2006

Data aggregation techniques in sensor networks: A survey

Ramesh Rajagopalan
Syracuse University, Department of Electrical Engineering and Computer Science, rarajago@syr.edu

Pramod K. Varshney
Syracuse University, Department of Electrical Engineering and Computer Science, varshney@syr.edu

Data aggregation techniques in sensor networks: A survey

Ramesh Rajagopalan and Pramod K. Varshney

Department of Electrical Engineering & Computer Science
Syracuse University
Syracuse, NY, USA 13244
e-mail: *{rarajago, varshney}@syr.edu*

Abstract

Wireless sensor networks consist of sensor nodes with sensing and communication capabilities. We focus on data aggregation problems in energy constrained sensor networks. The main goal of data aggregation algorithms is to gather and aggregate data in an energy efficient manner so that network lifetime is enhanced. In this paper, we present a survey of data aggregation algorithms in wireless sensor networks. We compare and contrast different algorithms on the basis of performance measures such as lifetime, latency and data accuracy. We conclude with possible future research directions.

1. INTRODUCTION

Wireless sensor networks (WSNs) have been used for numerous applications including military surveillance, facility monitoring and environmental monitoring. Typically WSNs have a large number of sensor nodes with the ability to communicate among themselves and also to an external sink or a base-station [1, 2]. The sensors could be scattered randomly in harsh environments such as a battlefield or deterministically placed at specified locations. The sensors coordinate among themselves to form a communication network such as a single multi-hop network or a hierarchical organization with several clusters and cluster heads. The sensors periodically sense the data, process it and transmit it to the base station. The frequency of data reporting and the number of sensors which report data usually depends on the specific application. A comprehensive survey on wireless sensor networks is presented in [3].

Data gathering is defined as the systematic collection of sensed data from multiple sensors to be eventually transmitted to the base station for processing. Since sensor nodes are energy constrained, it is inefficient for all the sensors to transmit the data directly to the base station. Data generated from neighboring sensors is often redundant and highly correlated. In addition, the amount of data generated in large sensor networks is usually enormous for the base station to process. Hence, we need methods for combining data into high quality information at the sensors or intermediate nodes which can reduce the number of packets transmitted to the base station resulting in conservation of energy and bandwidth. This can be accomplished by 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 aggregation usually involves the fusion of data from multiple sensors at intermediate nodes and transmission of the aggregated data to the base station (sink). In the rest of the paper, we use the term data aggregation to denote the process of data gathering with aggregation. We also use the term sink to represent the base station.

Data aggregation attempts to collect the most critical data from the sensors and make it available to the sink in an energy efficient manner with minimum data latency. Data latency is important in many applications such as environment monitoring where the freshness of data is also an important factor**.** It is critical to develop energy efficient data aggregation algorithms so that network lifetime is enhanced. There are several factors which determine the energy efficiency of a sensor network such as network architecture, the data aggregation mechanism and the underlying routing protocol. In this paper, we describe the influence of these factors on the energy efficiency of the network in the context of data aggregation. We now present a formal definition of energy efficiency.

Energy Efficiency*:* The functionality of the sensor network should be extended as long as possible. In an ideal data aggregation scheme, each sensor should have expended the same amount of energy in each data gathering round. A data aggregation scheme is energy efficient if it maximizes the functionality of the network. If we assume that all sensors are equally important, we should minimize the energy consumption of each sensor. This idea is captured by the network lifetime which quantifies the energy efficiency of the network.

Network lifetime, data accuracy, and latency are some of the important performance measures of data aggregation algorithms. The definitions of these measures are highly dependent on the desired application. We now present a formal definition of these measures.

Network lifetime*:* Network lifetime is defined as the number of data aggregation rounds till α% of sensors die where α is specified by the system designer. For instance, in applications where the time that all nodes operate together is vital, lifetime is defined as the number of rounds until the first sensor is drained of its energy. The main idea is to perform data aggregation such that there is uniform energy drainage in the network. In addition, energy efficiency and network lifetime are synonymous in that improving energy efficiency enhances the lifetime of the network.

Data accuracy*:* The definition of data accuracy depends on the specific application for which the sensor network is designed. For instance, in a target localization problem, the estimate of target location at the sink determines the data accuracy.

Latency*:* Latency is defined as the delay involved in data transmission, routing and data aggregation. It can be measured as the time delay between the data packets received at the sink and the data generated at the source nodes.

The design of efficient data aggregation algorithms is an inherently challenging task. There has been intense research in the recent past on data aggregation in WSNs. In this survey paper, we present an extensive overview of several data aggregation algorithms. We first present the basic functionality of the specific algorithm being described and its distinct features. We then discuss the performance of the algorithm and compare it with other similar approaches.

The rest of the paper is organized as follows. In Section 2, we categorize different data aggregation protocols based on the network architecture involved in data aggregation. Section 3 describes network flow based data aggregation protocols. In Section 4, we present Quality of Service (QOS) aware data aggregation protocols designed to guarantee QOS metrics such as end-to-end reliability and information throughput. Section 5 describes the tradeoffs involved in different data aggregation protocols. Section 6 discusses data aggregation protocols which address security issues involved in data transmission. Section 7 provides some concluding remarks and future research directions.

2. DATA AGGREGATION PROTOCOLS BASED ON NETWORK ARCHITECTURE

The architecture of the sensor network plays a vital role in the performance of different data aggregation protocols. In this section, we survey several data aggregation protocols which have specifically been designed for different network architectures.

2.1 Flat networks

In flat networks, each sensor node plays the same role and is equipped with approximately the same battery power. In such networks, data aggregation is accomplished by data centric routing where the sink usually transmits a query message to the sensors, e.g, via flooding and sensors which have data matching the query send response messages back to the sink. The choice of a particular communication protocol depends on the specific application at hand. In the rest of this subsection, we describe these protocols and highlight their advantages and limitations.

2.1.1 Push diffusion

In the push diffusion scheme, the sources are active participants and initiate the diffusion while the sinks respond to the sources. The sources flood the data when they detect an event while the sinks subscribe to the sources through enforcements. The *sensor protocol for information via negotiation* (SPIN) [4] can be classified as a push based diffusion protocol. The two main features of SPIN are negotiation and resource adaptation. For successful data negotiation, sensor nodes need a descriptor to succinctly describe their observed data. These descriptors are defined in SPIN as *metadata*. The format of the metadata is application specific. For instance, in area coverage problems, sensors that cover disjoint regions can use their unique ID as metadata.

The initiating node which has new data advertises the data to the neighboring nodes in the network using the metadata. A neighboring node which is interested in this kind of data, sends a request to the initiator node for data. The initiator node responds and sends data to the sinks. Each node has a resource manager which keeps track of its energy usage. Each node polls its resources such as battery power before data transmission. This allows sensors to cut back on certain tasks when its energy is low. Simulation results show that SPIN performs almost identical to flooding in terms of the amount of data acquired over time. However, SPIN incurs a factor of 3.5 less energy consumption compared to flooding and is able to distribute 60% more data per unit energy compared to flooding. SPIN is also well suited for environments with mobile sensors since the forwarding decisions are based on local neighborhood information. One of the main advantages of SPIN is that topological changes are localized since each node only requires the knowledge of its single hop neighbors. The main disadvantage of SPIN is its inability to guarantee data delivery. For instance, in intrusion detection applications, if the nodes interested in the data are farther away from the source node, and the intermediate nodes are not interested in the data, then the data is not delivered to the sink nodes.

2.1.2 Two phase pull diffusion

Intanagonwiwat et al. [5] have developed an energy efficient data aggregation protocol called directed diffusion. Directed diffusion is a representative approach of two phase pull diffusion. It is a data centric routing scheme which is based on the data acquired at the sensors. The attributes of the data are utilized message in the network. Figure 1 illustrates the interest propagation in directed diffusion. If the attributes of the data generated by the source match the interest, a gradient is set up to identify the data generated by the sensor nodes. The sink initially broadcasts an interest message in the network. The gradient specifies the data rate and the direction in which to send the data. Intermediate nodes are capable of caching and transforming the data. Each node maintains a data cache which keeps track of recently seen data items. After receiving low data rate events, the sink reinforces one particular neighbor in order to attract higher quality data. Thus, directed diffusion is achieved by using data driven local rules.

Average dissipated energy which is the ratio of total energy dissipated per node to the number of distinct events seen by sinks and average delay were used as the performance metrics. Simulation results indicate that directed diffusion has significantly higher energy efficiency than an omniscient multicast scheme in which each node transmits data along the shortest path multicast tree to all sinks. The average dissipated energy in directed diffusion is only 60% of the omniscient multicast scheme. The average delay of directed diffusion is comparable to omniscient multicast. Directed diffusion is an appropriate choice for applications with many sources and few sinks. In directed diffusion, it is not necessary to maintain global network topology unlike SPIN. However, directed diffusion is not suitable for applications which require continuous data delivery to the sink.

Impact of source- destination location on directed diffusion

The performance of the data aggregation protocol in directed diffusion is influenced by factors such as the position of source and destination nodes and network topology. Krishnamachari et al. [6] have studied the impact of source-destination placement and communication network density on the energy costs associated with data aggregation. The event radius model (*ER*) and random source (*RS*) model are considered for source placement. In the *ER* model, all sources are assumed to be located within a fixed distance of a randomly chosen "event" location. In the *RS* model, a fixed number of nodes are randomly chosen to be sources.

The analytical bounds on energy costs with data aggregation show that significant energy cost saving is achieved when the sources are close together and far away from the sink. The optimal data aggregation tree can be constructed in polynomial time if the set of source nodes are connected. Simulations were performed on a 100 node network with the number of sources varying from 1 to 15 ensuring that the sources were connected. The energy gains due to data aggregation are predominant in networks with a large number of sources that are several hops away from the sink.

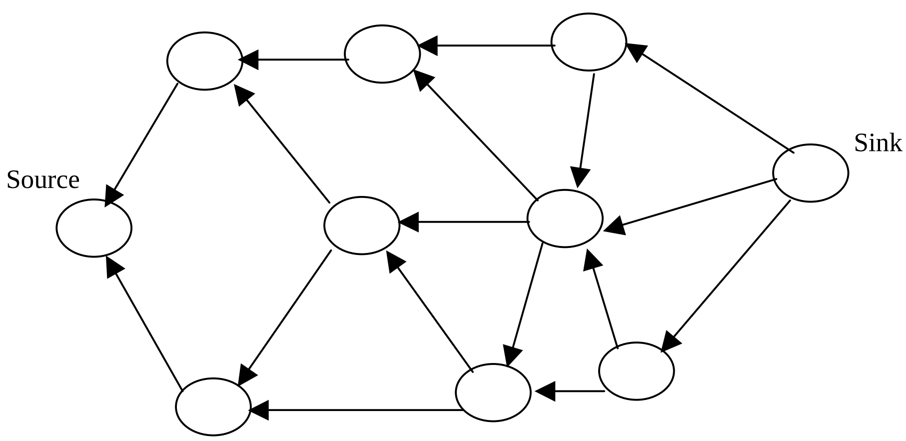

Figure 1: Interest propagation in directed diffusion.

2.1.3 One phase pull diffusion

Two phase pull diffusion results in large overhead if there are many sources and sinks. Krishnamachari et al. [7] have proposed a one phase pull diffusion scheme which skips the flooding process of directed diffusion. In one phase pull diffusion, sinks send interest messages that propagate through the network establishing gradients. However, the sources do not transmit exploratory data. The sources transmit data only to the lowest latency gradient pertinent to each sink. Hence, the reverse route (from the source to the sink) has the least latency. Removal of exploratory data transmission results in a decrease in control overhead conserving the energy of the sensors.

In [7], simulations have been performed comparing push diffusion with one phase pull diffusion. The simulation results show that one phase pull outperforms push diffusion when the source event rate is very high. However, when the sink interest rate is high push diffusion performs better than one phase pull diffusion. A wrong choice of diffusion mechanism results in excessive control overhead. For instance, when the source rate is high and the sink interest rate is low, employing push diffusion results in 80% increase in control overhead compared to one phase pull diffusion.

2.2. Hierarchical networks

A flat network can result in excessive communication and computation burden at the sink node resulting in a faster depletion of its battery power. The death of the sink node breaks down the functionality of the network. Hence, in view of scalability and energy efficiency, several hierarchical data aggregation approaches have been proposed. Hierarchical data aggregation involves data fusion at special nodes, which reduces the number of messages transmitted to the sink. This improves the energy efficiency of the network. In the rest of this subsection, we describe the different hierarchical data aggregation protocols and highlight their main advantages and limitations.

2.2.1 Data aggregation in cluster based networks

In energy constrained sensor networks of large size, it is inefficient for sensors to transmit the data directly to the sink. In such scenarios, sensors can transmit data to a local aggregator or cluster head which aggregates data from all the sensors in its cluster and transmits the concise digest to the sink. This results in significant energy savings for the energy constrained sensors. Figure 2 shows a cluster based sensor network organization. The cluster heads can communicate with the sink directly via long range transmissions or multi hopping through other cluster heads. Recently, several cluster based network organization and data aggregation protocols have been proposed. In this section we discuss three such protocols viz., *Low Energy Adaptive Clustering Hierarchy* (LEACH), *Hybrid Energy Efficient Distributed Clustering Approach* (HEED) and *clustered diffusion with dynamic data aggregation* (CLUDDA).

Heinzelman [8] et al. were the first to propose an energy conserving cluster formation protocol called LEACH. The LEACH protocol is distributed and sensor nodes organize themselves into clusters for data fusion. A designated node (cluster head) in each cluster transmits the fused data from several sensors in its cluster to the sink. This reduces the amount of information that is transmitted to the sink. The data fusion is performed periodically at the cluster heads. LEACH is suited for applications which involve constant monitoring and periodic data reporting. The two main phases involved in LEACH are: setup phase and steady state phase. The setup phase involves the organization of the network into clusters and the selection of cluster heads.

The steady state phase involves data aggregation at the cluster heads and data transmission to the sink. A predetermined fraction of nodes, f, elect themselves as the cluster head during the set up phase. A sensor node i compares a random number n between 0 and 1 with a threshold η_i. If $n > \eta_i$, the sensor node becomes a cluster head. The threshold η_i is given by

$$\eta_i = \frac{f}{1 - f\,(n \bmod (1/f))}$$

where *mod* stands for the modulus operator which returns the remainder after division. All elected cluster heads broadcast a message to all the other sensors in the network informing that they are the new cluster heads. All non-cluster head nodes which receive this advertisement decide as to which cluster they belong to based on the signal strength of the message received. LEACH employs randomization to rotate cluster heads and achieves a factor of eight improvement compared to the direct approach in terms of energy consumption. LEACH was compared with minimum transmission energy routing (MTE) in which intermediate nodes are chosen such that the sum of squared distances between adjacent nodes of the route is minimized. The simulation results show that LEACH delivers ten times more data than MTE for the same number of node deaths.

Although LEACH improves the system lifetime and data accuracy of the network, the protocol has some limitations. LEACH assumes that all sensors have enough power to reach the sink if needed. In other words, each sensor has the capability to act as a cluster head and perform data fusion. This assumption might not be valid with energy-constrained sensors. LEACH also assumes that nodes have data to send periodically. In LEACH, all nodes have the same amount of energy capacity in each election round which is based on the assumption that being a cluster head results in same energy consumption for every node. Hence, LEACH should be extended to account for node heterogeneity. In an improved version of this protocol, called LEACH-C [9], cluster formation is performed in a centralized manner by the sink. LEACH-C improves the performance of LEACH by 20 to 40 percent in terms of the number of successful data gathering rounds.

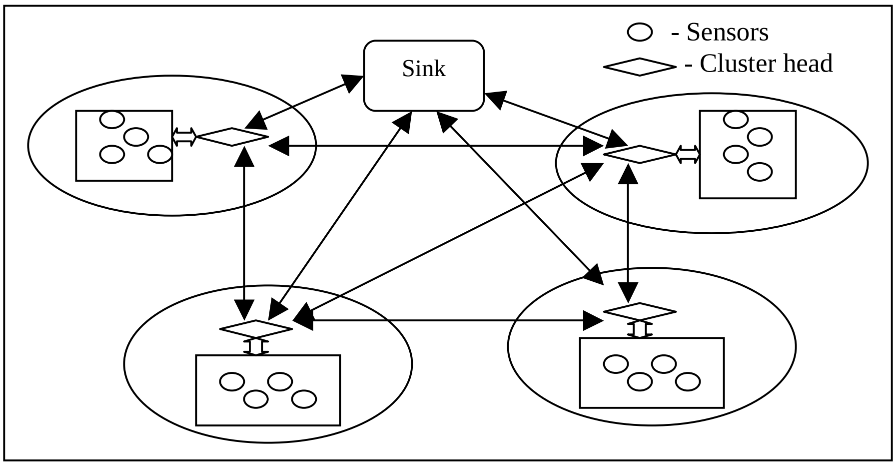

Figure 2: Cluster based sensor network. The arrows indicate wireless communication links.

Younis et al. [10] have proposed HEED whose main goal is to form efficient clusters for maximizing network lifetime. The main assumption in HEED is the availability of multiple power levels at sensor nodes. Cluster head selection is based on a combination of node residual energy of each node and a secondary parameter which depends on the node proximity to its neighbors or node degree. The cost of a cluster head is defined as its average minimum reachability power (AMRP). AMRP is the average of the minimum power levels required by all nodes within the cluster range to reach the cluster head. AMRP provides an estimate of the communication cost.

At every iteration of HEED, each node which has not selected a cluster head, sets its probability P_{CH} of becoming the cluster head as

$$P_{CH} = C \times \frac{E_{residual}}{E_{\max}}$$

where C denotes the initial percentage of cluster heads (specified by the user) , $E_{residual}$ is the estimated current residual energy of the node and E_{max} is its initial energy corresponding to a fully charged battery. Each node sends a *cluster_head_msg* where the selection status is set to tentative if P_{CH} is less than 1 or final if P_{CH} is 1. A node selects its cluster head as the node with the lowest cost (AMRP) in the set of tentative cluster heads**.** Every node then changes its probability to $\min(2 \times P_{CH}, 1)$ in the next iteration. The process repeats until every node is assigned to a cluster head.

Inter cluster communication has not been considered in HEED. The performance of HEED has been compared with generalized LEACH (gen-LEACH) proposed in [10]. In gen-LEACH, the routing protocol propagates the node residual energy throughout the network. The cluster head election probability at time t is given by

$$P_L(t) = \min\left(\left(\frac{E_i(t)}{E_{tot}}\right) \times k, 1\right)$$

where E_i is the residual energy of node i, $E_{tot} = \sum_{i=1}^{n} E_i(t)$ and k is the initial percentage of cluster heads. The protocols were simulated for varying network sizes. The simulation results show that HEED improves the network lifetime over gen-LEACH. In gen-LEACH the selection of cluster heads is random which may result in rapid death of certain nodes. However, in HEED the cluster heads are selected such that they are well distributed with minimum communication cost. In addition, the energy dissipated in clustering is less in HEED compared to gen-LEACH. This is due to the fact that gen-LEACH propagates residual energy. To conclude, HEED prolongs network lifetime and achieves a geographically well-distributed set of cluster heads.

Recently a hybrid approach [11] has been proposed which combines clustering with diffusion mechanisms. The new data aggregation scheme proposed in [11] is called clustered diffusion with dynamic data aggregation (CLUDDA). CLUDDA performs data aggregation in unfamiliar environments by including query definitions within interest messages. The interest messages of a new query initiated by the sink contains the query and also a detailed definition of the query. The query definition describes the operations that need to be performed on the data components in order to generate a proper response. This new format of the interest message has some interesting features such as interest transformation and dynamic aggregation. Interest transformation utilizes existing knowledge of queries in order to reduce the overhead in processing.

CLUDDA combines directed diffusion [5] with clustering during the initial phase of interest or query propagation. The clustering approach ensures that only cluster heads and gateway nodes which perform inter cluster communication are involved in the transmission of interest messages. This technique conserves energy since the regular nodes remain silent unless they are capable of servicing a request. In CLUDDA, the aggregation points are dynamic. The data aggregation task is not assigned to any specific group of nodes in the network. The nodes performing data aggregation change as the locations of source nodes change. Any cluster head or gateway node which has the knowledge of query definition can perform data aggregation.

An interesting feature of CLUDDA is that a query cache is maintained at the cluster heads and gateway nodes. The query cache lists the different data components that were aggregated to obtain the final data. It also contains the addresses of neighboring nodes from which the data messages originated. These addressees can be used to propagate interest messages directly to specific nodes instead of broadcasting. However, the memory requirements for this technique are yet to be investigated. The technique proposed also needs to be implemented and compared with other existing approaches.

2.2.2 Chain based data aggregation

In cluster-based sensor networks, sensors transmit data to the cluster head where data aggregation is performed. However, if the cluster head is far away from the sensors, they might expend excessive energy in communication. Further improvements in energy efficiency can be obtained if sensors transmit only to close neighbors. The key idea behind chain based data aggregation is that each sensor transmits only to its closest neighbor. Lindsey et al. [12] presented a chain based data aggregation protocol called power efficient data gathering protocol for sensor information systems (PEGASIS). In PEGASIS, nodes are organized into a linear chain for data aggregation. The nodes can form a chain by employing a greedy algorithm or the sink can determine the chain in a centralized manner. Greedy chain formation assumes that all nodes have global knowledge of the network. The farthest node from the sink initiates chain formation and at each step, the closest neighbor of a node is selected as its successor in the chain. In each data gathering round, a node receives data from one of its neighbors, fuses the data with its own and transmits the fused data to its other neighbor along the chain. Eventually the leader node which is similar to cluster head transmits the aggregated data to the sink. Figure 3 shows the chain based data aggregation procedure in PEGASIS. Nodes take turns in transmitting to the sink. The greedy chain formation approach used in [12] may result in some nodes having relatively distant neighbors along the chain. This problem is alleviated by not allowing such nodes to become leaders.

The PEGASIS protocol has considerable energy savings compared to LEACH. The distances that most of the nodes transmit are much less compared to LEACH in which each node transmits to its cluster head. The leader node receives at most two data packets from its two neighbors. In contrast, a cluster head in LEACH has to perform data fusion of several data packets received from its cluster members. The main disadvantage of PEGASIS is the necessity of global knowledge of all node positions to pick suitable neighbors and minimize the maximum neighbor distance. In addition, PEGASIS assumes that all sensors are equipped with identical battery power and results in excessive delay for nodes at the end of the chain which are farther away from the leader node. In [12], two other protocols viz., a binary chain based scheme and a three-level chain based scheme have been proposed. In the binary chain based protocol, each node transmits data to a close neighbor in a given level of the hierarchy. The nodes that receive data at each level form a chain in the next higher level of the hierarchy. At the highest level, the leader node transmits the aggregated data to the sink. In the three level scheme, the protocol starts with the formation of a linear chain among all nodes and then it divides them into G groups. Each group has N/G successive nodes of the chain where N is the total number of nodes. Only one node from each group participates in the second level of the hierarchy. The G nodes in the second level are further divided into two groups so that only three levels are maintained in the hierarchy.

Both energy efficiency and delay are considered while evaluating the performance of the above protocols. The metric is computed by multiplying the average energy cost per data gathering round with the unit delay (transmission time for a 2000 bit message) for the scheme. The performance of the algorithms was compared in terms of the *Energy*×*Delay* metric proposed in [12]. Simulation results show that the chain based binary scheme is eight times better than LEACH and 130 times better than the direct scheme for a 50m×50m network. The chain based three level scheme is 5 times better than PEGASIS and 140 times better than the direct scheme for a 100m×100m network. PEGASIS outperforms LEACH by 100 to 200 percent in terms of the number of data gathering rounds for different network sizes. No conclusions can be drawn about the optimality of a single scheme for optimizing the *Energy*×*Delay* metric. The energy costs of transmissions depend on the spatial distribution of nodes which preclude the optimality of a single scheme for all network sizes. However, experimental results indicate that the binary chain based scheme performs the best for small network sizes.

2.2.2.1 Chain construction algorithms

The effectiveness of chain based data aggregation protocols depends largely on the construction of an energy efficient chain. In this subsection, we describe some chain construction algorithms. Du et al. [13] have developed an energy efficient chain construction algorithm which employs insertion operations to add the least amount of energy consumption to the whole chain. The main focus is on energy efficient all to all broadcasting in sensor networks.

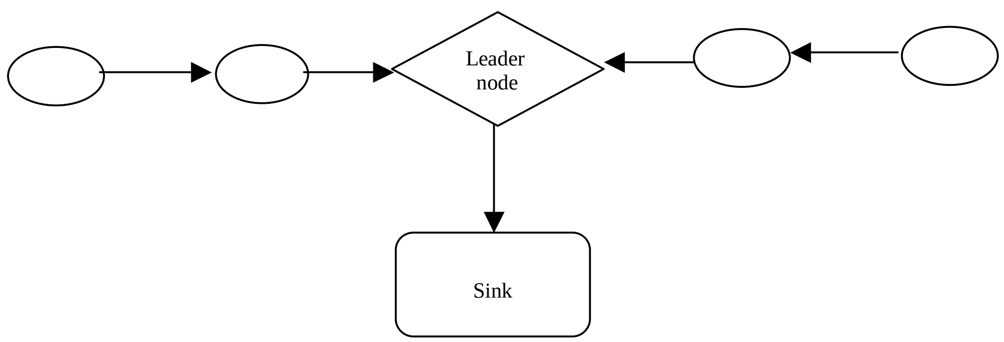

Figure 3: Chain based organization in a sensor network. The ovals indicate sensors and the arrows indicate the direction of data transmission.

A multiple chain scheme has been proposed which divides the whole network into four regions centered at the node that is closest to the center of the sensing region. For each region, a linear chain is constructed which ends at the center node. The multiple chain scheme aims to decrease the total transmission distance for all-to-all broadcasting.

In the greedy chain construction algorithm proposed in [12], the process starts with the farthest node from the sink. This node is the head of the chain. At each step, a non-chain node which is closest to the chain head is selected and appended to the chain as the new head. The procedure is repeated until all nodes are in the chain. This process does not necessarily minimize the total transmission energy. The authors in [13] have proposed a minimum total energy algorithm which constructs a chain with minimum $\sum d^2$ where d is the distance between two adjacent nodes in the chain. The chain construction starts with the node farthest from the sink as the leader. At each step, a new node is inserted such that $\sum d^2$ of the current chain with the new node increases to the minimum possible extent compared to the old chain. This new node becomes the leader. The algorithm has a complexity of $O(n^3)$ where n is the total number of nodes.

The algorithm proposed in [13] and the greedy algorithm were simulated on networks of different sizes with random sensor deployment. The number of rounds until the first node dies was used as the performance measure. The results indicate that in highly dense networks or in networks with a large distance to the sink, the two algorithms have identical performance. This is because, in dense networks, the distances between nodes are small and any node has a good chance of being selected in the chain construction. For networks with moderate density, the minimum total energy algorithm achieves 15% to 30% performance improvement compared to the greedy algorithm. In general, we need to consider factors such as density of the network and location of the sink while choosing an appropriate chain construction algorithm.

2.2.3 Tree based data aggregation

In a tree based network, sensor nodes are organized into a tree where data aggregation is performed at intermediate nodes along the tree and a concise representation of the data is transmitted to the root node. Tree based data aggregation is suitable for applications which involve in-network data aggregation. An example application is radiation level monitoring in a nuclear plant where the maximum value provides the most useful information for the safety of the plant. One of the main aspects of tree-based networks is the construction of an energy efficient data aggregation tree. In this subsection, we describe the construction of data aggregation trees.

Ding et al. [14] have proposed an energy aware distributed heuristic (EADAT) to construct and maintain a data aggregation tree in sensor networks. The algorithm is initiated by the sink which broadcasts a control message. The sink assumes the role of the root node in the aggregation tree. The control message has five fields: *ID, parent, power, status and hopcnt* indicating the sensor ID, its parent, its residual power, the status (leaf, non-leaf node or undefined state) and the number of hops from the sink. After receiving the control message for the first time, a sensor v sets up its timer to T_v. T_v counts down when the channel is idle. During this process, the sensor v chooses the node with the higher residual power and shorter path to the sink as its parent. This information is known to node v through the control message. When the timer times out, the node v increases its hop count by one and broadcasts the control message. If a node u receives a message indicating that its parent node is node v, then u marks itself as a non leaf node. Otherwise the node marks itself as a leaf node.

The process continues until each node broadcasts once and the result is an aggregation tree rooted at the sink. The main advantage of this algorithm is that sensors with higher residual power have a higher chance to become a non-leaf tree node. To maintain the tree, a residual power threshold P_{th} is associated with each sensor. When the residual power of a sensor falls below P_{th}, it periodically broadcasts help messages for T_d time units and shuts down its radio. A child node upon receiving a help message, switches to a new parent. Otherwise it enters into a danger state. If a danger node receives a hello message from a neighboring node *v* with shorter distance to the sink, it invites *v* to join the tree.

The protocol proposed in [14] was simulated on a sensor field of $160m \times 160m$. The results show that EADAT extends network lifetime and conserves more energy in comparison with routing methods without aggregation. The results also indicate that with EADAT, the average residual energy of all alive sensors decreases much more slowly compared to the scenario when no aggregation was used. Another interesting observation was regarding the variation of network lifetime with the network density. The network lifetime increases linearly with the network density. The heuristics proposed in EADAT can thus be used to construct energy efficient aggregation trees.

In applications where each sensor node has data to send to the sink in every round of communication, it is essential to maximize the network lifetime. Tan et al. [15] have proposed a power efficient data gathering and aggregation protocol (PEDAP). The goal of PEDAP is to maximize the lifetime of the network in terms of number of rounds, where each round corresponds to aggregation of data transmitted from different sensor nodes to the sink. PEDAP is a minimum spanning tree based protocol which improves the lifetime of the network even when the sink is inside the field. In contrast, LEACH and PEGASIS perform poorly when the sink is inside the sensor field. PEDAP minimizes the total energy expended in each communication round by computing a minimum spanning tree over the sensor network with link costs given by

$$C_{ij}(k) = 2 \times E_{elec} \times k + E_{amp} \times k \times d_{ij}^{\,2}$$

where $C_{ij}(k)$ is the cost of transmitting a *k* bit data packet from node *i* to node *j*, E_{elec} is the energy dissipated by the transmitter or receiver circuitry, E_{amp} is the energy dissipated by the transmit amplifier and d_{ij} is the distance between node *i* and node *j*. Prim's minimum spanning tree algorithm is employed to compute the routing paths with the sink as the root. The data packets are routed to the sink over the edges of the minimum spanning tree. Figure 4 illustrates tree based data aggregation in a sensor network.

In order to balance the load among the nodes, the residual energy of the nodes should be considered while aggregating the data. The power aware version of PEDAP (PEDAP-PA) aims to achieve this by modifying the link costs as

$$EC_{ij}(k) = C_{ij}(k) / e_i$$

where e_i is the normalized residual energy of node *i* where the normalization is with respect to the initial energy in the battery. Hence a node with a lower residual energy incurs more cost in transmission of packets to its neighbors. The cost of communication between nodes *i* and *j* is asymmetric. Consequently, for a low energy node, the cost of sending data to the sink is increased.

The PEDAP protocol requires global knowledge of the location of all nodes at the sink. The protocols operate in a centralized manner where the sink computes the routing information. The time complexity of the proposed protocols is $O(n^2)$ where *n* is the total number of sensors in the network.

The performance of the protocols proposed in [15] was compared with LEACH, PEGASIS and direct transmission. The goal was to determine the timings of node deaths in PEDAP-PA improves the lifetime of the first node by 400% while providing a similar lifetime for the last node when compared with PEGASIS. For the scenario in which the sink was placed in the center of the field, PEDAP and PEDAP-PA improve the lifetime of the last node by about two times when compared with PEGASIS and LEACH. These results indicate that if balancing the load among nodes is important, then PEDAP-PA performs the best among the alternative algorithms. PEDAP-PA is also a good choice if the lifetime of the last node is critical.

2.2.4 Grid based data aggregation

Vaidhyanathan et al. [16] have proposed two data aggregation schemes which are based on dividing the region monitored by a sensor network into several grids. They are: grid-based data aggregation and in-network data aggregation. In grid-based data aggregation, a set of sensors is assigned as data aggregators in fixed regions of the sensor network. The sensors in a particular grid transmit the data directly to the data aggregator of that grid. Hence, the sensors within a grid do not communicate with each other. In-network aggregation is similar to grid based data aggregation with two major differences viz.,

a) Each sensor within a grid communicates with its neighboring sensors.

b) Any sensor node within a grid can assume the role of a data aggregator.

terms of rounds until the last node dies. The simulation results show that LEACH and direct transmission perform the worst while PEGASIS offers a much improved performance.

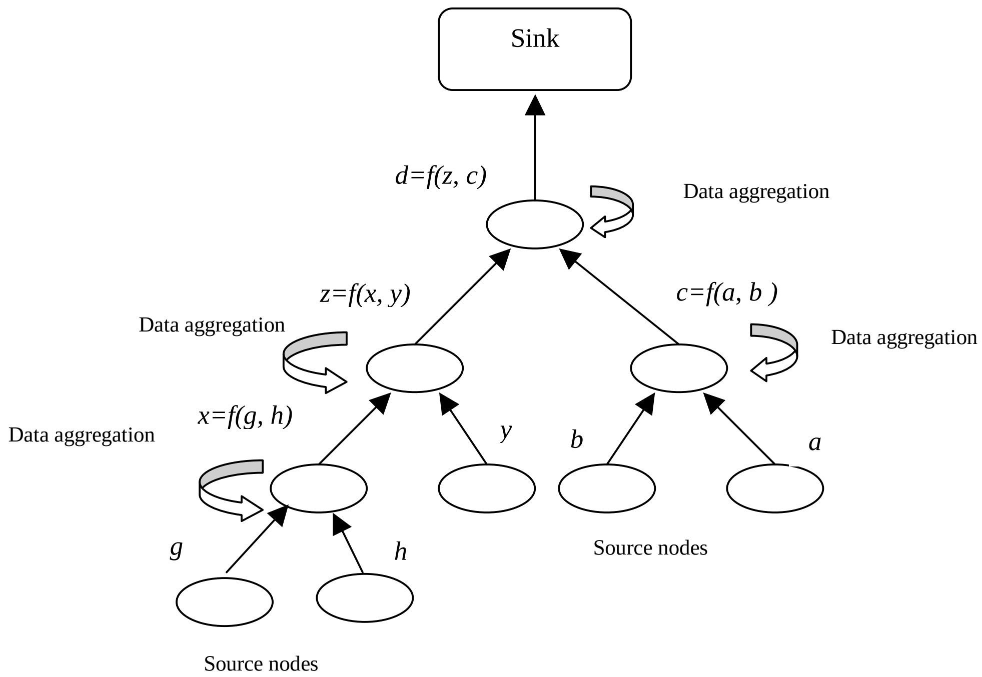

Figure 4: Minimum spanning tree based routing protocol in a sensor network. The arrows indicate the routing path and *f* (., .) is the data aggregation function.

We now describe these two data aggregation schemes in greater detail. In grid-based data aggregation, the data aggregator is fixed in each grid and it aggregates the data from all the sensors within the grid. This is similar to cluster based data aggregation in which the cluster heads are fixed. Grid-based data aggregation is suitable for mobile environments such as military surveillance and weather forecasting and adapts to dynamic changes in the network and event mobility.

In in-network aggregation, the sensor with the most critical information aggregates the data packets and sends the fused data to the sink. Each sensor transmits its signal strength to its neighbors. If the neighbor has a higher signal strength, the sender stops transmitting packets. After receiving packets from all the neighbors, the node that has the highest signal strength becomes the data aggregator. The in-network aggregation scheme is best suited for environments where events are highly localized.

Figures 5 and 6 show the in-network and grid-based data aggregation schemes respectively. From Figure 5, we observe that sensors exchange signal strengths with their neighbors to determine the in-network aggregator which is the node with the highest signal strength. On the other hand, Figure 6 shows that in grid based data aggregation, all sensors directly transmit data to a pre-determined grid aggregator. A more efficient approach would choose either the in-network or the grid-based scheme on the fly based on the type of event and its mobility. The authors in [16] have proposed a hybrid scheme which combines the salient features of the in-network and grid-based aggregation schemes. The hybrid scheme accomplishes this goal by combining the best of both the approaches. In the hybrid scheme, sensors are initially configured according to the in-network scheme. When an event occurs, the sensor with the most critical information is identified. The sensors also maintain a table of past events and the corresponding signal strengths. When a sensor detects an event, it checks its table for the previous event and identifies the nature of the event. The in-network scheme is followed if the sensor identifies the event as a localized event. If the signal strength measurements indicate that the event is mobile, it sends the information to a default aggregator which is a grid based aggregation scheme.

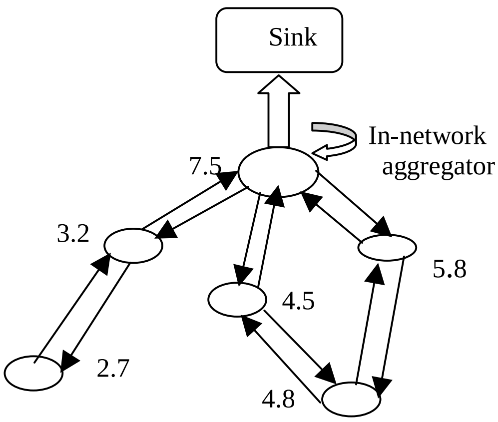

Figure 5: An in-network data aggregation scheme. The numbers indicate the signal strengths detected by the sensors. The arrows indicate the exchange of signal strengths between neighboring nodes.

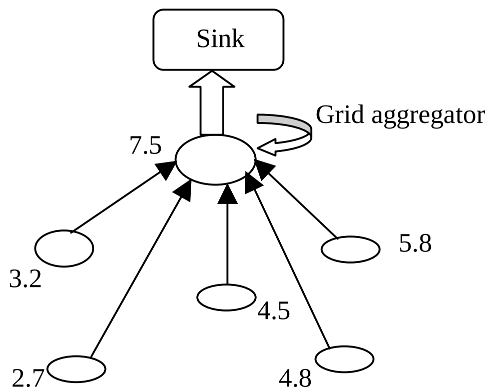

Figure 6: Grid based data aggregation. The arrows indicate the transmission of data from sensors to the grid aggregator.

Simulations were performed on a 100 node network with random deployment [16]. In terms of the data acquired (throughput), the hybrid scheme and the in-network scheme perform almost identical to the perfect aggregation scheme in which each sensor is assumed to know the best aggregator. In terms of data latency, the hybrid scheme performs much better than the no-aggregation (classic flooding) and grid based schemes. The schemes have also been compared with respect to the total energy consumption of the sensor network. The simulation results indicate that the energy consumed by the grid based scheme is a factor of 2.2 less than the no-aggregation scheme. The in-network scheme and the hybrid scheme achieve a factor of 2.4 improvement compared to the no-aggregation scheme. The results show the superiority of the aggregation schemes compared to a no-aggregation scheme. However, for a more complete performance evaluation, the schemes need to be simulated under more elaborate scenarios such as multiple event detection. Table 1 summarizes the different hierarchical data aggregation protocols and their vital characteristics. Table 2 presents some important differences between flat and hierarchical data aggregation protocols.

Table 1: Summary of hierarchical data aggregation protocols

Protocol	**Organization type**	**Objectives**	**Characteristics**
LEACH	cluster	Network lifetime: number of nodes that are alive, latency	Randomized cluster head rotation, non-uniform energy drainage across different sensors.
HEED	cluster	Lifetime: number of rounds until the first node death	Assumption: Multiple power levels in sensors. Cluster heads are well distributed. Achieves better performance than LEACH
PEGASIS	chain	Lifetime: average energy expended by a node	Global knowledge of the network is required. Considerable energy savings compared to LEACH.
Hierarchical chain based protocols	chain	Energy$\times$ delay	Binary chain based scheme is eight times better than LEACH and the three level scheme is 5 times better than PEGASIS.
EADAT	tree	Lifetime: number of alive sensors at the end of simulation time	Sink initiated broadcasting approach. It is not clear how to choose the threshold power (P_{th}) for broadcasting help messages. No comparisons made with other existing aggregation algorithms.
PEDAP-PA	tree	Lifetime: time until the death of last node	Minimum spanning tree based approach. Achieves two times performance improvement compared to LEACH, PEGASIS.

Table 2:Data aggregation in hierarchical networks versus flat networks

Hierarchical networks	Flat networks
Data aggregation performed by cluster heads or a leader node.	Data aggregation is performed by different nodes along the multi-hop path.
Overhead involved in cluster or chain formation throughout the network.	Data aggregation routes are formed only in regions that have data for transmission.
Even if one cluster head fails, the network may still be operational.	The failure of sink node may result in the break down of entire network.
Lower latency is involved since sensor nodes perform short-range transmissions to the cluster head.	Higher latency is involved in data transmission to the sink via a multi-hop path.
Routing structure is simple but not necessarily optimal.	Optimal routing can be guaranteed with additional overhead.
Node heterogeneity can be exploited by assigning high energy nodes as cluster heads.	Does not utilize node heterogeneity for improving energy efficiency.

3. NETWORK FLOW BASED DATA AGGREGATION PROTOCOLS

Though most data aggregation protocols can be classified based on network architecture, some protocols pursue a different approach in that the sensor network is represented as a graph. Such protocols can be classified as network flow based protocols, in which data aggregation is modeled as a network flow problem. The main goal of network flow based protocols is optimization of network lifetime subject to energy constraints on sensor nodes and flow constraints on information routed in the network. In this section, we describe network flow based protocols and the optimization approaches involved.

3.1 Data aggregation protocols for lifetime maximization

In this section, we discuss three data aggregation approaches viz., maximum lifetime data gathering with aggregation (MLDA), a polynomial time approximation scheme and flow optimization with data aggregation.

3.1.1 Maximum lifetime data aggregation

Kalpakis et al. [17] have studied the maximum lifetime data gathering with aggregation (MLDA) problem employing efficient data aggregation algorithms. The goal of the MLDA problem is to obtain a data gathering schedule with maximum lifetime where sensors aggregate incoming data packets.

The sensor network is modeled as a directed graph $G = (V,E)$. The edges of G have an associated capacity $f_{i,j}$ which indicates the number of packets transmitted from node i to node j. An optimal admissible flow network is obtained using integer programming with linear constraints. The integer program computes the maximum system lifetime T subject to energy constraints of the sensors and capacity constraints on the edges. A scheduling algorithm is proposed that finds a sequence of aggregation trees that can aggregate and transmit T data packets from each sensor to the sink. Figure 7 shows an admissible flow network G with 70 rounds. In the aggregation tree T of Figure 7, sensors B and D transmit one data packet to sensor A, which aggregates the incoming packets with its own packet and transmits to the sink C.

The integer programming approach involves solving a linear program with $O(n^3)$ variables and constraints where n is the total number of sensors. This approach is computationally expensive for large values of n. To alleviate this problem, a clustering based approach called greedy CMLDA has been proposed to obtain efficient data gathering schedules in large networks. Each cluster is referred to as a super-sensor. A maximum lifetime schedule is first obtained for the super-sensors which is then used to construct aggregation trees for the sensors. The initial energy of each super-sensor is equal to the sum of the initial energies of all the sensors within it. The distance between any two super-sensors S_i and S_j is assigned as the maximum distance between two nodes u and v such that $u \in S_i$ and $v \in S_j$. The maximum lifetime schedule for the super-sensors is obtained by using the MLDA algorithm. At every step of tree construction, the node with maximum residual energy is included in the tree. The time complexity of the approach is polynomial in the number of sensors which involves solving a linear program with $O(m^3)$ variables where m is the number of clusters. An incremental CMLDA heuristic has also been proposed which builds a flow network by incrementally provisioning capacities on the edges. Using this heuristic, the lifetime of the network scales linearly with the energy of the sensors.

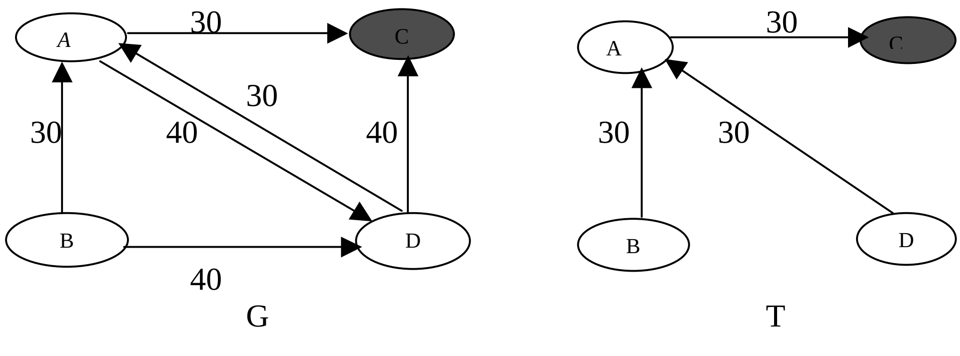

Figure 7: Illustration of an admissible flow network G with lifetime 70 rounds and an aggregation tree T with lifetime 30 rounds. The shaded nodes represent the sinks.

The algorithms in [17] were evaluated by comparing their performance with a chain based 3 level hierarchical protocol proposed by Lindsey et al. [12] (refer to Section 2.1). Simulation results show that the lifetime obtained with the incremental CMLDA is within 3% of the optimal solution while the greedy CMLDA obtains lifetime within 9% of the optimal solution. The greedy and incremental CMLDA protocols perform almost two times better than the hierarchical protocol proposed in [12] in terms of system lifetime. The proposed heuristics need to be generalized for more general situations in which a sensor acts as a data aggregator only for packets from certain sensors while it can be used as a router for other sensors.

3.1.2 A polynomial time approximation scheme

Xue et al. [18] have studied the data aggregation problem in the context of energy efficient routing for maximizing system lifetime. The problem was modeled as a multicommodity flow problem, where the data generated by a sensor node is analogous to a commodity. They have proposed an algorithm which computes (1-$\in$) approximation to the optimal lifetime for any $\in$ >0. The objective of the multicommodity flow problem is to maximize the network lifetime T (time until first node dies), subject to flow conservation and energy constraints. A fully polynomial time approximation scheme (FPTAS) finds an $\in$ approximate solution, which returns at least (1-$\in$) times the optimal value. Its time complexity is polynomial in the size of the network. A *Maxconcurrent flow (Maxlife)* algorithm was proposed which computes a shortest path for one commodity at each iteration of the algorithm. This is followed by updating the weight of each sensor s_k which represents the marginal cost of using an additional unit of the sensor's energy reserve. Since all data sources share a common destination, a shortest path tree rooted at the data sink is eventually formed. For the multi-sink data aggregation problem, a modification of Dijkstra's shortest path tree algorithm has been used. The objective is to compute an aggregation forest which is a unification of M trees routed at data sinks 1, 2, …, M.

The performance of the proposed algorithm (*Maxlife*) has been compared with the minimum energy routing algorithm *(MinEnergy*). The goal of *MinEnergy* is to minimize the energy consumption for each data packet routed through the network. Each source node computes a shortest path to the sink in terms of the total energy cost. The simulation results show that as the network size grows, *MaxLife* doubles the lifetime achieved by *MinEnergy*. The results indicate that *MaxLife* outperforms *MinEnergy* for different network sizes and different number of data sinks. However, the performance gain decreases when the number of data sinks grows.

3.1.3 Energy constrained network flow optimization

Hong et al. [19] have formulated the data gathering problem as a restricted flow optimization problem. The goal of maximal data gathering problem (MDG) is to maximize the number of data gathering rounds subject to the energy constraints of the sensors. The energy constraints on the nodes are transformed into edge capacitates. The quota constraint requires each node to generate a fixed number of packets in a given round. The MDG problem is reduced to a restricted flow problem with edge capacities (RFEC). The sensor network is modeled as a graph $G = (V, E)$ and the RFEC problem determines whether or not there exists a data flow which satisfies the flow constraints, quota constraint and the edge capacity constraints. The capacity of an edge is given by $c(u,v)=N\times n_u$ where n_u is the number of data packets generated per round and N is the total number of rounds.

Given a graph G and a flow f, the residual graph induced by f is a graph $G_f = (V, E_f)$. An edge (u, v) in E_f has residual capacity $c_f(u,v)=c(u,v) - f(u,v)$. The RFEC algorithm finds the shortest augmenting path P from the source to the sink in G_f. The RFEC algorithm obtains an integer valued solution that specifies the number of data packets to be transferred between two neighboring sensors for each round. The shortest path heuristic may not obtain the optimal solution because it searches over possible paths in the original graph instead of the residual graph. Examples have been presented in [19] where for networks with 4 or more sensors, the MLDA algorithm [17] achieves only 50% of the optimal system lifetime.

3.2 Network correlated data gathering

In sensor networks, the data gathered by spatially close sensors are usually correlated. Cristescu et al. [20] have studied the problem of network correlated data gathering. When sensors use source coding strategies, we have a joint optimization problem which involves optimizing rate allocation at the nodes and the transmission structure. Slepian-Wolf coding and joint entropy coding with explicit communication have been investigated in the context of data gathering. In Slepian-Wolf coding, optimal coding allocates higher rates to nodes closer to the sink and smaller rates to the nodes at the extremity of the network. In the explicit communication model, larger rates are allocated to nodes farther from the sink and smaller rates to nodes closer to the sink.

The sensor network is represented as a weighted graph $G= (V, E)$. Every node i transmits data at a rate R_i through the network to the sink. The minimum cost data gathering tree problem attempts to find a spanning tree (ST) of G and rate allocations R_i that minimize the cost C defined as $C = \sum_{e \in ST} h(x_e, w_e)$ where $h(x_e, w_e)$ is an arbitrary cost function of the flow x through an edge e with weight w_e. This is equivalent to minimizing $\sum_{i \in V} R_i d_{ST}(i, s)$ where $d_{ST}(i,s)$ is the total weight of the path from node i to s in the spanning tree. In Slepian –Wolf coding, in the presence of a single sink, the shortest path tree (SPT) is optimal for any rate allocation.

An optimal Slepain-Wolf rate allocation scheme has been proposed in [20]. In this scheme, the closest node to the sink codes data at a rate equal to its unconditioned entropy. All other nodes code at a rate equal to their respective entropies conditioned on all nodes which are closer to the sink than themselves. The main disadvantage of this scheme is that each sensor requires global knowledge of the network in terms of distances between all nodes. To overcome this problem, a fully distributed approximation algorithm has been proposed which provides solutions close to the optimum. In this scheme, data are coded locally at each node, and the conditioning is performed only on the neighbor nodes which are closer to the sink than the respective node.

In the explicit communication model, the data received by a node depends on the transmission structure. Hence the optimization of rates and transmission structure do not separate. Joint optimization in this case is hard, and approximate algorithms have been developed. These include the shortest path tree (SPT), leaves deletion, balanced shortest path tree (BSPT) and simulated annealing. SPT is computed with the distributed Bellman-Ford algorithm. The leaves deletion algorithm is based on the observation that cost improvements are obtained when the leaf nodes change their parent node to an alternative node to reduce the total cost of the tree. The BSPT algorithm is based on a combination of the SPT and multiple traveling salesman paths (TSP). Simulations were performed on networks of different sizes up to 500 nodes. Simulation results indicate that significant improvements in cost (C) are obtained by BSPT and the leaves deletion algorithm compared to SPT. For the explicit communication approach, experiments have shown that full conditioning on all children or a distance dependent correlation coefficient between pairs of nodes do not result in significant cost improvement. In terms of rate allocation, the Slepian-Wolf approach outperforms the explicit communication approach. This reduction in rates is achieved at the cost of increased network knowledge. Table 3 presents the main characteristics and limitations of different network flow based data aggregation algorithms.

Table 3: Summary of network flow based data aggregation algorithms

Algorithm	**Objective and constraints**	**Approach**	**Limitations**
CMLDA [17]	Maximize: network lifetime subject to energy and capacity constraints.	Integer linear programming approach.	High computational complexity for networks of large sizes.
Maxconcurrent flow algorithm [18]	Maximize: network lifetime subject to flow conservation and energy constraints.	Dijkstra's shortest path tree algorithm.	Performance gain decreases with increase in the number of sinks.
RFEC algorithm [19]	Maximize: number of data gathering rounds subject to edge and capacity constraints.	A residual graph based approach to determine the data flow.	Need to consider more realistic models such as dynamic environments where sensors may not generate a fixed number of packets in a round.
Shortest path tree algorithm [20]	Minimize: Total transmission cost of transporting information to the sink subject to capacity constraints on links.	Joint optimization of rate allocation at nodes and transmission structure based on Slepian–Wolf coding strategies.	The Gaussian random field model for characterizing spatial correlation is somewhat arbitrary and its validity should be established.

4. QOS AWARE DATA AGGREGATION PROTOCOLS

Most of the data aggregation protocols discussed so far are designed with energy efficiency as the main goal and hence result in networks with a long lifetime. However, in some applications, the main requirement is a desired quality of service in terms of metrics such as bandwidth, end-to-end delay and information throughput. In this section, we describe the data aggregation protocols whose main focus is on guaranteeing such QOS metrics. Such protocols are also based on the concept of network flow described in Section 3. However, the main difference is the performance measure involved. Research in QOS aware data aggregation can be categorized into two types:

a) Data aggregation protocols that maximize the amount of information collected at the sinks subject to constraints on energy, latency and data flows.
b) Data aggregation protocols which focus on congestion control and end to end reliability.

We now describe such protocols in greater detail.

4.1. Data aggregation protocols for optimal information extraction

Sadagopan et al. [21] have considered the problem of maximizing data extraction in energy limited heterogeneous sensor networks. The problem of maximizing data extraction from energy constrained sensors is formulated as a multi-commodity flow problem subject to constraints on flow conservation. An approximation algorithm based on efficient heuristics such as distance, hop count and residual energy has been proposed which reduces the number of iterations and incorporates selfish, greedy behavior. The heuristics differ in terms of the link metric chosen for distance vector routing. In the exponential metric, the link metric of a sensor at any iteration varies exponentially with the residual energy of the sensor. When all the nodes have similar data and energy levels, all greedy heuristics perform similarly. If there are nodes with very high energy and low data, the exponential metric outperforms the other heuristics. The exponential heuristic results in data flows that are within 15% of the optimum. However, the performance of the exponential heuristic is influenced by the node and data heterogeneity of the sensors. When all the sensors are homogeneous, other greedy heuristics such as distance and hop count perform equally well. In addition, the problem formulated in [21] does not incorporate the data fairness issue. In reality, data from different sensors may have different priority and hence it is important to incorporate priority in the data extraction problem.

Ordonez et al. [22] have considered the problem of optimal information extraction in energy limited sensor networks. The main goal is to find the coordinated operation of all the nodes by setting transmission powers and flow rates in order to maximize the amount of information that reaches the sink. End-to-end fairness is guaranteed by enforcing that each node sends at most a fraction α_i of the total information that is transmitted to the sink. The problem is formulated as a non-linear flow optimization problem subject to energy constraints. Data aggregation can be accommodated by using multiple flows to separately represent the flow of data and the usage of communication channel in order to identify which data can be aggregated together. Models have been proposed for two problems viz., maximizing the total information gathered subject to energy constraints and minimizing the energy usage subject to information constraints. It was shown that the two problems are equivalent to each other in terms of correspondence between optimal solutions and constraints. However, the latter model is computationally more efficient.

Two simple and efficient heuristics have been proposed in [22] for assigning energy to the nodes and distribution of information. In the first heuristic called the direct heuristic, each node sends data directly to the sink. Then they assign as much information as possible to the nodes with the smallest contribution to the objective function which minimizes the total energy consumption. In the second heuristic called the hop heuristic, information is routed from a node to the closest node in the direction of the sink.

Simulations were performed on line and square topologies of sensor networks. For the line topology, the hop heuristic performs poorly in terms of energy consumption. However, for the square topology, hop heuristic performs better than the direct heuristic. The effect of fairness pattern on optimal energy distribution has been investigated. The results show that when there are no fairness constraints, the optimal way for nodes to send data directly to the sink is such that their contributions to the objective function are equal.

4.2 Data aggregation protocol for end-to-end reliability and congestion control

He et al. [23] have proposed an aggregation scheme that adaptively performs application independent data aggregation (AIDA) in a time sensitive manner. Their work isolates aggregation decisions into a module that resides between the network and data link layers The main goal is to maximize the utilization of the communication channel. AIDA performs lossless aggregation in which the upper layer decides whether information compression is appropriate at that time. The AIDA architecture consists of a functional unit that aggregates and de-aggregates network packets. In addition, there is a control unit that adaptively controls timer settings and tunes the degree of aggregation. The transmission and control overhead is reduced by aggregation of multiple network units into a single AIDA aggregate. Figure 8 shows the AIDA architecture.

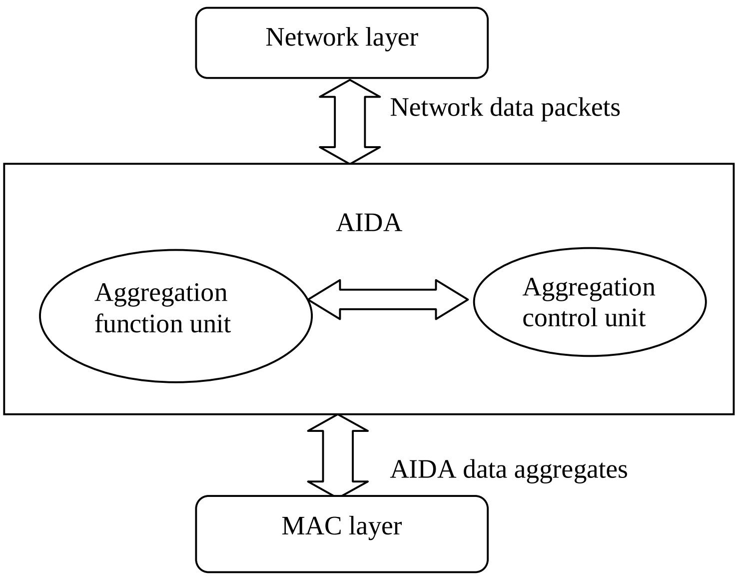

Figure 8: AIDA architecture

Several versions of AIDA have been proposed ranging from aggregation decisions based on static thresholds to a dynamic online feedback control mechanism. In the fixed aggregation scheme, AIDA aggregates a fixed number of network units into an AIDA packet. In the on-demand aggregation scheme, AIDA layer data aggregation takes place only when the MAC layer is available for transmission. The dynamic feedback scheme is a combination of on-demand and fixed aggregation where the degree of aggregation threshold is modified dynamically. This scheme tunes the degree of aggregation threshold and the sending rate to optimize the aggregation performance.

The dynamic feedback scheme was compared with fixed data aggregation, on demand data aggregation and no data aggregation schemes. The simulation results show that dynamic feedback scheme is the best technique with better performance in terms of end-to-end delay, energy consumption and control overhead. The dynamic scheme obtains delay information that directly reflects the current traffic situation resulting in a better control model and better performance. The results show that AIDA reduces the end-to-end delay by 80% and transmission energy by 30-50% compared to the no aggregation scheme under heavy traffic conditions. From the study in [23], it can be concluded that AIDA can complement the benefits of application specific data aggregation schemes.

5. HANDLING TRADE-OFFS IN DATA AGGREGATION

The performance of data aggregation protocols are characterized by performance measures such as energy consumption, latency and data accuracy. There is usually a tradeoff between the different objectives. In this section, we describe approaches for handling the tradeoffs in data aggregation schemes.

5.1 In-network aggregation tradeoffs

Ignacio et al. [24] have investigated in-network aggregation trade-offs for data aggregation in sensor networks. They have focused on sensor network applications such as environmental monitoring that generate data periodically. Timing models play a significant role in the accuracy and freshness of data aggregation. A new *cascading timeouts* data aggregation scheme has been proposed for periodic data aggregation. In this approach, the sink initially broadcasts a request to all nodes. The initial request triggers a tree establishment protocol. This sets up reverse paths from all nodes back to the sink. Each node waits for a certain period of time to receive data from their children after which it times out. The timeout of each node is set based on the position of the node in the data aggregation tree. A node’s timeout occurs before its parent’s timeout. This results in a cascading effect where data originating from leaves reaches nodes in the next tree level in time for aggregation. This is analogous to a “data wave” which reaches the sink. Timeout scheduling is a part of the tree setup protocol which is triggered by the initial request from the sink. The hop-count field in the request is utilized by the nodes to estimate their distance to the sink and schedule their timeouts.

The performance of cascading timeouts has been compared with periodic simple aggregation and periodic per hop aggregation. In periodic simple aggregation, all nodes wait for a pre-defined time interval, aggregate data received in that period and generate a single packet. In per-hop aggregation, once all data packets are received from a node’s children, an aggregated packet is produced and transmitted to the next hop. The proposed energy efficiency metric computes the number of aggregation packets per round N_p given by

$$N_p = \sum_{i \in N} d_i$$

where d_i is the depth of node i and N is the set of nodes in the tree. In addition, data accuracy, data freshness, and overhead were also used as the performance metrics. Data freshness is equal to the difference between the time a data item is generated and the time it is received at the sink.

Simulations were performed comparing no aggregation, periodic, periodic per hop and cascading timeouts. The results show that no aggregation and cascading timeouts achieve the highest percentage of fresh data. Interestingly, the sink placement has significant impact on data freshness. Placing the sink in the center of the field yields fresher data. All data aggregation schemes compared exhibit similar energy efficiency. A new metric called weighted accuracy has been used to compute the data item's age. The weighted accuracy W_a is defined as

$$W_a = \sum_{i \in E} r_i w^i$$

where E is the set of ages of readings, r_i is the number of readings of age i per period and w is the weight. Older readings are assigned an exponentially decaying weight. No aggregation and cascading timeouts exhibit the best weighted accuracy. The results show that cascading timeout maintains the same freshness and accuracy achieved by no aggregation with significant energy savings. However, the approach needs to be generalized to scenarios involving non-periodic data generation and applications where the aggregated data packet length is different from the length of the data packet generated.

5.2 Energy, accuracy, and latency tradeoffs

Boulis et al. [25] have studied the energy-accuracy tradeoffs for periodic data aggregation problems in sensor networks. They have considered a problem in which the sensors provide periodic estimates of the environment. A distributed estimation algorithm has been developed which uses the "max" aggregation function. Some of the unique features of the proposed estimation algorithm include:

a) Scalability with the network architecture
b) Time synchronization between the nodes is not required
c) All nodes have an estimate of the global aggregated value

The key idea of their approach is a threshold based scheme where the sensors compare their fused estimates to a threshold to make a decision regarding transmission. However, in the absence of prior information about the physical environment, setting the threshold is a non-trivial task. The threshold can be used as a tuning parameter to characterize the tradeoff between accuracy and energy consumption. The estimation algorithm was simulated on a 45m×45m network. The results indicate that the energy consumption varies from 5% to 67% of the total initial energy of all sensors in the network depending upon the chosen threshold. The main advantage of the proposed approach is that it does not depend on a hierarchical tree structure for performing data aggregation. Instead, every node has the global information about the aggregated data. The main disadvantage of the approach is that the functionality of the fusion algorithm depends on the aggregation function. Hence the fusion algorithm is not applicable for a wide range of aggregation functions such as "average", "count" or "min".

Yu et al. [26] have also studied the energy-latency tradeoffs for data aggregation in sensor networks. The main goal of their approach is to minimize the overall energy consumption of the network subject to a latency constraint. The non-monotonic energy model used in [26] is based on Quadrature Amplitude Modulation (QAM) scheme. The transmission time τ for transmitting a packet of size l bits is defined as

$\tau = \frac{l}{g \times R}$ where g is the modulation level (number of bits per symbol) and R is the symbol rate. The transmission energy is a nonlinear function of the transmission time. The principle of this energy model is that the transmission energy does not monotonically decrease as the transmission time increases.

The offline packet-scheduling scheme proposed in [26] assumes that the structure of the aggregation tree is known *a priori*. The packet-scheduling scheme is an iterative numerical optimization algorithm that optimizes the overall energy dissipation of the sensors in the aggregation tree. There is no guarantee on the convergence speed of the iterative algorithm. A pseudo-polynomial time approximation algorithm has been developed based on dynamic programming. The main drawback of the algorithm is that each node has to wait for information from all child nodes before performing data aggregation. This might increase the associated latency. Simulations were performed with 200 nodes randomly deployed in a unit square. The algorithms proposed in [26] were compared with a baseline approach in which all the sensors transmit packets with the highest speed and then shutdown their radio. The results show that the proposed algorithms can achieve 20% to 90% energy savings compared to the baseline approach.

5.3 Capacity-energy tradeoffs

Duarte-Melo et al. [27] have studied the transport capacity of data gathering sensor networks with different communication organizations. The hierarchical and flat organizations of sensor networks were compared in terms of capacity and energy consumption. They have discussed the tradeoffs between capacity and energy consumption for data aggregation applications in which every sensor sends an equal amount of original data to the sink. In the flat architecture, nodes communicate with the sink via multi-hop routes by using peer nodes as relays. In the hierarchical structure, nodes are organized into clusters where the cluster heads serve as simple relays for transmitting the data. For a hierarchical network, where cluster heads have the same transmission capacity as the sources, the minimum requirement on the number of clusters has been obtained for achieving the upper bound on the throughput. The main finding of their study is that higher throughput can be achieved by using clustering at the cost of the extra nodes functioning as cluster heads.

Simulation results reveal some interesting relations between the organization of the network, capacity and energy consumption. The flat (multi-hop) network consumes less energy if the area of the network is large while the hierarchical network consumes less energy if the area is small. Hence small networks should be organized into clusters which reduces the energy consumption and increases the capacity. The tradeoff between capacity and energy consumption becomes evident in large networks. If the capacity of the flat network is enough for the desired application, then it is beneficial to use flat (multi-hop) networks to reduce energy consumption. If the application requires a higher capacity, then a hierarchical network should be employed at the cost of increase in energy consumption. Table 4 summarizes the advantages and limitations of different approaches which characterize the trade offs between energy efficiency, latency, capacity and accuracy.

Table 4: Summary of different approaches which characterize the tradeoffs involved in data aggregation

Scenario	Approach / Trade offs	Advantages	Limitations
Periodic data aggregation	A cascading timeouts data aggregation scheme in which a node's timeout is based on its position in the aggregation tree [24]. Energy efficiency, data freshness and accuracy	Minimal control overhead and does not require clock synchronization among sensors.	The approach needs to be generalized for non-periodic data aggregation scenarios.
Periodic data aggregation	A threshold based distributed estimation algorithm [25]. Energy efficiency and accuracy	The aggregation scheme does not depend upon the tree structure.	The fusion algorithm is not applicable for a wide range of aggregation functions.
Real time event monitoring applications	Iterative numerical optimization algorithm that minimizes the energy dissipation of sensors in the aggregation tree [26]. Energy efficiency and latency	About 20-90% energy savings are obtained compared to a classic radio shut down technique.	There is no guarantee on the convergence speed of the iterative algorithm.
Applications which involve many to one communications such as detection in cluster based networks	Study of capacity and energy consumption of flat and hierarchical networks [27]. Energy efficiency and capacity	The study helps the system designer to choose a particular network architecture depending on the capacity and energy constraints.	The idealized disk shaped model used for sensor communication range is unrealistic and does not consider wireless channel fading.

6. SECURITY ISSUES IN DATA AGGREGATION

Security in data transmission and aggregation is an important issue to be considered while designing sensor networks. In many applications, sensors are deployed in open environments and are susceptible to physical attacks which might compromise the sensor's cryptographic keys. Secure aggregation of information is a challenging task if the data aggregators and sensors are malicious. In this subsection, we describe some recent work which solve the secure data aggregation problem and also discuss some of the main issues involved in implementing security in sensor networks.

Girao et al. [28] have analyzed the two main practical issues involved in implementing data encryption at the sensors viz., the size of the encrypted message and the execution time for encryption at the sensors. Privacy homomorphisms (PH) are encryption functions which allow a set of operations to be performed on encrypted data without the knowledge of decryption functions. In [28], PH has been used to analyze the feasibility of security implementation in sensors. PH uses a positive integer $d \geq 2$ for computing the secret key. The size of the encrypted data increases by a factor of d compared to the original data. Hence in the light of minimizing packet overhead, d should be chosen in the range of 2-4 as suggested in [28]. Execution times for encryption operation at the sensors increase with d. For instance when d=2, the execution time for encryption of one byte of data is 3481 clock cycles on a MICA2 mote which increases to 4277 clock cycles when d=4 as reported in [28]. MICA2 motes cannot handle the computation for $d \geq 4$. Hence, the tradeoff between security and computation complexity should be considered when implementing data encryption schemes on sensors.

The other main aspect of security in sensor networks is the establishment of secret keys between the sensor and the base station. Perrig et al. [29] have proposed security protocols for sensor networks which address the key establishment problem. In the approach proposed in [29], all nodes trust the base station at the network creation time and each node is given a master key which is shared with the base station. To achieve authentication between a sensor and base station, a message authentication code (MAC) is used. The keys for encrypting the data and computing the MAC are derived from the master key using a pseudo random function. All keys derived using this procedure are computationally independent. Hence, if an attacker hacks the key, it would not help in determining the master key or any other key. In scenarios where a key is compromised, a new key can be derived without transmitting confidential information.

Przydatek et al. [30] have proposed a framework for secure data aggregation in large sensor networks. They have presented secure protocols for the computation of median, maximum, minimum and average of sensor measurements and estimation of network size. The following issues have been addressed for secure data aggregation.

a) Some sensor nodes may be compromised and transmit wrong data values to the aggregator that corrupts the aggregation result.
b) The aggregator may be compromised and report malicious aggregate values to the home server or sink.
c) Estimation errors introduced by the sampling techniques used by the aggregator to compute the result.

The proposed approach called *aggregate-commit-prove* involves the construction of efficient random sampling mechanisms and interactive proofs enabling the end user to verify the authenticity of the information provided by the aggregator. The three main steps involved in *aggregate-commit-prove* are:

a) Computation of result: The aggregator gathers the data from sensors and locally computes the aggregation result.
b) Committing to the gathered data: In this stage, the aggregator commits to the collected data. This is accomplished by Merkle hash tree construction. In this approach, the aggregator computes a binary hash tree starting from the leaf nodes. Each internal node in the hash tree is computed as the hash value of the concatenation of the two child nodes. The root of the tree is denoted as the commitment of the gathered data. The Merkel hash tree is a commitment to all the leaf nodes. Given an authentic root node, a verifier can authenticate any leaf node by verifying that the leaf value is used to derive the root node.
c) Server-aggregator communication: The aggregator communicates the aggregated result and commitment to the server. The aggregator uses interactive proof protocols to prove the correctness of the reported results to the server. This protocol enables the home server to check the authenticity of the committed data and conclude if the aggregator is malicious.

The proposed framework enables secure data aggregation. However, simulations and experimental study are necessary to demonstrate the effectiveness of the approach. Although some discussion is included about the extension of the approach for hierarchical networks, a more detailed analysis is needed. In particular, functions such as median may not support hierarchical aggregation.

Cam et al. [31] have developed an energy efficient and secure pattern based data aggregation protocol (ESPDA) for sensor networks. They have demonstrated the advantages of ESPDA compared to conventional data aggregation techniques with respect to energy, bandwidth efficiency and security. In ESPDA, the sensor nodes send the pattern codes to the cluster head for data aggregation. The sensor data is transmitted to the sink in an encrypted form without being decrypted anywhere in the transmission path. ESPDA aims at achieving energy efficient data aggregation with secure data communication. Each sensor node executes the pattern generation (PG) algorithm to generate the pattern code. The cluster head uses a pattern comparison algorithm to analyze the patterns.

The characteristics of sensed data are compared with the intervals defined in the lookup table of the PG algorithm and a corresponding critical value is assigned. The critical values of all parameters of the data are combined to generate the pattern code. The main disadvantage of the PG algorithm is that it requires application specific aspects such as environmental parameters, type of sensed data and threshold levels as input. The pattern seed is periodically changed to prevent data manipulation by the intruders. This technique enforces security and data freshness. The sensor nodes that correspond to the unique pattern set, transmit the actual data. Symmetric key cryptographic algorithms are used to guarantee security in sensor networks. ESPDA is more secure since the cluster head does not decrypt the data. The bandwidth occupancy rate (ratio of bandwidth occupancy to total available bandwidth) was used as the performance measure. The simulation results show that at 100% redundancy, the bandwidth occupancy rate of ESPDA is close to zero. The bandwidth occupancy rate increases with decrease in redundancy reaching a 100% when the redundancy is close to zero. ESPDA outperforms conventional data aggregation in terms of bandwidth occupancy. In conventional data aggregation where all sensor nodes transmit the actual data to the cluster head, the bandwidth occupancy is more than 50% of the total bandwidth. However, the performance of the protocol in terms of data security and total energy consumption has not been analyzed. It is intuitive that ESPDA improves the energy efficiency by reducing the number of packets transmitted in a data gathering round. Extensive simulations on different network sizes are necessary to substantiate the results. Nevertheless, to the best of our knowledge, ESPDA is the first attempt to combine energy efficiency with security for data aggregation.

7. CONCLUSIONS

We have presented a comprehensive survey of data aggregation algorithms in wireless sensor networks. All of them focus on optimizing important performance measures such as network lifetime, data latency, data accuracy and energy consumption. Efficient organization, routing and data aggregation tree construction are the three main focus areas of data aggregation algorithms. We have described the main features, the advantages and disadvantages of each data aggregation algorithm. We have also discussed special features of data aggregation such as security and source coding. The trade-offs between energy efficiency, data accuracy and latency have been highlighted. Most of the existing work has mainly focused on the development of an efficient routing mechanism for data aggregation. However, the performance of the data aggregation protocol is strongly coupled with the infrastructure of the network. There has not been significant research on exploring the impact of heterogeneity and mode of communication (single hop versus multi-hop) on the performance of the data aggregation protocols. Although, many of the data aggregation techniques presented look promising, there is significant scope for future research. Combining aspects such as security, data latency and system lifetime in the context of data aggregation is worth exploring. A systematic study of the relation between energy efficiency and system lifetime is an avenue of future research. Analytical results on the bounds for lifetime of sensor networks is another area worth exploring. Existing work has provided bounds on lifetime for networks with specific network topologies and source behaviors. It would be interesting to extend this work to more general network topologies such as cluster based sensor networks.

Security is another important issue in data aggregation applications and has been largely unexplored. Integrating security as an essential component of data aggregation protocols is an interesting problem for future research. Data aggregation in dynamic environments presents several challenges and is worth exploring in the future. Another interesting domain of research is the application of source coding theory for data gathering networks. The sensor data are usually highly correlated and energy efficiency can be achieved by joint source coding and data compression. Although some research has been pursued in this direction [20], there is significant scope for future work.

References

[1] C. Shen, C. Srisathapornphat, and C. Jaikaeo, "Sensor information networking architecture and applications," *IEEE Personnel Communications*, Aug. 2001, pp.52-59.

[2] S. Tilak, N. Abhu-Gazhaleh, W. R. Heinzelman, "A taxanomy of wireless micro-sensor network models," *ACM SIGMOBILE Mobile Comp. Commun. Rev*. , vol. 6, no. 2, Apr. 2002, pp. 28-36.

[3] I. Akyldiz, W.Su, Y. Sankarasubramanian and E. Cayirci, "A survey on sensor networks," IEEE Commun. Mag., vol. 40, no. 8, Aug. 2002, pp. 102-14.

[4] J. Kulik, W.R. Heinzelman and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wireless Networks*, vol. 8, March 2002, pp. 169-185.

[5] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed Diffusion: A Scalable and robust communication paradigm for sensor networks," *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCOM '00),* August 2000.

[6] B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," *Proc. 22nd International Conference on Distributed Computing Systems Workshops*, July 2002, pp. 575-78.

[7] B. Krishnamachari and J. Heidemann, "Application specific modeling of information routing in wireless sensor networks", *Proc. IEEE international performance, computing and communications conference*, vol. 23, pp. 717-722, 2004.

[8] W. R. Heinzelman, "Application-specific protocol architectures for wireless networks", PhD Thesis, Massachusetts Institute of Technology, June 2000.

[9] W.R. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans Wireless Communications*, October 2002, pp. 660-670.

[10] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol .3 , no. 4, Dec 2004, pp. 366-79.

[11] S. Chatterjea and P.Havinga, "A Dynamic data aggregation scheme for wireless sensor networks," *Proc. Program for Research on Integrated Systems and Circuits*, Veldhoven, The Netherlands, Nov. 2003.

[12] S. Lindsey, C. Raghavendra, and K.M. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," *IEEE Trans. Parallel and Distributed Systems*, vol. 13, no. 9, September 2002, pp. 924-935.

[13] K. Du, J. Wu and D. Zhou, " Chain-based protocols for data broadcasting and gathering in sensor networks," *International Parallel and Distributed Processing Symposium*, April 2003.

[14] M. Ding, X. Cheng and G. Xue, "Aggregation tree construction in sensor networks," *2003 IEEE 58th Vehicular Technology Conference*, vol.4, no.4, October 2003, pp 2168-2172.

[15] H. O. Tan and I. Korpeoglu, " Power efficient data gathering and aggregation in wireless sensor networks," *SIGMOD Record*, vol. 32, no. 4, December 2003, pp 66-71.

[16] K. Vaidhyanathan, S. Sur, S. Narravula, P. Sinha, "Data aggregation techniques sensor networks," Technical Report, OSU-CISRC-11/04-TR60, Ohio State University, 2004.

[17] K. Kalpakis, K. Dasgupta and P. Namjoshi, "Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks," *Computer Networks*, vol. 42, no. 6, August 2003, pp.697-716.

[18] Y. Xue, Y. Cui and K. Nahrstedt, "Maximizing lifetime for data aggregation in wireless sensor networks," *ACM/Kluwer Mobile Networks and Applications (MONET) Special Issue on Energy Constraints and Lifetime Performance in Wireless Sensor Networks*, Dec. 2005, pp. 853-64.

[19] B. Hong, V.K. Prasanna, "Optimizing system lifetime for data gathering in networked sensor systems," *Workshop on Algorithms for Wireless and Ad-hoc Networks* (*A-SWAN*), August 2004, Boston.

[20] R. Cristescu, B. Beferull-Lozano and M.Vetterli, "On network correlated data gathering," *IEEE INFOCOM*, vol. 4, no. 4, March 2004, pp 2571-82.

[21] N. Sadagopan, and B. Krishnamachari, "Maximizing data extraction in energy-limited sensor networks," *INFOCOM 2004*, vol.3, March 2004, pp. 1717-1727.

[22] F. Ordonez, and B. Krishnamachari, "Optimal information extraction in energy-limited wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, August 2004, pp. 1121-1129.

[23] T. He, B.M. Blum, J.A. Stankovic, and T. Abdelzaher, "AIDA: Adaptive application-independent data aggregation in wireless sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 3, no .2, May 2004, pp. 426-457.
[24] I. Solis and K. Obraczka, "In-Network Aggregation Trade-offs for Data Collection in Wireless Sensor Networks, " University of California Santa Cruz Technical Report, 2004.
[25] A. Boulis, S. Ganeriwal and M. B. Srivastava, "Aggregation in sensor networks: An energy-accuracy tradeoff," *1st IEEE International Workshop on Sensor Network Protocols and Applications*, USA, May 2003.
[26] Y. Yu, B. Krishnamachari, and V. K. Prasanna, "Energy-latency tradeoffs for data gathering in wireless sensor etworks," *IEEE INFOCOM*, March 2004.
[27] E.J. Duarte – Melo, and M. Liu, "Data-gathering wireless sensor networks: organization and capacity," *Computer networks: The International Journal of Computer and Telecommunications Networking*, vol. 43, no. 4, Nov. 2003.
[28] E.J. Duarte – Melo, and M. Liu, "Data-gathering wireless sensor networks: organization and capacity," *Computer networks: The International Journal of Computer and Telecommunications Networking*, vol. 43, no. 4, Nov. 2003.
[29] J. Girao, D. Weshoff and M. Schneider, "CDA: Concealed Data Aggregation for Reverse Multicast Traffic in Wireless Sensor Networks," *IEEE International Conference on Communications*, vol. 5, 2005, pp. 3044-9.
[30] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen and D. E. Culler, "SPINS: security protocols for sensor networks, " *Wireless Networks*, vol. 8, no. 5, 2002, pp. 521-34.
[31] B. Przydatek, D. Song, and A. Perrig, "Secure information aggregation in sensor networks", *Proc. ACM Conference on Embedded Networked Sensor Systems*, Nov. 2003.
[32] H. Cam, S. Ozdemir, P. Nair, D. Muthuavinashiappam, "ESPDA: Energy-Efficient and secure pattern based data aggregation for wireless sensor networks," *Proceedings of IEEE Sensors*, no.2, 2003, pp 732-736.

BOIGRAPHIES

RAMESH RAJAGOPALAN received the Master's (Honors) degree in Physics and B.E (Honors) degree in Electrical and Electronics from the Birla Institute of Technology and Science, Pilani, India in 2002. He received the Master's degree in Electrical Engineering from Syracuse University, Syracuse, NY in 2004. He is currently a Ph.D. candidate in the Electrical Engineering department at Syracuse University. His research interests are in wireless communications, sensor networks, evolutionary computing and multi-objective optimization.

PRAMOD K. VARSHNEY received the B.S. degree in electrical engineering and computer science and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana-Champaign in 1972, 1974, and 1976 respectively. Since 1976 he has been with Syracuse University, Syracuse, NY where he is currently a Professor of Electrical Engineering and Computer Science. His current research interests are in distributed sensor networks, communications, signal and image processing and remote sensing.