value of d for each scenario since it is immediately very informative.

We generated a network with parameters $N = 100$, $R = 20m, l = 100m \Rightarrow d = 10$. On this network, we repeated the experiments on LMST and RNG topologies with three alternative parent selection strategies. We compared the methods in terms of the lifetime they provide for the first node (normalized lifetime) and how well they approximate the PEDAP tree (approximation percentage). Normalized lifetime is the ratio of the lifetime to the lifetime provided by PEDAP for the same network, whereas approximation percentage is the ratio of the number of common edges with PEDAP tree to the total number of edges.

In Tables 1 and 2, we provide results of experiments that compare the efficiency of three parent selection methods. Here, n denotes the power of distance in the cost function. We can conclude with these results that if the propagation is free space ($n = 2$), using FP algorithm on RNG can be advantageous because the setup cost is minimal in that case and the performance is almost the same as in the best solution SWP on LMST (< 9 percent lesser lifetime). If $n = 4$, however, choosing the shortest weighted path on LMST gives considerably better performance in terms of lifetime. We can also see that the difference among parent selection strategies is more striking when $n = 4$. These results also show that the SWP strategy outperforms its alternatives in each case. Therefore, for the rest of the

TABLE 2
Comparison of Algorithms—Approximation Percentage
N:100, R:20, l:100, d:10

	RNG			LMST		
n	SWP	MH	FP	SWP	MH	FP
2	0.686	0.682	0.666	0.897	0.895	0.894
4	0.825	0.629	0.613	0.927	0.869	0.869

TABLE 3
Upper Bound for FNF—R:20, l:100

N, d	50, 5	100, 10	200, 20	500, 52	1000, 105
OPT	4748	4949	5063	5125	5165

simulations, we always use SWP approach to compare the performances of different topologies.

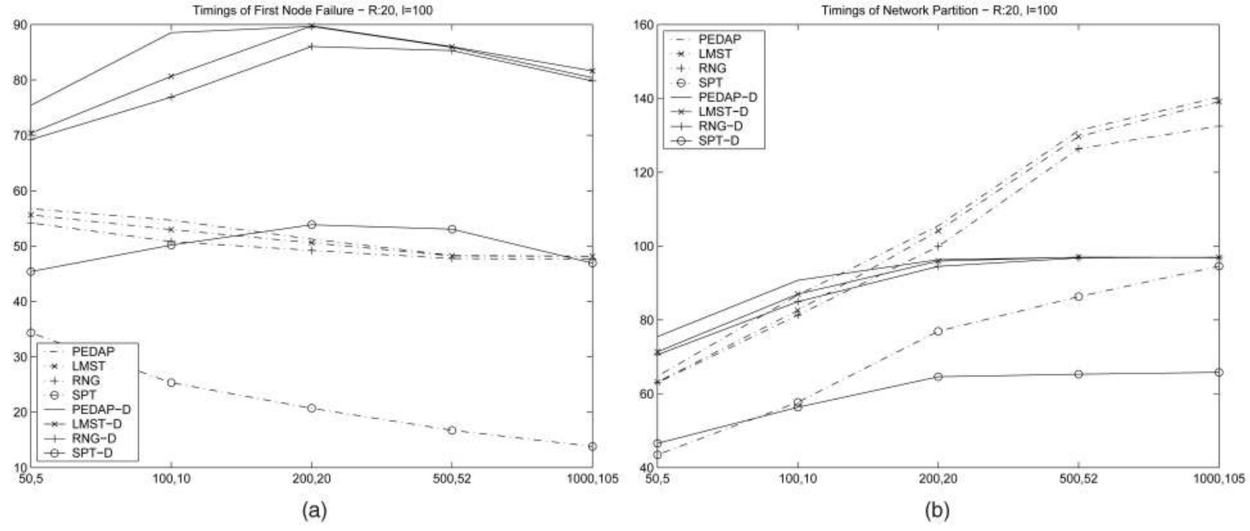
The rest of the simulations evaluates the performance of our routing scheme. We conducted experiments with different values of N, R, and l. For each parameter, we ran the experiments 100 times and obtained an average value for the two evaluation metrics: First Node Failure Time (FNF) and Network Partition Time (NPT). The initial energies of the nodes were given as 0.5 J. For dynamic algorithms (PEDAP-D, LMST-D, etc.), we used the power-aware cost functions given in (3) and recomputed the routing paths every 100 rounds. For transmission costs, we used the parameters of the first order radio model given in [9]. In all of the routing schemes, data aggregation is used at every step for a fair comparison. Also, for all methods, the setup and maintenance costs are not included in energy expenditures, which means that only the cost of data packets is considered. We used a fixed value of 1,000 bits for data packet size k.

In order to compare our algorithms based on LMST and RNG, we also implemented the centralized PEDAP algorithm and the shortest weighted path tree (SPT) as other alternatives. With the dynamic versions, it adds up to eight different methods to compare (PEDAP, LMST, RNG, SPT, PEDAP-D, LMST-D, RNG-D, and SPT-D).

Since the most informative parameter for our scenarios is d, we try to investigate the performances on different values of d. However, there are three ways of changing d: for each of the parameters $N, R,$ and l, we can keep two of them fixed and change the third one. One important point is that in the rest of simulations, we give the results normalized to upper bound $\bar{L}$ which is computed as given in Section 4. In all cases, we provide the actual values of $\bar{L}$.

Consider the impact of the number of nodes N on the lifetime. In Fig. 3a, we can see the normalized lifetimes for various values of N in terms of FNF. Since the upper bound for each case is different, we give the exact values of the upper bound in Table 3. We can see that the upper bound slightly increases with increasing N. We observe that the effect of N is not much significant in static MST-based approaches. For the static SPT approach, however, increasing d decreases the lifetime of the system. This is mainly because of the fact that the SPT approach cannot balance the degree of a node. So, if N is increased, the maximum node degree will also increase. However, in MST-based approaches, since the maximum node degree is bounded by 6, the decrease is not much significant. On the other hand, in all MST-based approaches, the maximum node degree is increasing (from 3-4 to 4-5) with increasing d, and thus, the network lifetime is slightly decreased when N is increased.

Next, we study the impact of N on the dynamic versions of algorithms (PEDAP-D, LMST-D, RNG-D, and SPT-D). When N increases, the lifetimes increase until reaching a maximum and decrease afterward. Since the dynamic versions of the algorithms almost balance the degree among the nodes, this behavior is due to the distances between the neighbors. In low-density case, the distances are long, and since the overhead because of the distance is exponential, the lifetime is far from being optimal. As d increases, the average distance among the nodes becomes closer to the optimal distance—which may be the same as given in [12], [28]. After some point, however, the decrease in distances has a negative effect due to the constant cost of wireless transmission. So, we can conclude that using too many nodes is not always very



(a)

(b)

Fig. 3. Effect of number of nodes on network lifetime for various data gathering schemes. (a) Normalized FNF timings versus (N, d). (b) Normalized NPT timings versus (N, d).

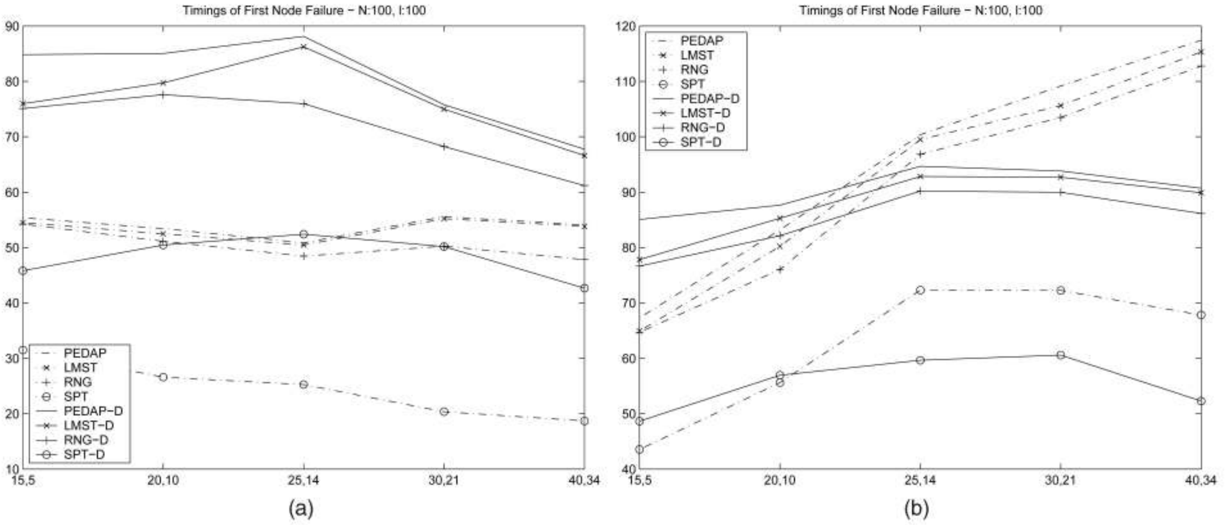


Fig. 4. Effect of transmission radius on network lifetime for various data gathering schemes. (a) Normalized FNF timings versus (R, d). (b) Normalized NPT timings versus (R, d).

effective in providing longer network lifetime. If we compare RNG- and LMST-based approaches, RNG gives very close results with LMST, but LMST performs always slightly better than RNG. At their best, both PEDAP-D and LMST-D achieve 90 percent of the upper bound.

In Fig. 3b, the lifetimes in terms of network partition time are given, normalized to the values given in Table 3. Again, as expected, the lifetime improves with increasing d in static versions of the protocols. However, for the power-aware (dynamic) methods, the increase is smaller. This is explained by the fact that in order to provide longer lifetime in terms of FNF, the system uses more resources.

The lifetimes for different R values are given in Figs. 4a and 4b. As can be seen in Fig. 4a, increased transmission radius dramatically reduces the lifetime of the dynamic methods after some point. The maximum value is achieved when $R = 25$ m. This can be explained by the effect of the distance to parent. With increasing R, although there exist more alternative nodes to choose, the average distance of the alternatives also increases. So, the nodes will tend to send to long distances as the residual energy of the neighbor nodes decreases, and this will cause a decrease in FNF. So, we can say that increasing the radius above some point has an inverse effect on lifetime for our dynamic approach. The dynamic versions may give the best performance when R is chosen equal to the same optimal distance mentioned above. One important point here is that the upper bound for the lifetime is always the same in this scenario, since increasing R does not affect the MST topology. On the other hand, similar results are observed for the network partition times as in the previous case (Fig. 4b).

Another scenario that changes the density is increasing the area size while keeping graph with parameters N and R the same. Fig. 5a shows the normalized simulations results for this case. The upper bounds of each specific case are given in Table 4. The optimal value of the same graph is decreasing with increasing area size, since the average distance among the nodes is increasing. However, if we

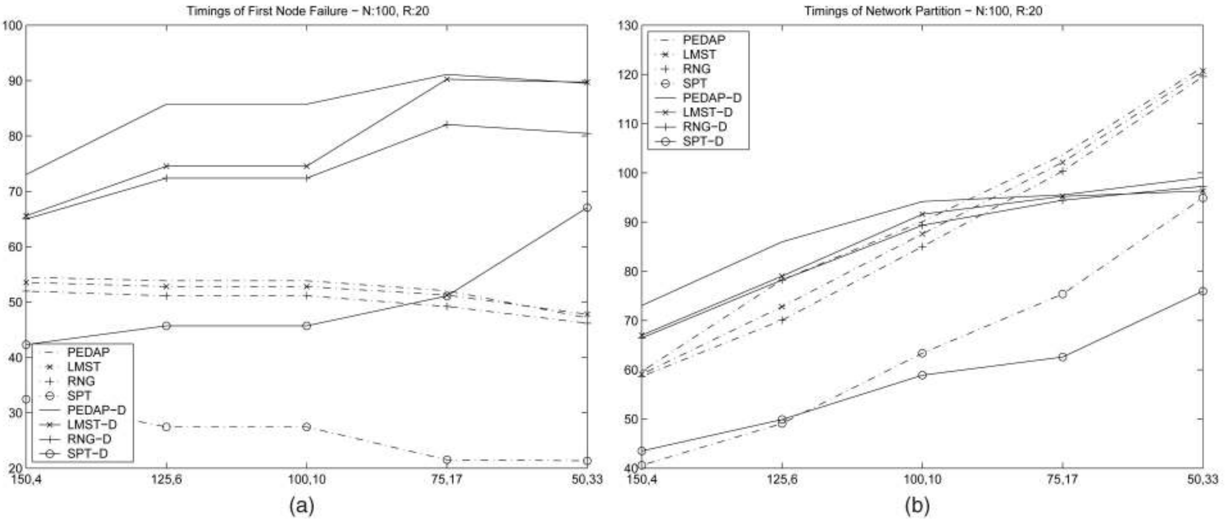


Fig. 5. Effect of area size on network lifetime for various data gathering schemes. (a) Normalized FNF timings versus (l, d). (b) Normalized NPT timings versus (l, d).

TABLE 4
Upper Bound for FNF—N:100, R:20

l, d	50, 33	75, 17	100, 10	125, 6	150, 4
OPT	5871	5256	4983	4776	4576

normalize the lifetime, we observe that for the static methods, the normalized lifetime is slightly increasing with decreasing density. If the dynamic versions are examined, above some density, the PEDAP-D and LMST-D methods can achieve 90 percent of the upper bound. With decreasing density, after some point, the lifetime decreases dramatically. This is expected since when there are more alternative neighbors to choose, our dynamic version can balance the load among the nodes. If the density is low, the number of alternative routing trees becomes also small. This fact combined with the distance effect reduces the system lifetime considerably on wide networks. The reason of the decrease in lifetime on high-density networks is that as the area size becomes smaller, the effect of the distance gets smaller. Similar to the first scenario, the degree plays more important role to determine the lifetime. So, as in the first case, the maximum degree is increased slightly and the overall lifetime decreases.

We can observe similar result also for the NPT timings (Fig. 5b). As the area enlarges, connectivity decreases, and distances get longer. This leads to a decrease in NPT timings.

7 Conclusion

In this paper, we presented a new energy-efficient routing approach that combines the desired properties of minimum spanning tree and shortest path tree-based routing schemes. The proposed scheme uses the advantages of the powerful localized structures such as RNG and LMST and provides simple solutions to the known problems in route setup and maintenance because of its distributed nature. The proposed algorithm is robust, scalable, and self-organizing. The algorithm is appropriate for systems where all the nodes are not in direct communication range of each other. We show through simulations that our algorithm outperforms the shortest weighted path-based approaches, and can achieve 90 percent of the upper bound on lifetime.

One important contribution of this paper is the easily computable theoretical upper bound. By using this value, we can see how good a data aggregation protocol is. The simulation results showed that the SWP over LMST approach is the best among our new family of protocols and by using this approach, one can achieve almost the same performance with the best known centralized solution PEDAP.

Another important result is that dynamic methods tend to increase both FNF and NPT timings especially in reasonable densities for sensor networks ($d < 15$). This means that dynamic methods can balance the energy expenditure among the nodes well while providing good lifetimes for bottleneck nodes.

As a result of the experiments, we also conclude that increasing the node density up to some point results in higher system lifetime. However, after this point, high density leads into poor network performance. With this result, we can see that there should be an optimal density which gives the maximum possible performance.

Although in this work, we have used 100 rounds to recompute the aggregation tree as in PEDAP-PA, it is worth to mention that the period of the recomputation is an important factor for achieving long lifetimes. With a small period, we can achieve a good balance among the nodes, whereas we have larger overhead due to control packets. Determination of the optimal recomputation period needs complex mathematical analysis, and it is beyond the scope of this work. An example of changing recomputation period dynamically in a centralized solution can be found in [23].

The area size and the maximum transmission range are usually set by the application itself. It is an interesting open problem to theoretically derive the optimum number of nodes for given R and l. Also, based on this result, one could combine our method with some sort of sleep scheduling algorithm to get a performance increase on high-density networks. So, if a sleep scheduling algorithm [29] recomputes the roles of the nodes periodically, the same period can be used to recompute the routing tree spanning only the active nodes with our protocols. Moreover, with the advantage of using periodic recomputations, our dynamic methods can be used efficiently in such a scenario. One can also investigate the application of connected dominating sets (CDSs) [30] to limit internal tree nodes to such a set, and rotating periodically these sets. Tree computation via broadcasting is possible only via nodes in CDS, and leaves can even sleep temporarily while data are being gathered.

We did not measure the cost of setup and maintenance. However, our motivation is exactly to address this setup cost and maintenance cost by proposing localized solutions. Almost all existing papers do ignore these costs by describing centralized solutions, without even mentioning the communication overhead involved in gathering needed information. In our study, measuring this cost would be even counterproductive. This cost in our approach is negligible compared to the same cost in existing algorithms which are centralized. By ignoring this cost, we were able to conclude that our localized solutions perform almost as well as centralized, and with over 90 percent.

Acknowledgments

This research is partially supported by NSERC Discovery grant and NSERC Strategic Grant on sensor and actuator networks STPGP 336406-07. This work is also supported in part by the European Commission in the framework of the FP7 Network of Excellence in Wireless COMmunications NEWCOM++ (contract no. 216715).

References

[1] J. Wu and I. Stojmenovic, "Ad Hoc Networks," *Computer,* vol. 37, no. 2, pp. 29-31, Feb. 2004.

[2] H.Ö. Tan and I. Korpeoglu, "Power Efficient Data Gathering and Aggregation in Wireless Sensor Networks," *SIGMOD Record,* vol. 32, no. 4, pp. 66-71, 2003.

[3] R. Prim, "Shortest Connecting Networks and Some Generalizations," *Bell System Technical J.,* vol. 36, pp. 1389-1401, 1957.

[4] N. Li, J.C. Hou, and L. Sha, "Design and Analysis of an mst-Based Topology Control Algorithm," *Proc. IEEE INFOCOM,* 2003.

[5] G. Toussaint, "The Relative Neighborhood Graph of a Finite Planar Set," *Pattern Recognition,* vol. 12, pp. 231-268, 1980.

[6] H.O. Tan, I. Korpeoglu, and I. Stojmenovic, "A Distributed and Dynamic Data Gathering Protocol for Sensor Networks," *Proc. 21st Int'l Conf. Advanced Networking and Applications,* pp. 220-227, 2007.

[7] J. Bachrach and C. Taylor, "Localization in Sensor Networks," *Handbook of Sensor Networks: Algorithms and Architectures,* I. Stojmenovic, ed., pp. 277-310, Wiley, 2005.

[8] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *Computer,* vol. 34, no. 8, pp. 57-66, Aug. 2001.

[9] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proc. 33rd Ann. Hawaii Int'l Conf. System Sciences,* pp. 3005-3014, 2000.

[10] V. Rodoplu and T. Meng, "Minimum Energy Mobile Wireless Networks," *IEEE J. Selected Areas in Comm.,* vol. 17, no. 8, pp. 1333-1344, Aug. 1999.

[11] S. Singh, M. Woo, and C.S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," *Proc. Int'l Conf. Mobile Computing and Networking,* pp. 181-190, 1998.

[12] I. Stojmenovic and X. Lin, "Power-Aware Localized Routing in Wireless Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 12, no. 11, pp. 1122-1133, Nov. 2001.

[13] J.-H. Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Ad-Hoc Networks," *Proc. IEEE INFOCOM '00,* pp. 22-31, Mar. 2000.

[14] J. Chang and L. Tassiulas, "Maximum Lifetime Routing in Wireless Sensor Networks," *Proc. Advanced Telecomm. and Information Distribution Research Program Conf.,* 2000.

[15] C.E. Perkins and E.M. Royer, "Ad-Hoc On-Demand Distance Vector Routing," *Proc. Second IEEE Workshop Mobile Computing Systems and Applications,* p. 90, 1999.

[16] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proc. Int'l Conf. Mobile Computing and Networking,* pp. 56-67, 2000.

[17] K. Kalpakis, K. Dasgupta, and P. Namjoshi, "Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks," *Proc. 2002 IEEE Int'l Conf. Networking (ICN '02),* pp. 685-696, Aug. 2002.

[18] S. Lindsey and C.S. Raghavendra, "Pegasis: Power-Efficient Gathering in Sensor Information Systems," *Proc. IEEE Aerospace Conf.,* Mar. 2002.

[19] C. Hua and T.-S.P. Yum, "Optimal Routing and Data Aggregation for Maximizing Lifetime of Wireless Sensor Networks," *IEEE/ACM Trans. Networking,* vol. 16, no. 4, pp. 892-903, Aug. 2008.

[20] S. Upadhyayula and S.K.S. Gupta, "Spanning Tree Based Algorithms for Low Latency and Energy Efficient Data Aggregation Enhanced Convergecast (DAC) in Wireless Sensor Networks," *Ad Hoc Networking,* vol. 5, no. 5, pp. 626-648, 2007.

[21] I.S.-J. Park and I.-R. Sivakumar, "Energy Efficient Correlated Data Aggregation for Wireless Sensor Networks," *Int'l J. Distributed Sensor Networks,* vol. 4, no. 1, pp. 13-27, 2008.

[22] M.D. Penrose, "The Longest Edge of the Random Minimal Spanning Tree," *The Annals of Applied Probability,* vol. 7, no. 2, pp. 340-361, 1997.

[23] S. Hussain and O. Islam, "An Energy Efficient Spanning Tree Based Multi-Hop Routing in Wireless Sensor Networks," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '07),* pp. 4383-4388, Mar. 2007.

[24] Y. Wu, S. Fahmy, and N.B. Shroff, "On the Construction of a Maximum-Lifetime Data Gathering Tree in Sensor Networks: Np-completeness and Approximation Algorithm," *Proc. IEEE INFOCOM,* pp. 356-360, 2008.

[25] F.J. Ovalle-Martinez, I. Stojmenovic, F. Garcia-Nocetti, and J. Solano-Gonzalez, "Finding Minimum Transmission Radii for Preserving Connectivity and Constructing Minimal Spanning Trees in Ad Hoc and Sensor Networks," *J. Parallel and Distributed Computing,* vol. 65, no. 2, pp. 132-141, 2005.

[26] O. Escalante, T. Perez, J. Solano, and I. Stojmenovic, "Rng-Based Searching and Broadcasting Algorithms over Internet Graphs and Peer-to-Peer Computing Systems," *Proc. ACS/IEEE 2005 Int'l Conf. Computer Systems and Applications (AICCSA '05),* p. 17-I, 2005.

[27] T. Perez, J. Solano-Gonzalez, and I. Stojmenovic, "Lmst-Based Searching and Broadcasting Algorithms over Internet Graphs and Peer-to-Peer Computing Systems," *Proc. IEEE Int'l Conf. Signal Processing and Comm. (ICSPC '07),* pp. 1227-1230, Nov. 2007.

[28] M. Bhardwaj, A. Chandrakasan, and T. Garnett, "Upper Bounds on the Lifetime of Sensor Networks," *Proc. IEEE Int'l Conf. Comm.,* pp. 785-790, 2001.

[29] A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stojmenovic, "Localized Sensor Area Coverage with Low Communication Overhead," *Proc. Fourth Ann. IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom '06),* pp. 328-337, Mar. 2006.

[30] I. Stojmenovic and M. Seddigh, "Broadcasting Algorithms in Wireless Networks," *Proc. Int'l Conf. Advances in Infrastructure for Electronic Business, Science, and Education on the Internet SSGRR,* 2000.

Hüseyin Ozgür Tan received the BS and MS degrees in computer engineering, in 2002 and 2004, respectively, from Bilkent University, Ankara, Turkey, where he is currently working toward the PhD degree. Currently, he is a senior software engineer at HAVELSAN, Inc. His research interests include wireless ad hoc and sensor networks. He is a member of the IEEE.

Ibrahim Korpeoglu received the BS degree in computer science from Bilkent University, Turkey, and the MS and PhD degrees in computer science from the University of Maryland at College Park. Since 2002, he has been an assistant professor in the Computer Engineering Department, Bilkent University. Prior to that, he worked in several research and development companies including IBM T.J. Watson Research Center, Ericsson, Bell Laboratories, and Bellcore. He received Turkish Scientific and Technological Research Council (TUBITAK) Young Investigator Career Award in 2004. In 2006, he received Bilkent University Distinguished Teaching Award, and in 2009, IBM Faculty Award. His research interests include computer networks and systems. His current research focus is on wireless ad hoc and sensor networks, wireless mesh networks, and P2P networks. He is a member of the IEEE and the IEEE Computer Society.

Ivan Stojmenovic received the PhD degree in mathematics. He held regular and visiting positions in Serbia, Japan, USA, Canada, France, Mexico, Spain, United Kingdom (as the chair in Applied Computing at the University of Birmingham), Hong Kong, and Brazil, and is a full professor at the University of Ottawa, Canada. He published more than 250 different papers, and edited four books on wireless, ad hoc and sensor networks, and applied algorithms with Wiley/IEEE. He is an editor of over dozen journals, the editor in chief of the *IEEE Transactions on Parallel and Distributed Systems* (in January 2010), and the founder and editor in chief of three journals (*Multiple-Valued Logic and Soft Computing, Parallel, Emergent and Distributed Systems*, and *Ad Hoc & Sensor Networks*). He has h-index 35 and >6,000 citations. One of his articles was recognized as the Fast Breaking Paper, for October 2003 (as the only one for all of computer science), by *Thomson ISI Essential Science Indicators*. He is the recipient of the Royal Society Research Merit Award, United Kingdom. He is a fellow of the IEEE (Communications Society, class 2008), and is the recipient of Excellence in Research Award of the University of Ottawa 2008-09. He chaired and/or organized > 50 workshops and conferences, and served in more than 100 program committees. Among others, he was/is the program co/vice chair at the IEEE PIMRC 2008, IEEE AINA-07, IEEE MASS-04&07, EUC-05&08, WONS-05, MSN-05&06, and ISPA-05&07; founded workshop series at IEEE MASS, IEEE ICDCS, and IEEE DCOSS; and the workshop chair at the IEEE MASS-09, ACM Mobicom/Mobihoc-07, and Mobihoc-08. He presented more than dozen tutorials.