31. M. Franceschetti, O. Dousse, D. Tse, and P. Thiran. Closing the gap in the capacity of random wireless networks via percolation theory. *IEEE Transactions on Information Theory* 53(4):1009–1018, 2007.
32. V. Bonifaci, P. Korteweg, A. Marchetti-Spaccamela, and L. Stougie. An approximation algorithm for the wireless gathering problem. *Operations Research Letters* 36:605–608, 2008.
33. X.-Y. Li, Y. Wang, and Y. Wang. Complexity of data collection, aggregation, and selection for wireless sensor networks. *IEEE Transactions on Computers* 60(3):386–399, 2011.

Chapter 2

Data Aggregation and Data Gathering

Lalit Kumar Awasthi and Siddhartha Chauhan
National Institute of Technology

Contents

2.1 Introduction

A wireless sensor network (WSN) is a self-organizing network that does not need user intervention for configuration or setting up of routing paths. Due to their advantageous characteristics (low cost, multifunctionality, small size, and mobility), WSNs can be widely used in agriculture, industry, transportation, health care, and everyday life. WSNs can be used in virtually any environment, even in tough terrain or where the physical placement is difficult. WSNs can combine various readings and computations over a very large area of observation to impart aggregated values in different formats and with different observed parameters. Thus, they make it possible to monitor real-world events to an unprecedented level of granularity. However, distinctive features, such as limited energy and memory or unreliable communication, bring up many problems for scientists and engineers working in this field. The ideal WSN should be scalable, low cost, less power-consuming, efficient in data gathering, reliable and accurate, and above all, maintenance-free.

Sensor nodes (SNs), which are deployed over a sensor field, organize themselves into a network, sense real-world phenomena, and forward observed measurements back to base stations or sinks. As shown in Figure 2.1, the SNs sense an event and forward their data toward the sink through other SNs. Data delivery to a sink may include a large number of wireless hops among the networked set of small, resource-limited SNs. Data gathered at the sink from various SNs has to be pieced together into meaningful information; therefore, it is important that all the SNs are well synchronized. This makes the problem of data gathering and dissemination in WSNs a real challenge for researchers. SNs are subject to frequent failures because of the operational environment as well as energy and memory constraints. It is important for a large-scale deployment of WSNs that the network provide reliable, robust, and accurate measurements. WSNs should be self-healing so that critical information is delivered promptly despite node failures.

SNs can have dense and large-scaled deployment due to their small size and low cost. Consequently, monitoring can be done more efficiently and at a higher sensing resolution compared

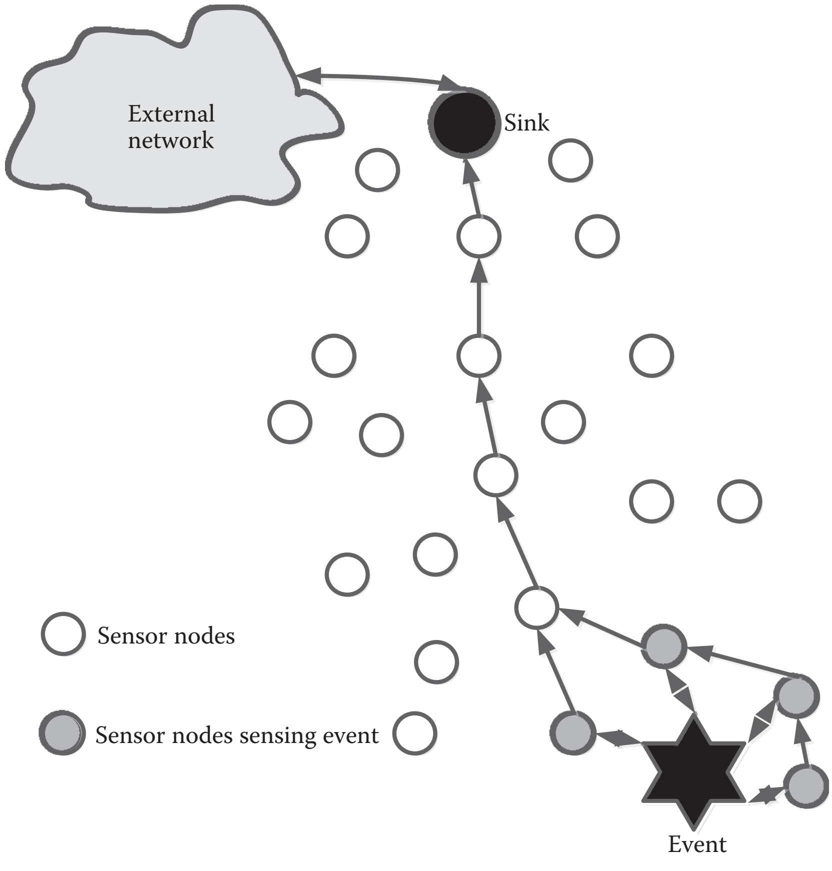


Figure 2.1 Scenario of WSN deployment.

with traditional sensor systems. The previously unobservable becomes observable. Abstracting simple sensor readings to interrelated events allows for the guarding of complex processes. Due to intelligence and collaboration between the individual nodes, the network (by itself) can carry out complex tasks related to observing.

The dense and large deployment of sensor nodes creates a large amount of redundant sensor data. The transmission of such redundant data consumes precious resources from the network and nodes. Data aggregation is an effective technique in this context because it reduces the number of packets to be sent to the sinks by aggregating similar packets. Data aggregation not only saves energy but also increases network life. In this chapter, we present some data aggregation and data-gathering algorithms proposed by several researchers and academicians.

2.2 Data Aggregation

Data aggregation is defined as the process of aggregating the data from multiple sensors to eliminate redundant transmission and provide fused information to the base station. Data latency and accuracy are important in many applications such as environmental monitoring, in which the freshness of data is also an important factor. It is critical to develop energy-efficient and fast data-aggregation algorithms. Aggregation can be done using two basic approaches. In the first approach, every sensor node sends the sensed values to the sink. After receiving all messages, the sink computes the aggregated value. In the second approach, all the nodes send the sensed values to their neighbor/parent. The parent node has to wait for its children before computing the aggregate to be forwarded to the sink. The amount of transmitted data depends on the type of aggregate function.

2.3 Data Aggregation and Data Gathering in WSNs

Various protocols proposed for aggregation, data gathering, and routing are interdependent and some authors have used the terms interchangeably. Routing in WSNs takes into consideration data aggregation at some nodes and accordingly decides packet routing. Similarly, for data aggregation, the routing protocol used underneath plays a vital role. Data generated by different sensors can be jointly processed while being forwarded toward the sink. This data generated by different nodes can be fused together (data fusion), processed locally, and any redundancy can be removed before transmission. It is important that for data fusion or other data aggregation operations, the WSNs should be time-synchronized. Data aggregation techniques are closely related to the way data is gathered at SNs as well as to how packets are routed through the network. Data aggregation has a significant effect on energy consumption and overall network energy consumption.

Data aggregation is the simplest in-network processing activity in which data from different nodes is combined into a single entity. In-network aggregation is the global process of gathering and routing information through a multihop network, processing data at intermediate nodes with the objective of reducing resource consumption (in particular energy), thereby increasing network lifetime [1]. There are two ways in which data from different sources can be combined. The first method combines data packets from different sources into one data packet that is smaller compared with the combined size of the data packets from all sources. The reduction in size of the data packet transmitted into the network is energy-conserving because smaller-sized aggregated packets are transmitted. However, this type of aggregation is lossy aggregation and sometimes results in the loss of granularity of the sensed observations. The second method combines data packets from different sources into one bigger packet. The packets' overheads in this case are reduced and the granularity of data is preserved.

Data-gathering protocols are formulated for configuring the network and collecting information from the desired environment [2]. In each round of the data-gathering protocol, data from the nodes needs to be collected and transmitted to the sink [3], where the end user can access the data. A simple way of doing this is by aggregating (sum, average, min, max, count) the data originating from different nodes [4]. A more elegant solution is data fusion, which can be defined as a combination of several unreliable data measurements to produce a more accurate signal by enhancing the common signal and reducing the uncorrelated noise. Sensor nodes use different data aggregation techniques to achieve energy efficiency. The aim is the efficient transmission of all data to the base station so that the lifetime of the network is maximized. Existing data-gathering protocols [5] can be classified into different categories based on the network topology and on routing protocols [3,6], which are aimed at power saving and prolonging network lifetime [7,8].

2.4 Protocols for Data Aggregation and Data Gathering

In the literature, there are different heuristics to a Steiner tree problem, which is a well-known NP hard problem. A network represented by graph; $G = (V,E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of SNs, and E are the edges that represent the connections among them with a cost associated with each edge. The problem is how to build a minimal cost tree that connects all source nodes $S = \{s_1, s_2, \ldots, s_m\}$, $S \subseteq V$, and the sink node. The cost of the Steiner tree is the sum of the costs of its edges. Some proposals [9,10] that have tackled this problem require large amounts of messages to set up a routing tree and consequently lead to high energy consumption in the SNs. There are distributed approaches to solve this problem, such as the shortest path tree (SPT) [11], center-at-nearest source tree (CNS) [12], greedy incremental tree (GIT) [13], and information fusion–based role assignment (InFRA) [14]. However, none of these approaches were designed with fault-tolerance in mind. SPT is the simplest strategy to build a routing tree in a distributed fashion. In this approach, every node that detects an event reports its collected information by using the shortest path to the sink. Information fusion occurs whenever paths overlap (opportunistic information fusion).

In the CNS algorithm [12], every node that detects an event sends its information to a specific node, called an aggregator, by using the shortest path. The aggregator is the closest node to the sink (in hops) that detects an event. In the GIT strategy [13], when the first event is detected, the nodes send their information similar to the SPT and, for every new event, the information is routed using the shortest path to the current tree. There is a new aggregation point every time a new branch is created. Some practical issues make GIT unaffordable for WSNs. For example, each node needs to identify the shortest path to all nodes in the network. The communication cost to create this infrastructure is $O(n^2)$. Furthermore, the space needed to store this information at each node is $O(Dn)$, where D is the number of hops in the shortest path connecting the farthest node $v \in V$ to the sink node (network diameter). After the initial phase, the algorithm needs $O(m\ n)$ messages to build the routing tree.

The InFRA algorithm [14] builds a cluster for each event; including only those nodes that were able to detect it. Then, cluster heads merge the data within the cluster and send the results to the sink node. The InFRA algorithm aims to build a SPT that maximizes this information fusion. Thus, once clusters are formed, cluster heads choose the shortest path (to the sink node) that maximizes information fusion by using the aggregated coordinators' distance. The disadvantage of the InFRA algorithm is that at each new event that arises in the network, the information about the event must be flooded throughout the network to inform other nodes about the occurrence of the new event.

Pattem et al. [15] proposed a model to describe the spatial correlation in terms of joint entropy. They analyzed a symmetric line network with different degrees of correlation among neighboring nodes. The authors showed that, for uncorrelated data, the best routing strategy was to forward packets along the shortest paths. In case of correlated information, the data is aggregated as soon as possible and a single aggregated packet is sent to the sink along the shortest path. In a study by Zhu et al. [16], the effect of data correlation on energy expenditure of data distribution protocols was studied. The focus was on various energy-aware data aggregation trees under different network conditions such as node density, source density, source distribution, and data-aggregation degree.

Chen et al. [17] proposed an energy-efficient protocol for aggregator selection (EPAS) for single-level aggregation. The optimal number of aggregators with generalized compression and power consumption models were derived and fully distributed algorithm hierarchical EPAS, an extension of EPAS, were presented.

A tree-based aggregation algorithm that exploits data correlation based on a shallow length tree (SLT), which unifies the properties of the minimum Steiner tree (MST) and the SPT, was presented by von Rickenbach and Wattenhofer [18]. In an SLT, the total cost of the tree is only a constant factor larger than that of the MST, whereas the distances (and delays thereof) between any node and the sink are only a constant factor larger than the shortest paths. Cristescu et al. [19] analyzed the aggregation properties of a tree structure that is based on an SPT of nodes close to the sink node, whereas nodes that are further away are connected to the leaves of the SPT via paths found by an approximation algorithm for the traveling salesman problem. In studies by Albert et al. [20], Dasgupta et al. [21], and Ding et al. [22], the ways through which the sink organizes routing paths to evenly and optimally distribute energy consumption while favoring the aggregation of data at the intermediate nodes were investigated. Dasgupta et al. [21] used linear programming to compute aggregation topologies by taking into account the residual energy of each node.

Hong and Kim [23] have suggested an integrated gateway node control protocol (IGCP). An algorithm, namely, the integrated gateway node (IGN) algorithm that compensates for the vulnerability of both hierarchical and flat structures in the network has been proposed. The study suggests a mixed algorithm with virtual gateway nodes, which include both the advantages of existing hierarchical structure algorithms and flat structure algorithms in WSN. The algorithm forms virtual gateway nodes consisting of several nodes such as the cluster of a hierarchical structure routing protocol and allows flat structure routing protocols between virtual nodes. The suggested algorithm communicates with the application of flat structure–type protocols after bundling up several nodes like the hierarchical structure cluster and making them work as one node. Virtual gateway nodes allow efficient energy management because it not only increases energy efficiency but also creates virtual nodes. It also makes up for the disadvantages of existing hierarchical structure routing protocols by allowing even energy use of each node as well as performing data-aggregation and in-network processing that are characterized in existing WSN routing protocols.

Nodes in the network have been exploited as aggregation points for optimal performance. Al-Karaki et al. [24] present exact and approximate algorithms to find the minimum number of aggregation points to maximize the network lifetime. Algorithms use a fixed virtual wireless backbone that is built on top of the physical topology. The tradeoffs between energy savings and the potential delay involved in the data-aggregation process have also been studied. Studies by Hartl and Li [25] and Solis and Obraczka [26] focused on the nodes that should be entrusted with the transmission of the sensed values, whereas in a study by Erramilli et al. [27] the emphasis was put on the proper scheduling of sleeping/active periods. Optimal paths are calculated in a centralized manner at the sink by exploiting different assumptions on the data correlation and selecting the best aggregation points using cost functions [28].

An energy-efficient data-gathering algorithm for prolonging the lifetime of WSNs has been proposed by Zhu et al. [29]. The authors have proposed a routing algorithm, called energy-efficient routing algorithm to prolong lifetime (ERAPL), that is able to dramatically prolong network lifetime while efficiently spending energy. In ERAPL, a data-gathering sequence (DGS) to avoid mutual transmission and loop transmission among nodes is first constructed. Each node proportionally transmits traffic to the links confined in the DGS.

A novel data-gathering scheme called data-aggregating ring (DAR) has been proposed by Bi et al. [30]. In the DAR scheme, all sensor nodes are classified according to the number of hop counts to the sink (hop grades). The nodes from different hop grades, ordered in a certain sequence, would spend different amounts of time taking charge of gathering the data packets from the nodes in other hop grades and transmitting them to the sink directly by one hop, respectively. The sequence is elaborated according to the traffic characteristic and routing strategy of the given network to balance the workloads between the nodes in different hop grades. As a result, the nodes at a distance of only one hop to the sink tend to consume an equal amount of energy as those with more hops upon delivering data. Therefore, the DAR scheme can nearly balance the energy consumption over a whole network range, increase energy efficiency and extending network lifetime notably.

Chuan-Ming Liu et al. [31] argue that a cluster-based architecture is an effective architecture for data-gathering in WSNs. However, in a mobile environment, the dynamic topology poses a challenge to designing an energy-efficient data-gathering protocol. In this study, a cluster-based architecture was considered. Two distributed clustering algorithms for mobile sensor nodes, which minimize the energy dissipation for data-gathering in a wireless mobile SN, have been proposed. There are two steps in the clustering algorithm: cluster-head election step and cluster formation step. Two distributed algorithms proposed for cluster-head election are the algorithm of cluster-head election by counting (ACE-C) and the algorithm of cluster-head election with location (ACE-L). Considering the effect of node mobility, a mechanism has been proposed (i.e., clusters with mobility or CM) for sensor nodes to select a proper cluster-head to join the cluster formation. The CM mechanism is shown to achieve better performance in terms of energy consumption and system lifetime when the sensor nodes are capable of mobility. Proposed clustering algorithms achieve the following three objectives: (1) there is at least one cluster-head elected, (2) the number of cluster-heads generated is uniform, and (3) all the generated clusters have the same cluster size.

Steiner points grid routing [32] reduces the total energy consumption for data transmission between the source node and the sink node. A virtual grid structure is constructed based on the square Steiner trees [33]. Once the sensor nodes are deployed in the sensor field, the sink node starts to construct the grid structure. The sink divides the plane into a grid of cells. Cross-points of the grid are the dissemination points (DPs). The size of the cells, denoted as α, is determined by the sink such that DPs are not within direct transmission range (the sink is the first DP). Recognizing its own position and the size of each cell, the sink is able to send a data request (in the form of a data announcement message) to each adjacent DP in the grid. Any node that is within the target region of a received QUERY message stores the appropriate routing information and starts to send the sensed data (in the form of a DATA message) to the sink. The routing information contains the appropriate upstream dissemination nodes (DNs) through which DATA messages will be forwarded. DN will find the appropriate path to transmit the DATA message depending on which DP it belongs to.

Tree-based schemes for real-time or time-constrained applications have been proposed [34,35]. An approach that relies on the construction of connected dominating sets has been proposed by Gupta et al. [36]. These consist of a small subset of nodes that form a connected backbone and whose positions are such that they can collect data from any point in the network. Nodes that do not belong to these sets are allowed to sleep when they do not have data to send. Some rotation of the nodes in the dominating set is recommended for energy balancing. More algorithms on data aggregation and data gathering have presented in other studies [23–27,37,38].

The advances in sensor node architectures, such as the inclusion of multiple sensing units and other components with variable power mode capability, have made data gathering more challenging. A sensor node with multiple sensing units is usually unable to simultaneously process the data generated by multiple sensing units, thereby resulting in missed events.

The multiple sensing unit scheduling (MSUS) [39] algorithm is the first of its kind in dealing with the tasks of multiple sensing units of the same sensor node. It schedules the tasks of different sensing units based on their priority and according to the timing constraints imposed by the application, and the existing as well as predicted future tasks for all the sensing units of a sensor node. MSUS treats task timing constraints as hard requirements whereas minimizing energy consumption and missed events. This work addresses the problem of scheduling in multiple sensing units of a sensor node in which the arrival of events and the corresponding tasks are not known in advance. Therefore, MSUS first predicts the time and the type of the next task that will soon arrive, then determines the best power state for the sensor node by considering the power and timing constraints of the current and future tasks. In MSUS, the prediction of future tasks for the sensing units is based on the past history of the occurrence of events for a particular environment.

Ozgur Sanli et al. [40] present a collaborative task scheduling algorithm (CTAS), to minimize event misses and energy consumption by exploiting power modes and overlapping sensing areas of sensor nodes. CTAS enables sensor nodes to keep only a subset of their sensing units' active at any time even though each sensor node has multiple sensing units. Although some of the sensing tasks are scheduled to neighboring nodes, the degree of coverage in the network is still maintained at a specific level for each event type. The novel idea of CTAS is that it employs a two-level scheduling approach to the execution of tasks collaboratively at group and individual levels among neighboring sensor nodes. CTAS first implements coarse grain scheduling at the group level to schedule the event types to be detected by each group member. Then, CTAS performs fine-grain scheduling to schedule the tasks corresponding to the assigned event types. The coarse grain scheduling of CTAS is based on a new algorithm that determines the degree of overlapping among neighboring sensor nodes.

2.5 Energy-Efficient Clustering and Data Aggregation Protocol for Heterogeneous WSNs

A novel energy-efficient clustering and data aggregation (EECDA) [41] protocol for heterogeneous WSNs combines the ideas of energy-efficient cluster-based routing and data aggregation to achieve a better performance in terms of lifetime and stability. The EECDA protocol includes a novel cluster-head election technique and a path would be selected with the maximum sum of energy residues for data transmission instead of the path with minimum energy consumption. In their work, the authors have shown that EECDA balances the energy consumption and prolongs the network lifetime by a factor of 51%, 35%, and 10% when compared with low-energy adaptive clustering hierarchy (LEACH), energy-efficient hierarchical clustering algorithm (EEHCA), and effective data-gathering algorithm (EDGA), respectively.

The main goal of the EECDA protocol is to efficiently maintain the energy consumption of sensor nodes by involving them in a single-hop communication within a cluster. The data aggregation and fusion technique is used to reduce the number of transmitted messages to the base station to save the energy and prevent the congestion. The authors have adopted a few reasonable assumptions for implementing the protocol as follows: (i) sensor nodes are uniformly dispersed within a square field, (ii) all sensor nodes and the base station are stationary after deployment, (iii) the WSN consists of heterogeneous nodes in terms of node energy, (iv) cluster heads (CHs) perform data aggregation, and (v) the base station is not energy limited in comparison with the energy of other nodes in the network.

A novel cluster-head election technique and a path with the maximum sum of energy residual for data transmission can maintain the balance of energy consumption in the network.

2.6 A Noble Data Aggregation Algorithm for Low Latency in Wireless SNs

In their work, Tianbo Wang et al. have optimized the energy consumption and latency of data transmission. A bilayer-based data aggregation scheme [42] has been adopted in which the wireless network is divided into two layers and each layer has a different number of cluster heads optimized. The members (nodes or cluster heads) in the region of detection of each layer send data to the related head it belongs to and the head aggregates the data. First, the number of nodes in a certain cluster is calculated, and then nodes of the network are divided into two layers. Employing these strategies, as well as the new cluster head selection method in which the aggregation is done at the cluster heads, the authors have shown that energy consumption and latency can be reduced compared with the LEACH process.

The number of nodes in a certain cluster is calculated with the cover rate and overlap rate in every round. In WSN, the cluster heads selected by the LEACH process broadcast in a radius (R) to form the measuring area. The nodes in the measuring area would return the signal to the respective cluster head, so the cluster head would acquire the number of nodes in the measuring area. The broadcasting radius of the measuring area is affected by the following two factors. The broadcasting radius R designed is long enough to measure the nearby nodes' distribution area. The total measuring area of the cluster head would reach a certain cover rate of the network area so the node distribution situation in the measuring area could reflect the node distribution situation in the clustering area. As the R extends, the overlap rates of the measuring areas of the cluster heads increase accordingly. Furthermore, the energy consumption in the measuring process would also increase with the length of R. According to the two factors previously mentioned, the length analysis of the broadcasting radius has to be optimized. The bilayer-based algorithm has been designed to improve on the LEACH process. The bilayer structure is constructed according to the following algorithms:

1. Select the group heads under probability of randomness. The selected nodes have a flag that cannot be selected in the future N/Kk rounds, and it will be stored in an array in the meantime (N is the number of nodes, K is the number of group heads, and k is the number of cluster heads).
2. The normal nodes select the nodes with the shortest distance to the group head.
3. The group selects the cluster heads distribution. The selected heads compute the distance to the upper heads and the energy consumption.
4. The normal nodes select the nodes in the same group as the cluster head according to the distance and the selected heads will be stored in another array as the cluster head. Similar to the group heads, the cluster heads will assign a flag marking that will not be selected in future N/Kk rounds. The normal nodes transmit the collected data to the related head. After network construction, each layer has the optimal heads and each node has the related head with least energy consumption.

The WSN is thus divided into several clusters in which the cluster head aggregates the sensed data from the nodes. The data sensed by the nodes in a zone is aggregated and relayed to members in the upper layers (aggregators or sensors) by the CH in the respective zone. A set of clusters that monitor the same phenomenon form a group. Each group has only one group head responsible for collecting and aggregating the data from its zone.

2.7 A Scalable and Dynamic Data Aggregation Aware Routing Protocol for WSNs

Villas et al. [43] have considered the problem of constructing a dynamic and scalable structure for data aggregation in WSN that addresses the load balancing problem. A protocol called dynamic and scalable tree (DST) reduces the number of messages required for setting up a routing tree, maximizes the number of overlapping routes, and selects routes with the highest aggregation rate. The DST is a routing tree with the shortest routes (in distance) that connects all source nodes to the sink node. The routing tree created by DST does not depend on the order of events and is not held fixed along the occurrence of events. DST considers the following roles for the creation of routing infrastructure:

- Collaborator: a node that detects events
- Coordinator: a node that gathers events from collaborator nodes, aggregates data, and notifies them
- Aggregator: a node that forwards aggregated data from two or more source nodes
- Sink: a node interested in receiving data from a set of coordinator and collaborator nodes
- Relay: a node that forwards data toward the sink

The DST has four phases. In phase 1, the sensor nodes store the sink's position and the neighbor's position. This is done by the sink, which floods a configuration message from the neighbors' position (CMNP). Phase 2 consists of cluster formation and the election of a coordinator among the nodes that detected the occurrence of a new event in the network. When an event is detected by one or more nodes, the leader election algorithm is started with the nodes running for leadership (group coordinator). For this election, all nodes are eligible; however, the group leader is the node that is closest to the sink. In phase 3, when an event occurs, the coordinator sends a package to the sink node reporting its position. The sink then notifies all other coordinators of the new coordinators' position. The sink also notifies the new coordinator of the positions of the previous coordinators. The node chosen as the event coordinator in phase 2 gathers the information collected by the collaborators. Based on its position and the sink's position, the coordinator creates a straight line segment that connects itself to the sink. The sensor nodes closest to this straight line segment and to the sink are chosen to notify of the occurrence of a new event and send data to the sink node. Finally, phase 4 is responsible for creating the routing tree connecting all coordinators to the sink node and sending the collected data to the sink node.

2.8 A Hierarchical Multiparent Cluster-Based Data Aggregation Framework for WSNs

Alemu and colleagues [44] have proposed an efficient fault-tolerant data aggregation framework based on hierarchical clustering. The proposed scheme is not only an energy-efficient aggregation scheme but is also able to overcome faulty readings and detect faulty nodes. Each cluster has two heads—primary and secondary—to aggregate sensed data. Sensor nodes in each cluster have two parents wherein the secondary parent decreases the packet drop rate by overhearing those packets that fail to reach the primary head. This maximizes the communication efficiency within the cluster. Both the spatial and temporal characteristics of nodes within and across clusters have been exploited to substitute missed values based on regression analysis.

The framework considers a network with the aggregation tree formed at the initialization stage. Cluster heads (selected based on LEACH) in each cluster elect the secondary parent based on the average distance of the nodes. The two parents synchronize with each other using identical time slots so that data packets that fail to reach the primary parent can be heard and transmitted by the secondary parent without the need for any transmission from the source node.

The aggregation process has three phases. The first phase is for building an aggregation tree. It follows a top-down hierarchical flow of messaging from the sink to the leaf nodes for constructing the basic clustering structure of the system. The data collection phase is the bottom-up process of sending the sensed data back to the sink node. Based on the maximum allowable round-trip time, each node in each cluster will send their data to their cluster head; at the same time, the secondary parent will overhear it. In this process, transient error is highly minimized as the reading is being heard by either of the parents. In addition, the secondary parent can perform simple aggregation functions like MIN/MAX, and send a single value to the primary head. The third phase is an error recovery phase. One characteristic of sensor readings is their correlation within the neighboring nodes (spatial) and a high probability of repeated values within itself (temporal). Hence, at the intracluster level, data value is lost only if it is not received by both parents. In this case, the primary parent estimates the lost data from the spatially related neighboring nodes using multiple regression analysis. If data value is heard by the secondary parent, the primary parent will get the reading before performing a regression analysis.

2.9 Cluster Tree–Based Data Gathering in WSN

Network lifetime, scalability, and load balancing are important requirements for many data-gathering SN applications. The proposed scheme is an improved version that uses both cluster- and tree-based protocols to improve the performance. The proposed protocol improves the power consumption. Chhabra and Sharma [45] have proposed the combination of cluster-based and tree-based protocols. The protocol has been designed with the following assumptions:

1. Each node or sink has the ability to transmit messages to any other node and sink directly.
2. Each sensor node has a radio-powered control node that can tune the magnitude according to the transmission distance.
3. Each sensor node has the same initial power in WSNs.
4. Each sensor node has location information.
5. Every sensor node is fixed after they are deployed.
6. WSNs would not be maintained by humans.
7. Every sensor node has the same process and communication ability in WSNs, and they play the same role.
8. Wireless sensor nodes are deployed densely and randomly in a sensor field.

The setup phase is composed of cluster formation and cluster head selection. The cluster, once formed, will not be changed much but the selected cluster head may be different in each round.

During the first round, the base station first splits the network into two subclusters, and proceeds further by splitting the subclusters into smaller clusters. The whole process is repeated until the desired number of clusters is formed. At the end of this entire process, a cluster head for each cluster will be selected by the base station. After the first round, the primary cluster topology is formed; the task of cluster formation is shifted from the base station to the sensor nodes. The decision to choose a new cluster head is made locally within each cluster based on the node's weight value.

The next phase is for constructing a cluster-based tree in which the minimum spanning tree is used to compute the tree path after labeling the cluster head. After the routing mechanism has been established, every tip node transmits gathered data to nodes in the upper level. Then, the upper level nodes will fuse received data and sensed data by itself, and send these data to next upper level nodes. This proposed method has several advantages in WSNs for data gathering. It reduces power consumption by avoiding direct communication between sink and sensor nodes. Use of the threshold mechanism increases the network lifetime. The threshold mechanism protects the death of the parent node death slowly, as each node has the chance to be a parent (Figure 2.2).

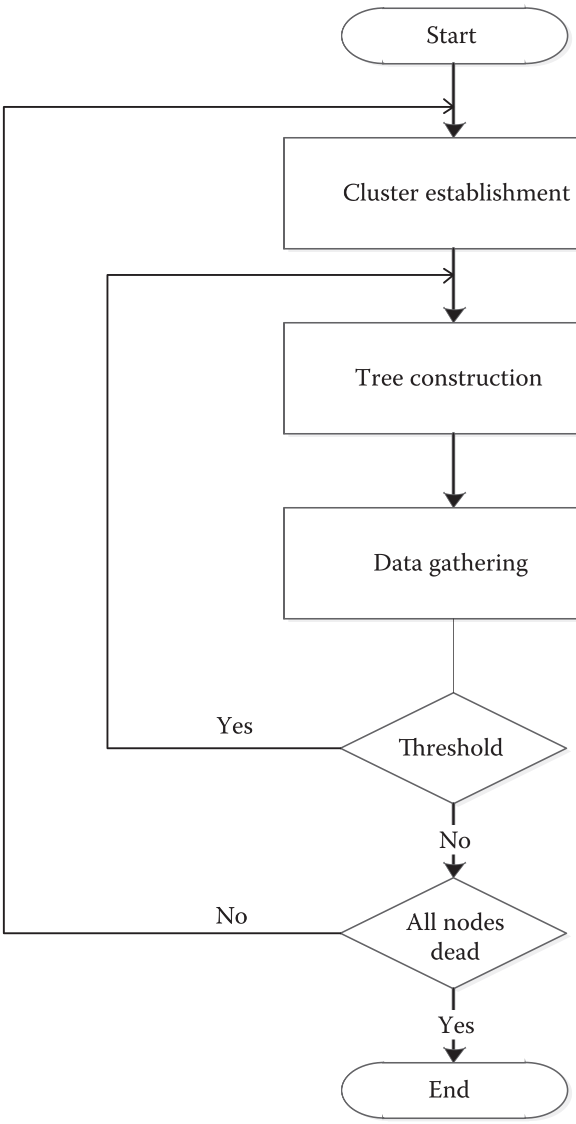


Figure 2.2 Flow chart for the cluster tree–based data-gathering protocol.

2.10 Compressive Data Gathering for Large-Scale WSNs

Luo and colleagues [46] presented a complete design to apply compressive sampling theory to sensor data gathering for large-scale WSNs. The proposed compressive data gathering is able to reduce global scale communication costs without introducing intensive computation or complicated transmission controls. The scheme used load balancing, which extends the lifetime of the network. The proposed scheme compresses sensor readings to reduce data traffic and distributes energy consumption evenly thus improving the network lifetime. In compressive data gathering (CDG), higher efficiency can be achieved by transmitting correlated sensor readings jointly.

The data-gathering process of CDG is illustrated in Figure 2.3, which is a detailed view of a small fraction of a routing tree. Leaf nodes initiate the transmission process when all nodes have acquired their readings. S2 generates a random number φ_{i2}, computes $\varphi_{i2}d_2$, and transmits the value to S1. The index i denotes the ith weighted sum ranging from 1 to M. Similarly, $s4$, $s5$, and $s6$ transmit $\varphi_{i4}d_4$, $\varphi_{i5}d_5$, and $\varphi_{i6}d_6$ to S3, respectively. Once S3 receives the three values, it computes $\varphi_{i3}d_3$, adds it to the sum of relayed values, and transmits $\sum_{j=3}^{6}\varphi_{ij}d_j$ to S1. Then, S1 computes $\varphi_{i1}d_1$ and transmits $\sum_{j=1}^{8}\varphi_{1j}d_j$. Finally, the message containing the weighted sum of all readings in a subtree is forwarded to the sink.

2.11 An In-Network Approximate Data-Gathering Algorithm Exploiting Spatial Correlation in WSNs

Several schemes have been proposed that utilize the spatial correlation of sensor readings to achieve energy savings, but most of the proposed schemes experienced high control overhead or did not fully exploit spatial correlation. To overcome these shortcomings, an in-network approximate data-gathering algorithm exploiting spatial correlation has been proposed by Huangy and colleagues [47]. The proposed algorithm consists of two phases: an in-network clustering phase and a reading streaming phase. In the in-network clustering phase, the first clusters are initialized; then,

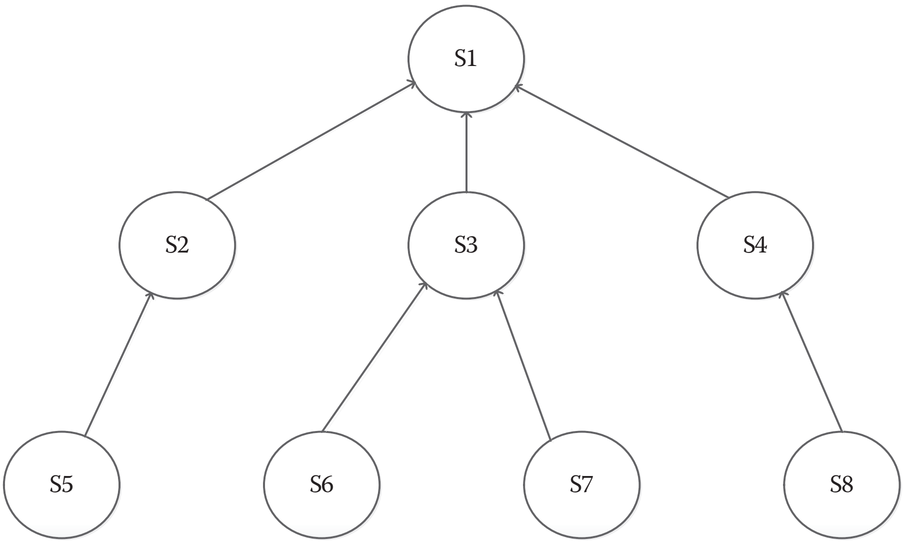


Figure 2.3 The data-gathering process.

cluster heads collect the readings from the nodes. In this phase, the in-network clustering scheme exploits the spatial correlations of sensor readings as well as cluster readings to further reduce the number of representative nodes. On the other hand, the reading streaming phase employs an adaptive cluster maintenance scheme which ensures that the user will obtain the reading answers of the desired quality despite changing sensor readings.

2.11.1 Cluster Initialization

When a user wants to monitor a phenomenon of interest p_i (e.g., temperature) and tolerates an error threshold of α, the user submits a query Q = (Query ID; p_i; α) to the sink. The sink floods the query Q into the SN and executes cluster initialization. In cluster initialization, the spatial correlation of sensor readings to group sensor nodes into disjoint clusters is exploited and representative cluster heads are determined. A data-gathering tree is created at the same time. Because the focus is not on initial clustering, the clustered aggregation technique (CAG) algorithm for initial clustering (due to its simple and distributed nature) was adopted. When the query flooding process terminates, the SN is partitioned into disjoint clusters and organized into a data-gathering tree. The cluster heads are called the initial cluster heads.

2.11.2 Cluster Reading Collection

After cluster initialization, the cluster head invokes the cluster reading collection mechanism to obtain the readings of cluster members in the same cluster, and the most representative cluster member is represented as the new cluster head. After a sensor node joins a cluster and rebroadcasts the received query, the node sends the current reading to the cluster head. With the readings of cluster members received, the cluster head can calculate the introduced reading range and tolerance range of the cluster.

2.11.3 Cluster Merging Problem

The reading of a cluster head can represent the cluster members in the same cluster or reading of a cluster head is likely to represent other nearby cluster heads. The number of representative cluster heads is further reduced by grouping multiple neighboring clusters into a larger cluster. To preserve spatial correlation, only the clusters in a merging candidate set are considered to be merged together.

2.11.4 In-Network Cluster Merging

To enable in-network cluster merging, when a cluster head i reports a reading to the sink, it attaches additional information to the report. A report is formatted as (CID$_i$; R$_i$; MaxDiff$_i$; State$_i$). CID$_i$ is the ID of the cluster head i. This is used to identify the merging candidate clusters. An intermediate node receiving a report modifies the value of the state according to its current value.

2.11.5 Reading Streaming Phase

The clusters must be dynamically readjusted due to the changes of sensor readings because the reported readings are bounded by the user-tolerable error threshold. In this phase, an adaptive cluster maintenance scheme to merge or split clusters, which offers a threshold guarantee while minimizing the number of representative cluster heads, is used. According to changes in the sensor readings, the cluster maintenance scheme readjusts the clusters in the following three cases:

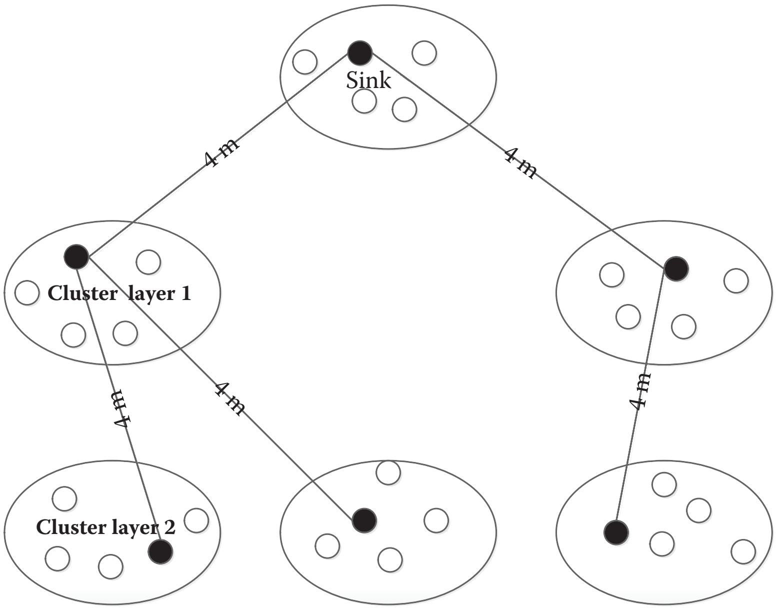
Case 1—Cluster merging: a merging check node maintains a merging table to record CIDs, R_s, MaxDiff_s, as well as the reading and the tolerance range of received reports. On the basis of the reports, the clusters are merged.

Case 2—Cluster creation: when a cluster member *A* realizes that its new reading lies outside the tolerance range of the cluster head, it creates a new cluster because the cluster head cannot be representative of it. Node *A* first elects itself as a new cluster head. MaxDiff_A is set to 0 because no other cluster member is in the cluster. Then, new cluster head *A* notifies the original cluster head of its departure by sending a leave message which upon receiving the leave message decrements the number of the cluster members by 1. Although a new cluster head causes an increase in the number of report transmissions, it is highly likely that the new cluster will soon be merged with other nearby clusters.

Case 3—Cluster splitting: when a slave cluster head *A* refreshes MaxDiff_A and it is not dominated by the master cluster, it simply resumes to report the readings to the sink at regular intervals.

EXERCISES

1. What is the difference between data fusion and data gathering?
2. Many authors have used these words interchangeably, that is, data aggregation and data gathering. What is the major difference between data aggregation and data gathering?
3. Assume that there are 4 clusters with 10 cluster nodes per cluster. One node among each cluster is chosen as the cluster head. Each cluster head is capable of transmitting the packets directly to the sink. Assuming that the transmission and per bit for 1 m is 50 nJ. The reception and processing energy per bit is 50 and 0.5 nJ, respectively. The distance between the sink and cluster heads is 4 m. The length of the aggregated data packet is 64 bytes. Calculate the energy consumed by the cluster head in 5 min if four packets of 36 bytes in size are received in 1 s from all its child nodes.
4. Assume the scenario of question 3. Calculate the lifetime (number of rounds, transmission rounds) and number of total packets the cluster head will transmit with an initial energy of 2 J.
5. Considering the energy consumption of sensor nodes given in question 3, suppose the network topology is as given in the figure in which each cluster has five sensor nodes:



i. Calculate the energy consumed by clusters of layers 2 and 3 if the sensing event is continuous for 10 min.

ii. If all the cluster heads can transmit directly to the sink and 64-byte packets are received from cluster nodes by each cluster head. The cluster head combines these packets to form one bigger packet in which the aggregated packet size is 256 bytes. Which is more energy-consuming in a network?
   a. To transmit directly to sink by each cluster head, where layer 2 cluster heads are at a distance of 6 m from the sink.
   b. Layer 2 cluster heads transmit their packet to layer 1 cluster heads, which aggregate the packets received (aggregated packet size is 256 bytes).

References

1. E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi. In-network aggregation techniques for wireless sensor networks: a survey. *IEEE Wireless Communications*, Vol. 14, No. 2, pp. 70–87, April 2007.
2. P. Mohanty, S. Panigrahi, N. Sarma, and S. S. Satapathy. Security issues in wireless sensor network data gathering protocols: a survey. *Journal of Theoretical and Applied Information Technology*, pp. 14–27, 2010.
3. J. Norman, J. P. Joseph, and P. P. Roja. A faster routing scheme for stationary wireless sensor networks—A hybrid approach. *International Journal of Ad Hoc, Sensor and Ubiquitous Computing*, Vol. 1, Issue 1, pp. 1–10, 2010.
4. M. R. E. Jebarani and T. Jayanthy. An analysis of various parameters in wireless sensor networks using adaptive FEC technique. *International Journal of Ad Hoc, Sensor and Ubiquitous Computing*, Vol. 1, Issue 3, pp. 33–43, 2010.
5. P. Samundiswary, D. Sathian, and P. Dananjayan. Secured greedy perimeter stateless routing for wireless sensor networks. *International Journal of Ad Hoc, Sensor and Ubiquitous Computing*, Vol. 1, Issue 2, pp. 9–20, 2010.
6. M. P. Singh and Md. Z. Hussain. A top-down hierarchical multi-hop secure routing protocol for wireless sensor networks. *International Journal of Ad Hoc, Sensor and Ubiquitous Computing*, Vol. 1, Issue 2, pp. 33–52, 2010.
7. J. Liang, J. Wang, and J. Chen. A delay-constrained and maximum lifetime data gathering algorithm for wireless sensor networks. *5th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN '09)*, pp. 148–155, December 2009.
8. S. K. Narang, G. Shen, and A. Ortega. Unidirectional graph-based wavelet transforms for efficient data gathering in sensor networks. *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP 2010)*, pp. 2902–2905, March 2010.
9. G. Robins and A. Zelikovsky. Improved Steiner tree approximation in graphs. *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2000)*, Philadelphia, PA, USA, pp. 770–779, 2000.
10. S. Hougardy and H. J. Prömel. A 1.598 approximation algorithm for the Steiner problem in graphs. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '99)*, Philadelphia, PA, USA, pp. 448–453, 1999.
11. B. Krishnamachari, D. Estrin, and S. B. Wicker. The impact of data aggregation in wireless sensor networks. *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCSW '02)*, Washington, DC, USA, pp. 575–578, 2002.
12. A. Roumy and D. Gesbert. Optimal matching in wireless sensor networks. *IEEE Journal on Selected Topics in Signal Processing*, Vol. 1, No. 4, December 2007.
13. F. Bauer and A. Varma. Distributed algorithms for multicast path setup in data networks. *IEEE/ACM Transaction on Networking*, Vol. 4, pp. 181–191, 1996.
14. E. F. Nakamura, H. A. B. F. de Oliveira, L. F. Pontello, and A. A. F. Loureiro. On demand role assignment for event-detection in sensor networks. *Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC '06)*, Washington, DC, USA, pp. 941–947, 2006.

15. S. Pattem, B. Krishnamachari, and R. Govindan. The impact of spatial correlation on routing with compression in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, Vol. 4, No. 4, April 2008.
16. Y. Zhu, K. Sundaresan, and R. Sivakumar. Practical limits on achievable energy improvements and useable delay tolerance in correlation aware data gathering in wireless sensor networks. *Proceedings of the IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON 2005)*, pp. 328–329, September 2005.
17. Y. P. Chen, A. L. Liestman, and J. Liu. A hierarchical energy efficient framework for data aggregation in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, Vol. 55, Issue 3, pp. 789–796, May 2006.
18. P. von Rickenbach and R. Wattenhofer. Gathering correlated data in sensor networks. *Proceedings of the ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC 2004)*, pp. 60–66, October 2004.
19. R. Cristescu, B. Beferull-Lozano, M. Vetterli, and R. Wattenhofer. Network correlated data gathering with explicit communication: NP-completeness and algorithms. *IEEE/ACM Transactions on Networking (TON)*, Vol. 14, No. 1, pp. 41–54, February 2006.
20. H. Albert, R. Kravets, and I. Gupta. Building trees based on aggregation efficiency in sensor networks. *Elsevier Ad Hoc Networks*, Vol. 5, No. 8, pp. 1317–1328, 2007.
21. K. Dasgupta, K. Kalpakis, and P. Namjoshi. An efficient clustering based heuristic for data gathering and aggregation in sensor networks. *IEEE Wireless Communications and Networking (WCNC 2003)*, Vol. 3, pp. 1948–1953, March 2003.
22. M. Ding, X. Cheng, and G. Xue. Aggregation tree construction in sensor networks. *Proceedings of the 58th IEEE Vehicular Technology Conference (VTC 2003)*, Vol. 4, pp. 2168–2172, October 2003.
23. S.-H. Hong and B-K. Kim. An efficient data gathering routing protocol in sensor networks using the integrated gateway node. *IEEE Transactions on Consumer Electronics*, Vol. 56, Issue 2, pp. 627–632, May 2010.
24. J. N. Al-Karaki, R. Ul-Mustafa, and A. E. Kamal. Data aggregation in wireless sensor networks: Exact and approximate algorithms. *Proceedings of the IEEE Workshop on High Performance Switching and Routing (HPSR 2004)*, pp. 241–245, April 2004.
25. G. Hartl and B. Li. Infer: A Bayesian approach towards energy efficient data collection in dense sensor networks. *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS 2005)*, pp. 371–380, June 2005.
26. I. Solis and K. Obraczka. Isolines: Energy-efficient mapping in sensor networks. *Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC 2005)*, pp. 379–385, June 2005.
27. V. Erramilli, I. Matta, and A. Bestavros. On the interaction between data aggregation and topology control in wireless sensor networks. *Proceedings of the 1st IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004)*, pp. 557–565, October 2004.
28. H. O. Tan and I. Korpeoglu. Power efficient data gathering and aggregation in wireless sensor networks. *ACM SIGMOD Record*, Vol. 32, No. 4, pp. 66–71, December 2003.
29. Y. Zhu, W. Wu, J. Pan, and Y. Tang. An energy-efficient data gathering algorithm to prolong lifetime of wireless sensor networks. *Computer Communications*, Vol. 33, pp. 639–647, 2010.
30. Y. Bi, N. Li, and L. Sun. DAR: An energy-balanced data-gathering scheme for wireless sensor networks. *Computer Communications*, pp. 2812–2825, 2007.
31. C.-M. Liu, C.-H. Lee, and L.-C. Wang. Distributed clustering algorithms for data-gathering in wireless mobile sensor networks. *Journal of Parallel and Distributed Computing*, Vol. 67, Issue 11, pp. 1187–1200, 2007.
32. C.-K. Liang, J.-D. Lin, and C.-S. Li. Steiner points routing protocol for wireless sensor networks. *5th International Conference on Future Information Technology (FutureTech)*, pp. 1–5, May 2010.
33. C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M.B. Srivastava. Topology management for sensor networks: exploiting latency and density. *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 135–145, 2002.