International Journal of Distributed Sensor Networks, 4: 12–26, 2008

ISSN: 1550-1329 print / 1550-1477 online
DOI: 10.1080/15501320701774592

Taylor & Francis
Taylor & Francis Group

Energy Efficient Correlated Data Aggregation for Wireless Sensor Networks

SEUNG-JONG PARK[1], *Member*, *IEEE* and
RAGHUPATHY SIVAKUMAR[2], *Senior Member*, *IEEE*

[1]Department of Computer Science, Louisiana State University, Baton Rouge, Louisiana, USA
[2]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA

Data aggregations from Sensors to a sink in wireless sensor networks (WSNs) are typically characterized by correlation along the spatial, semantic, and temporal dimensions. Exploiting such correlation when performing data aggregation can result in considerable improvements in the bandwidth and energy performance of WSNs. For the sensors-to-sink data delivery, we first explore two theoretical solutions: the shortest path tree (SPT) and the minimum spanning tree (MST) approaches. To approximate the optimal solution (MST) in case of perfect correlation among data, we propose a new aggregation which combines the minimum dominating set (MDS) with the shortest path tree (SPT) in order to aggregate correlated data. To reduce the redundancy among correlated data and simplify the synchronization among transmission, the proposed aggregation takes two stages: local aggregation among sensors around a node in the MDS and global aggregation among sensors in the MDS. Finally, using discrete event simulations, we show that the proposed aggregation outperforms the SPT and closely approximates the centralized optimal solution, the MST, with less amount of overhead and in a decentralized fashion.

Keywords Wireless Sensor Networks; Reliable Transport Protocols

1. Introduction

Wireless Sensor Networks (WSNs) have gained tremendous importance in recent years due to their potential use in various fields[1]. In WSNs, devices used for sensing and communication are usually small, cheap, and low-powered; and hence, have limited resources for computation as well as communication. This has spurred a need for energy-efficient protocols tailored specifically toward sensor network environments.

In general, WSNs have two kinds of traffics: query and data at opposite directions: a downstream from a sink to sensors and an upstream from sensors to a sink, respectively. In this paper, we consider the problem of data aggregation collecting sensor data which are *correlated* to each other. In WSNs, there are two types of data correlation:

(i) *Spatial Correlation*, of data containing information about an event or phenomenon, which overlaps in the spatial domain. An extreme case in the above example is a

Address correspondence to Seung-Jong Park, Department of Computer Science, Louisiana State University, 289 Coates Hall, Baton Rouge, LA, USA.

scenario where two sensors that are right next to each other report the temperature, as the data are then (almost) perfectly correlated. And

(ii) *Semantic Correlation*, of data that are reporting information, which is semantically correlated.

For example, consider the query: *What is the number of cars within the field defined by the coordinates (x1, y1, x2, y2)?* In this example, even if sensors are reporting data about *different* cars, the information is still correlated and can be aggregated, as the required information is merely the count of the number of cars.

Depending on the type or the degree of correlation among data, an energy-efficient data aggregation structure can be different. In this paper, we focus on the construction of an energy-efficient correlation-aware structure to optimize aggregation costs for scenarios when there is spatial and/or semantic correlation.

Such correlation among data can be leveraged by appropriately fusing the data inside the network to the best extent possible, thereby reducing delivering cost for the gathering process. In WSNs, this processing at intermediate nodes is called "in-network processing." Hence, the specific problem we address can be stated as: *Given that there is correlation among sensor data, how can the data gathering structure be built so as to minimize the cost of delivering data ?*

We first limit the scope of the problem to the perfect correlation among data since no optimal solution is known yet in case of partial correlation among data. In this context, we present a simple, scalable, and distributed approach called EnCAS for approximating the *Steiner minimum tree*, and thereby achieve the potential cost benefits introduced earlier.

To aggregate perfectly correlated data in an energy-efficient way, the EnCAS basically uses two structures:

(i) the minimum dominating set (MDS) which is same to the core structure proposed in [2] and

(ii) the shortest path tree which is constructed through a basic flooding.

The purpose of the MDS structure is to aggregate correlated data from neighboring sources; that of SPT is to gather aggregated data among core nodes in the MDS. Through theoretical analysis and simulations, we derive the delivery cost incurred by EnCAS for varying conditions, and compare them with those of other structures, e.g., SPT and MST.

2. Motivation and Idealized Models

2.1. Correlation of Data

In this paper, we consider a multi-hop WSN with one sink at the center and n sensors distributed randomly in a sensor field. The sink sends a query and k of the n sensors respond to that query. We consider the problem of efficiently aggregating the information sent by the k sensors to the sink.

Specifically, the goal is to optimize the message complexity, or the transmission cost, for the sensor data generated by the k sensors to reach the sink. It is assumed that there is correlation, defined as ρ, among the sensor data generated by the k sensors. We consider the case when there is a reasonable degree of correlation between the information collected from different sensors.

For example, let m_1 and m_2 be the amount of data generated by two sensors in response to a query. Without loss of generality, if the size of the data generated by any

sensor in response to a particular query is the same, m, we have $m_1 = m_2 = m$. Now, the message size after aggregation of data from these two sources is:

$$A(m_1, m_2) = m_1 + (1 - \rho) \times m_2 \tag{1}$$

where $A(m_1, m_2)$ is the amount of data after aggregating m_1 and m_2. For this case, when the information from the two sensors are perfectly correlated ($\rho = 1$), we see that the message size after aggregation is the same as the amount of data generated by the sources (m). On the other hand, if there is no correlation ($\rho = 0$) between the two sensor data, the message size after aggregation is of size $2m$.

Considering a more general case where the correlated information from $(i-1)$ sensors merges with the sensor data of the ith sensor, the message size after the merge is

$$A(m_1, \cdots, m_{i-1}, m_i) = A(m_1, \cdots, m_{i-1}) + (1 - \rho) \times m_i \tag{2}$$

where $A(m_1, \cdots, m_i)$ is the amount of data after aggregating a set of nodes from m_1 to m_i. Without loss of generality, if we assume that the message size m_i of all the sensors is m, then the message size after the merge is

$$A(m_1, \cdots, m_{i-1}, m_i) \quad = \quad m + (1 - \rho) \times (i - 1) \times m \tag{3}$$

Thus, we take a conservative standpoint in only considering the correlation between any pair of sensor data.

2.2. Problem Statement

For our discussions, we determine the message complexity of the correlation aware and unaware schemes for a given k number of sources and distribution of k sources among n sensors in a network. We assume that perfect aggregation is possible whenever data from two or more sensors merge. Given these assumptions, the objective is to minimize the *message complexity*. We define the objective function as follows:

$$MC = \sum_{i=1}^{T} m_i \tag{4}$$

where MC represents the objective function, i.e., the message complexity, T represents the total number of transmissions required for the query response from all k sensors, and m_i represents the size of the message for the ith transmission. Note that each transmission may not have the same message size even if the sensor data generated in response to a query has the same size, because when two or more sensor data are aggregated at an aggregation point the amount of information generated may not be the same as the size of the original sensor data. The problem can now be formulated as "*Construct the optimal aggregation structure for receiving the query response from k of the n sensors in a sensor field, where the optimal aggregation structure is the structure that can minimize the objective function by enabling the best aggregation possible*."

It is representative of most of the current sensor network routing protocols [3], [4], [5] in the upstream direction. While there might be some opportunistic aggregation possible because of the overlap of paths from different sources, these structures do not optimize for the total number of transmissions and typically have a large message complexity. If

we assume the perfect correlation ($\rho = 1$), the sizes of the messages forwarded will be the same even if sensor data from several sources fuse. If we refer to this size of the message as m bytes, the message complexity (in bytes) for the default case is:

$$MC = m \times T = m \times 35 = 35m \tag{5}$$

where T refers to the total number of transmissions. In contrast, if we consider the optimal aggregation structure for the same scenario, which minimizes the number of transmissions, the message complexity (in bytes) is:

$$MC = m \times T = m \times 26 = 26m \tag{6}$$

The above example clearly illustrates the potential benefit of having an optimal aggregation structure in terms of reducing the message complexity.

2.3. Optimal Solution for Perfect Correlation

In this section, we determine an optimal structure for the case of perfect correlation among data. Since all data from sources have the same kinds of information, the size of aggregated data should be equal to that of each original data.

2.3.1 Proposition 1. The optimal aggregation structure for the perfect correlation with n nodes and k sources, when the sensor data from k sources are perfectly correlated ($\rho = 1$) is a network Steiner tree.

The proof follows from the definition of a network Steiner tree [6]. Let $G = (V, E, d)$ be the network graph with a vertex set V, an edge set E and distance function d. The distance function in our environment is the edge cost, which is a function of both the message size and the distance as described in (4). When $\rho = 1$, the message size is the same even after fusion of data. So, in this case, the edge cost is a function of the distance only. As defined in [6], the network Steiner tree is the shortest tree spanning a given vertex subset within a network G. From this definition and our problem environment, the Steiner tree is the optimal aggregation structure when $\rho = 1$. ■

It has been observed in [7] that for $\rho < 1$, there is no existing optimal aggregation structure. The reason is that the message complexity now is a function of both the message size and the number of transmissions.

Although the Steiner tree has the optimal cost for our target problem (in case of $\rho = 1$), there are no polynomial time algorithms for finding the Steiner tree in a graph [8]. Even approximation algorithms, such as the ones proposed in [9], [10], are computationally expensive. And also, the MST is a solution of the special case when all nodes are sources. For these two reasons, we consider the *Minimum Spanning Tree* (MST) as the approximated optimal solution. It has been proved in [11], [10], [12] that the cost of the Euclidean Steiner tree and Euclidean MST are of the same order. It has also been shown that the cost of the Steiner tree and the MST is also of the same order [10], [6]. From now on, we will consider the *MST* as the optimal solution of the target problem.

In general, the solution needs centralized computation and the exact location of all the k sources at the sink [11], [10]. Even the approximation algorithms for computing MST require the knowledge of the location of sources. However, it is not practical to assume that a sink knows which sensor is going to send a message *a priori*. For this reason, we conclude the need for a decentralized approach that approximates the optimal aggregation structure without the knowledge of the exact location of sources.

3. Related Works

3.1. Default Aggregation in WSNs

[3] is a data-centric routing framework for gathering information from the sensors to the sink in a WSN. While, it is possible that aggregation can happen opportunistically due to any overlapping paths from the sources to sink, it may not be efficient because of following two reasons:

(i) the structure from the sensors to the sink does not approximate a Steiner tree or a MST; and
(ii) the nodes that are incidentally chosen as aggregation points do not have any notion of the amount of time to wait before it can aggregate the data from all sensors downstream of it efficiently.

While, [13] addresses the second problem to a certain extent, it is still of concern that the proposed structure may not be optimal. [14], [15] are other aggregation structures proposed in the context of sensors-to-sink communication in WSNs. However, they are not proposed in the context of aggregation of information for correlated sensor data and are not efficient for the problem considered in this work.

3.2. Intelligent Aggregation in WSNs

There have been a couple of works that have been proposed to do explicit aggregation in the context of sensor networks. [7], [16] propose a simplified information model and try to solve the problem of aggregating correlated data with two simple heuristics. They consider a correlation model similar to the one considered in this paper and propose two simple heuristics for a given correlation factor ($0 < \rho < 1$). For both solutions, *a priori* knowledge of the location of the sources is assumed. Furthermore, due to the computation cost of tree construction, both approaches are more suitable for continuous information collection rather than an one-shot collection process.

[17] assumes a more generic cost function $f(x)$ given x sources. Assuming that $f(x)$ is a concave non-decreasing function and $f(0) = 0$, it argues that there exists an information collection structure that is good for all canonical concave cost functions f. This algorithm gives a good approximation for a large class of cost functions in the context of WSNs. However, the algorithm is centralized in nature, and assumes the knowledge of the exact location of sources. For this reason, this solution can only be useful for off-line computation.

3.3. Graph Theory Techniques

In [9], an information flow model is considered, where a single server sends a data item to a set of clients requesting this data. The distribution and number of clients is not known *a priori*. The paper uses existing heuristics for the Steiner minimum tree construction. This approach is centralized as the clustering and the determination of hubs are all done in a centralized fashion. [12], [6] present a set of heuristics to approximate the optimality of the Steiner minimum tree. However, these approaches are still centralized and cannot be realized in the context of WSNs.

4. Design Goals And Key Idea

4.1. Problem Scopes and Goals

In the paper, we limit the correlation factor to 1 when all data are perfectly correlated with each other. As discussed before, the optimal solution is the Steiner minimum tree (SMT) known to be a NP-hard problem[18]. Although there have been many previous works in [19] on the approximation of the SMT, those schemes still require computational and communication overheads that WSNs cannot support. In this paper, we design an aggregation structure that approximate the optimal solution in a distributed fashion with less amount of overhead than the distributed approximation of the SMT.

The following are the key goals that the design of our proposed data aggregation strategy is based on:

(i) Perfect Correlation,
(ii) Energy Efficiency,
(iii) Scalability to large number of nodes,
(iv) Decentralization,
(v) Loose Synchronization among nodes,
(vi) Tolerance to Mobility and Node Failures.

4.2. Heuristics in EnCAS

From the definition of the Steiner minimum tree (SMT), we need to find an additional set of nodes that are not sources and inserted into the SMT in order to achieve the shortest connectivity. In graph theory, this set is called "Steiner points." Therefore, one of the above heuristics also tries to find these Steiner points. However, since these Steiner points depend on the locations of the sources, we need to find the optimal set of Steiner points after we know the exact locations of sources.

Instead of solving the SMT problem of which optimal solutions are different to each other based on a given set of sources, we address it with the minimum dominating set (MDS) problem of which the optimal solution is not changed irrespective of the given set of sources. Assuming perfect correlation among all data, it is well known that the early aggregation around sources is to reduce redundant data in tree structures. And we can utilize the above heuristic using the MDS approach. Each node in MDS can work as a Steiner point if it has any neighboring sources around it.

Furthermore, we already have the simple and decentralized solution, the core, for the MDS problem in reliable downstream data delivery discussed in [2]. One of the major applications in reliable downstream data delivery is a query dissemination that is tightly coupled with data gathering problem. Therefore, after a query flooding constructs the core structure, data aggregation can use the core to find the set of Steiner points which aggregate data from neighboring sources. Then the data at some core nodes can be forwarded to its upstream core located at the inside core band since the core structure has the shortest path information toward a sink. Eventually, all data from the core nodes will reach a sink through the path that was constructed during query flooding.

Although there is a gap between the optimal solution of the Steiner minimum tree and the approximated solution using the minimum dominating set, the proposed MDS approach can obtain a promising result compared to other approximations that assume centralized coordination and high computational complexity.

5. EnCAS Design

There are three key elements in the design of the EnCAS approach:

(i) Designation of aggregation nodes: approximate the minimum dominating set through the core construction procedure proposed in [2], and guarantee that every node should know its aggregation node around itself;
(ii) Data gathering at each core node: first allows sources not in the core set to transmit data based on the contention based scheduling;
(iii) Aggregation among core nodes: next allows nodes in the core set to transmit aggregated data from sources to any node in the inside core band so that aggregated data can reach a sink eventually through the shortest path to a sink; and
(iv) Synchronization: bounds nodes in the core set to the time constraint based on band-id (the number of hops from a sink) so that data between the core nodes can be aggregated efficiently. We will describe the details of each of the elements and the rationale behind the realization of these elements in the sections below.

5.1. Construction of the Core Set

The purpose of constructing the core set C is to find the MDS of which size is minimum enough to cover all nodes in set N in a network graph $G = \{N, E\}$. Since the nodes in the MDS act as the Steiner points to aggregate data, it is better to minimize the number $|C| = n_c$ of nodes in the MDS. The core construction can be implemented by any flooding methods which disseminate queries at downstream.

5.2. Aggregation at a Core Node

Once the core set C is determined by the instantaneous construction procedure, each non-core node nc_i not in set C should know its core node at core bands or non-core at core bands. If a node does not have any neighboring node at core bands, it declares itself as a leaf node which exceptionally will send data up to a core node in an inner core band through the shortest path tree.

If a source node not in set C wants to transmit data, it sends data to a core node in set C or a neighboring non-core node in core bands so that data can be aggregated at a core node. Since a core node acts as a Steiner point, the MDS aggregates data with a minimum number of Steiner points so that early aggregation can reduce redundant data as early as possible.

5.3. Aggregation Among Core Nodes

While constructing the core structure, EnCAS also has the shortest path tree rooted at a sink. Basically, every node in a network has its precedent node in the shortest path tree so that all data can be forwarded toward a sink. Therefore, EnCAS does not require any explicit routing scheme that requires the overhead to construct because it uses the shortest path tree, the by-product of a query flooding.

After core nodes aggregate data from neighboring non-core nodes, they send aggregated data to a core node in an inner core band through the shortest path tree. Instead of reaching a sink directly from each core node, data will be forwarded to a core node at an inner core band to prevent aggregated data from selecting individual paths to a sink. Therefore, each core node transmits aggregated data to another core node in an inner core band through at most three hops.

5.4. Synchronization

In general, data aggregation trees, e.g., the Steiner minimum tree, the shortest path tree and the minimum spanning tree, require synchronization for data transmission between two nodes sharing a link in a tree. This will lead to inefficient aggregation and consequently increase the message complexity of aggregation. Therefore, most of the aggregation schemes require synchronization for data transmission of all nodes in an aggregation tree.

EnCAS, however, does not require the above tight synchronization that all nodes in a tree should follow. Instead, EnCAS needs a loose synchronization that only core nodes in a tree should follow. The other non-core are not required to follow the loose synchronization since they are located at leaf nodes of a tree. Practically, we can implement the loose synchronization between two core nodes at different core bands by an easy way. For example, we can synchronize those two core nodes with band-id (hop distance from a sink to a node). Core nodes with lower band-id wait for longer time than other nodes with higher band-id.

5.5. Message Complexity

In this section, we derive the message complexity of the proposed scheme, EnCAS, for the case when there is perfect correlation ($\rho = 1$) among all data generated by sources. Given a network graph $G = \{N, E\}$, we assume that there is a minimum dominating set C of which size is n_c. In the worst case, all k sources are located at non-core nodes so that each source should transmit its data at least one time. And all core nodes should transmit the aggregated data up to another core node at an inner band by at most three transmissions because any pair of two core nodes has a path consisting of at most two non-core nodes. Therefore, total message complexity is

$$MC = k + 3n_c, \tag{7}$$

where k is the number of sources and n_c is the cardinality of MDS C.

There are several upper bounds for the cost of the MDS as a function of the number of nodes n. [20] proves that $n_c \leq n\frac{1+\ln(D_G+1)}{D_G+1}$, where D_G is the minimum degree of a network. However, to the best of our knowledge, there is not known yet the upper bound for the cost of the MDS, which is comparable to a logarithmic or square root function. Therefore, we will compare the message complexity of EnCAS with those of the other two theoretical approaches, SPT and MST, using extensive simulations in Section VII.

6. EnCAS Framework

In this section, we will present the EnCAS approach in detail. After core construction, to aggregate perfectly correlated data, EnCAS uses two stages:

(i) at stage 1, sources only at non-core nodes including leaf nodes transmit data; and
(ii) at stage 2, core nodes transmit data to another core nodes. Those two stage will occur separately.

6.1. Core Construction

Figure 1 shows the instant result for core construction by disseminating a query through a network. Basically, all nodes can access a precedent in the shortest path tree rooted at

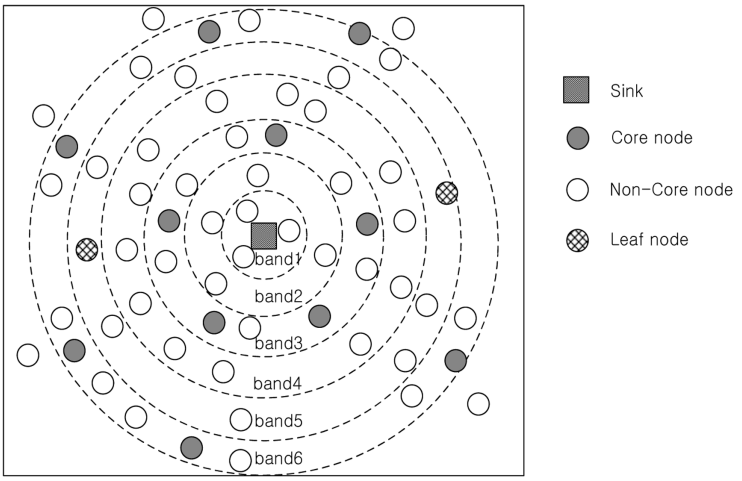


Figure 1. Instantaneous core construction in EnCAS.

a sink. Based on this core structure, a node in a network should be one of core nodes, non-core nodes, or leaf nodes as follows:

- A core node is a node at a core band of which band-id (The band-id means the shortest hop distance from a sink to a node in a network.) is $3i$. Two core nodes in the same core band should have at least a two-hop distance between each other to reduce the total number of core nodes. A core node also keeps the information of a precedent in the shortest path tree root at a sink so that the core at $3i$ band can transmit the data to another core node at inner core $3(i-1)$ band eventually.
- All nodes at non-core bands $3i+1$ or $3i-1$ should be a non-core node. And some nodes at core band $3i$ might become a non-core node based on the core construction procedure. All non-core nodes should access two nodes: its core node at $3i$ band and its precedent in the SPT, of which band-id is less than its band-id. Some non-core nodes at $3i+1$ or $3i-1$ band cannot have a neighboring core node at $3i$ band. In this case, they can still access a core node at $3i$ band through its neighboring non-core node at $3i$ band indirectly.
- For exceptional cases, some non-core nodes of which band-id is $3i+2$ cannot have any neighboring nodes located at core band $3i+3$. These non-core nodes declare themselves as a leaf node. Then they always transmit data to a precedent that is a non-core node at inner band $3i+1$.

6.2. Stage 1: Original Data Transmission

6.2.1 Non-core Nodes. If a non-core node at $3i-1$ or $3i+1$ band is a source node, it will transmit data to its core at core bands after a delay δ_{nc}[1]. If the receiving node at core band $3i$ does not declare itself as a core node, it will forward the data to its core node at the same core band $3i$. We use a contention-free medium access control scheme

[1]The delay δ_{nc} is set based on the maximum number of leaf nodes around a non-core node so that the leaf nodes can transmit data successfully to a non-core node within δ_{nc}.

to coordinate all non-core sources around a core node based on the number of non-core nodes around the core node. In Fig. 2, all non-core nodes, white circles, send data to core nodes, gray circles. Between different groups around each core node, we do not need to consider scheduling because they are separated with each other at least two-hop distance.

6.2.2 Leaf Nodes. If a leaf node at $3i+2$ band is a source node, it will transmit data immediately to its neighboring non-core node at $3i+1$ bands so that the neighboring non-core node can receive the data successfully before it sends its own data. In Fig. 2, leaf nodes at band 5, checked circles, send data to non-core nodes at band 4.

6.2.3 Core Nodes. If a core node at core bands is not a source node, it does not need to transmit data unless it receives any data from its non-core nodes or core nodes at the outer core band. Although the core node has data to send, it will wait for a time δ_c[2] so that it can wait and aggregate its own data with incoming data from other core nodes that are located at outer bands.

6.3. Stage 2: Aggregated Data Transmission

After stage 1, we assume that all data from the non-core nodes are received by core nodes and aggregated with other data. The remaining procedure is to deliver the aggregated data to a sink. To deliver these aggregated data, EnCAS uses the shortest path tree that was constructed during the corresponding query flooding. Figure 3 shows delivery paths between the core nodes at different core bands. Compared to the original shortest path tree, the paths have some differences. Instead of reaching a sink directly using the SPT, it is better to reach another core node at inner band since it can reduce redundancy among other aggregated data. Whenever a non-core node at core bands receives aggregated data from other core nodes at outer bands, it will forward them to its core node at the same core band.

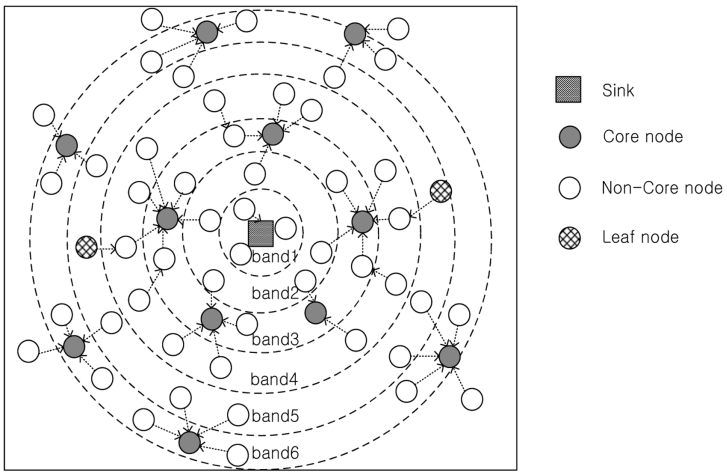


Figure 2. Stage 1: Original data transmission in EnCAS.

[2]The delay δ_c is set inverse proportionally to the band-id.

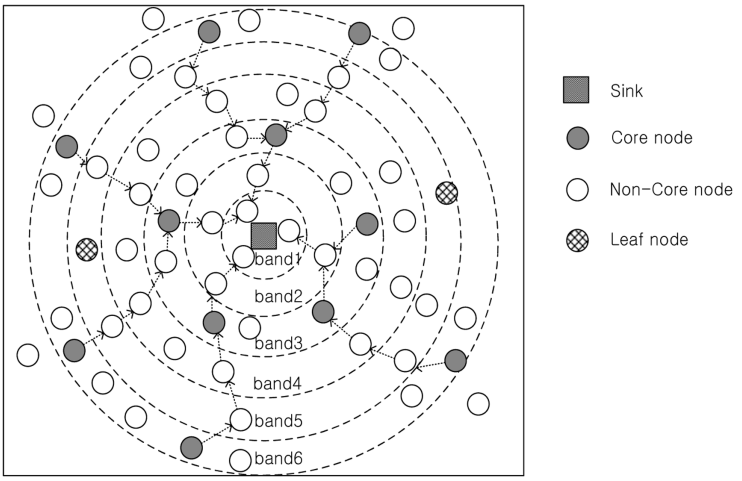


Figure 3. Stage 2: Aggregated data transmission in EnCAS.

7. Performance Evaluation

This section evaluates the performance of the EnCAS approach that uses a decentralized and simple scheme under different network configurations. To do the evaluation systematically, we compare it with two schemes: the shortest path tree (SPT) constructed in a decentralized fashion and the minimum spanning tree (MST) constructed in a centralized fashion with high computational complexity. We vary the node density, source density, source distribution; and compare the performance of the three schemes.

7.1. Simulation Environments

We assume a typical one-shot query-response model in sensor networks. In this model, a sink broadcasts a query to the entire network and sensors that have corresponding information will reply with one message. In terms of message size, we assume that every source sends one message of the same size, but the specific length of the message does not matter.

We use a discrete event simulator for all evaluations. And the simulation topologies are largely similar to that used in general sensor networks: 2000 to 8000 nodes uniformly distributed within a circular field of radius 400m. The number of sources that generate messages for one specific query varies from $\frac{1}{10}$ to $\frac{1}{2}$ of the total number of nodes in the network.

We compare EnCAS with SPT since most of the current routing protocols in the context of WSNs such as Directed Diffusion and GPSR try to approximate the message complexity of SPT. And we are interested in how EnCAS performs better compared with the centralized algorithm. We also compare it with MST, which represents the optimal solution in the target environment. Ideally, we should have compared it with the Steiner minimum tree. But as we mentioned before, the computation overhead is very high, especially when we are considering thousands of nodes, the time it takes to generate even one sample is prohibitive. For this reason, we use MST to approximate the Steiner Tree performance which has the same message complexity order ($O(\sqrt{k})$) and a competitive

cost ratio of less than $\frac{1}{2}$ as that of Steiner minimum tree, but a much less computation cost. We generate SPT with Dijkstra's algorithm, and MST with Prim's algorithm.

To highlight the benefit of EnCAS as a distributed solution, we also compare it with a decentralized version of the shortest path tree (DSPT). In DSPT, we use GPSR routing protocol to approximate SPT in a distributed fashion because as [21] stated, the routes generated by GPSR closely approximate SPT, especially when the node density is high. We evaluate the EnCAS approach using message complexity that is equal to the total cost of data aggregation. For message complexity, we measure the total number of transmissions required for all responses to reach the sink.

To focus on the comparison of aggregation efficiency of different structures, we assume a perfect Media Access Control (MAC) layer that avoids collisions for all approaches. All the simulation results are derived after averaging results over 10 random seeds and are presented within 95% confidence intervals.

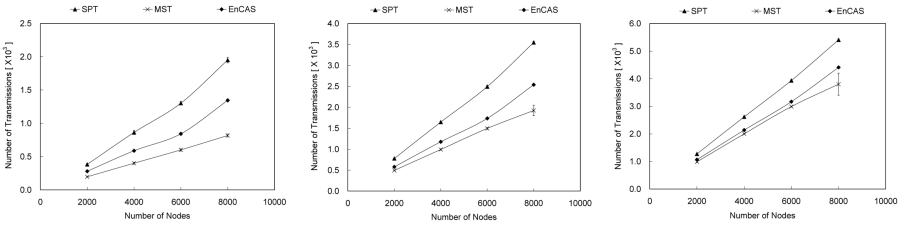
7.2. Different Node Densities

We first compare the performance of the decentralized EnCAS with those of the SPT and the MST assuming that data from all sources are correlated perfectly ($\rho = 1$).

Figure 4 (a), (b), and (c) show the cost of three schemes as a function of the number of nodes for different number of sources k. In these simulations, we choose the total number of nodes n as 2000, 4000, 6000, and 8000; and the number of sources k as $\frac{n}{10}$, $\frac{n}{4}$, and $\frac{n}{2}$, respectively. To compare the efficiency of those three schemes, we measure the message complexity, the number of total transmission during aggregation for different schemes.

It can be seen that EnCAS outperforms the SPT scheme under all situations. From the results, we can observe that EnCAS reduces the message complexity from 16% to 35% compared with the SPT with a small amount of overhead for constructing the core structure without centralized coordination. Although the MST uses the centralized coordination with high computational complexity, it can only reduce the message complexity from 22% to 50% compared with the SPT. Therefore, from the simulation results, we can say that EnCAS is a good decentralized approximation to the MST.

We can also see that the cost of the SPT increases faster than that of the EnCAS approach as the number of nodes increases. This is expected since more number of nodes reduce the efficiency of aggregation in the SPT as the paths chosen by different sources are less likely to overlap. Therefore, EnCAS can be considered as a more scalable decentralized approach as the number of nodes increases. Furthermore, it is observed that



(a) Number of Sources $k = \frac{n}{10}$ (b) Number of Sources $k = \frac{n}{4}$ (c) Number of Sources $k = \frac{n}{2}$

Figure 4. Performance comparison among SPT, MST, and EnCAS for varying number of nodes and fixing the ratio of number of nodes to that of sources to 10, 4, and 2.

the difference between two schemes increases as the ratio of the number of sources to the number of nodes, $\frac{k}{n}$, decreases because more number of sources increase the probability of aggregation for the SPT. As the ratio $\frac{k}{n}$ goes to 1, message complexities of three schemes converge into n transmissions.

7.3. Different Source Densities

In Fig. 5, we compare the cost of three schemes: SPT, MST, and EnCAS, as a function of the number of sources k for different number of nodes n. We also can see that EnCAS still outperforms the SPT and approaches the MST.

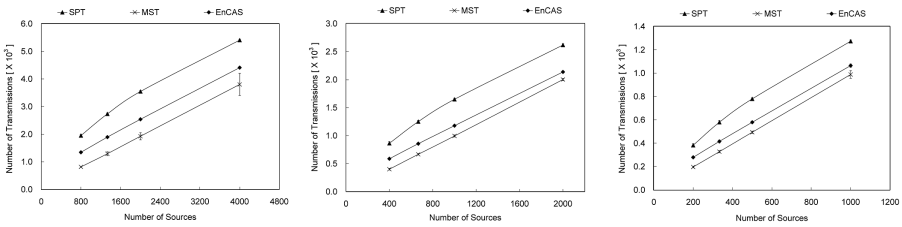
Based on our analysis of the message complexities of SPT and MST, we expect the difference in costs of the two approaches to increase up to a certain k and converge to 0 as $k \to n$. Figure 5 shows that the maximum difference occurs when $k = \frac{n}{4}$. When k is larger than $\frac{n}{4}$ and approaches n, the costs of both schemes will converge to n because each node aggregates all the downstream data and transmits exactly once.

7.4. Number of Core Nodes

To analyze the reasons of EnCAS's outperforming the SPT and approximating to the MST, we observe the number of core nodes of EnCAS. Among two parameters on EnCAS's message complexity: the number of source nodes k and the number of core nodes n_c, the number of core nodes n_c is the only parameter that changes the message complexity of EnCAS since k is a given value. Figure 6 shows the number of core nodes selected in simulations varying the number of nodes n from 2000 to 8000. It shows that the percentage ratio of $\frac{n_c}{k}$ is below 2% over all scenarios from 2,000 to 8,000 nodes and less than the number of sources k. In the simulations, we assume that the number of sources is larger than 10% of the nodes. Therefore, in message complexity of EnCAS, a dominating factor is the number of sources k of which message complexity order is lower than the order of the SPT.

8. Conclusion

In this paper, we proposed the decentralized and scalable data aggregation algorithm, EnCAS, which approximates the optimal solution, the Steiner minimum tree (SMT), of perfectly correlated data aggregation. Since the EnCAS adopts the concept of the minimum dominating set (MDS) as like GARUDA [2], its performance is approaching that of the minimum spanning tree. With numerical results, we compare the proposed



(a) Number of Nodes $n = 8000$ (b) Number of Nodes $n = 4000$ (c) Number of Nodes $n = 2000$

Figure 5. Performance comparison among SPT, MST, and EnCAS for varying number of sources and fixing number of nodes to 8000, 4000, and 2000.

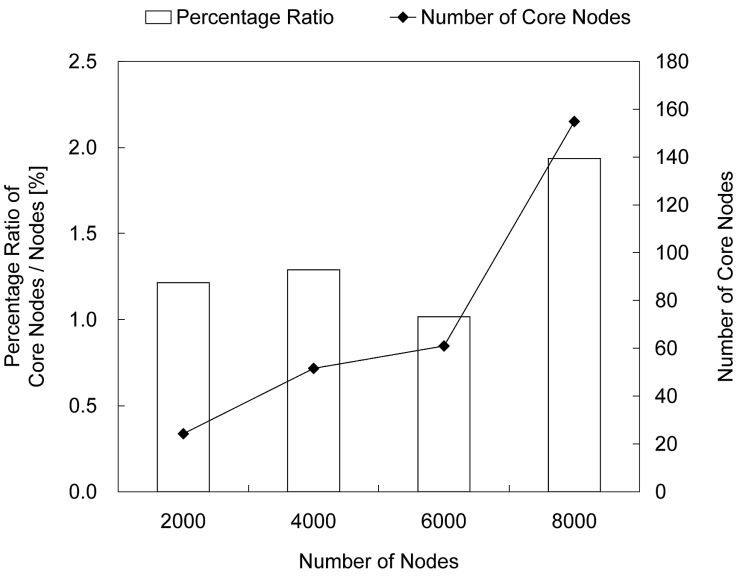


Figure 6. Percentage ratio of the number of core nodes to the number of nodes in EnCAS simulations.

EnCAS scheme with the decentralized simulation SPT and the centralized MST schemes. Simulation results indicate that EnCAS outperforms the SPT substantially in terms of delivery cost for all environments. The cardinal reason for EnCAS's superior performance over that of the SPT is because EnCAS approximates the MST using heuristics to choose the Steiner points from the minimum dominating set. Moreover, the proposed scheme, EnCAS, can be used with the GARUDA [2] since they share the same core structure.

About the Authors

Seung-Jong Park is an assistant professor in the Computer Science Department and Center for Computation Technology at Louisiana State University. He received his Ph.D. from The School of Electrical and Computer Engineering at Georgia Institute of Technology, 2004. Prior to that, he had also received a B.S. degree in Computer Science at Korea University, Seoul, Korea and a M.S. degree in Computer Science from KAIST (Korea Advanced Institute of Science and Technology), Teajon, Korea in 1993 and 1995, respectively. From 1995 to 2000, he had worked for Shinsegi Telecomm, which is the first CDMA cellular service provider in the world and has now merged with SK Telecom.

Raghupathy Sivakumar is an Associate Professor in the School of Electrical and Computer Engineering at Georgia Tech. He leads the Georgia Tech Networking and Mobile Computing (GNAN) Research Group, where he and his students do research in the areas of wireless networking, mobile computing, and computer networks. Professor Sivakumar received his Ph.D. and M.S. degrees in Computer Science from University of Illinois at Urbana-Champaign in 2000 and 1998 respectively, and his B.E. degree in Computer Science from Anna University (Chennai) in 1996.

References

1. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks Journal*, vol. 38, no. 4, pp. 393–422, March 2002.
2. Seung-Jong Park, R. Vedantham, R. Sivakumar, and I.F. Akyildiz, "A scalable approach for reliable downstream data delivery in wireless sensor networks," in *ACM MOBIHOC*, Tokyo, Japan, May 2004, pp. 78–89.
3. Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *MOBICOM*, Boston, USA, Aug. 2000, pp. 56–67.
4. B. Karp and H.T. Kung, "Greedy perimeter stateless routing for wireless networks," in *MOBICOM*, Boston, USA, Aug. 2000, pp. 243–254.
5. O. B. Akan and I. F. Akyildiz, "Event-to-sink reliable transport in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 1003–1017, October 2005.
6. Alexander Zelikovsky, "Better approximation bounds for the network and Euclidean Steiner Tree problems," in *Tech. Rep. CS-96-06, University of Virginia, Charlottesville*, VA, USA, 1996.
7. R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On network correlated data gathering," in *INFOCOM*, Hong Kong, Mar. 2004.
8. R. M. Karp, *Reducability Among Combinatorial Problems, Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Plenum Press, New York, 1972.
9. David R. Karger and Maria Minkoff, "Building Steiner Trees with incomplete global knowledge," in *Proceedings of the 41th Annual IEEE Symposium on Foundations of Computer Science*, 2000.
10. H. Takahashi and A. Matsuyama, "An approximate solution for the steiner problem in graphs," *Math. Japonica 24*, vol. 24, no. 1, pp. 573–577, Jan. 1980.
11. E. N. Gilbert and H. O. Pollak, "Steiner Minimal Trees," in *SIAM J. Applied Math*, 1968, vol. 16, pp. 1–20.
12. Ning-Yang B. Wang and Reui-Chuan Chang, "An upper bound for the average length of the Euclidean Minimum Spanning Tree," in *J. Computer Math*, 1989, vol. 30, pp. 1–12.
13. W. Yuan, S. V. Krishnamurthy, and S. K. Tripathi, "Synchronization of Multiple Levels of Data Fusion in Wireless Sensor Networks," 2003.
14. Haiyun Luo, Fan Ye, Jerry Cheng, Songwu Lu, and Lixia Zhang, "TTDD - A Two-Tier Data Dissemination for Large-scale Sensor Networks," in *MOBICOM*, 2002.
15. S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, "GHT - A geographic hash-table for datacentric storage," in *In First ACM International Workshop on Wireless Sensor Networks and their Applications*, 2002.
16. R. Cristescu and M. Vetterli, "Power efficient gathering of correlated data: Optimization,NP-completeness and heuristics," in *The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, June 2003.
17. A. Goel and D. Estrin, "Simultaneous optimization for concave cost: Single sink aggregation or single source buy-at-bulk," in *ACM-SIAM Symposium on Discrete Algorithm*, 2003.
18. R. M. Karp, "Reducibility among combinatorial problems," *Complexity of Computer Computations*, no. 1, pp. 85–103, May 1972.
19. F. K Hwang, D.S. Richards, and P. Winter, *The Steiner Tree Problem*, North-Holland, 1992.
20. N. Alon and J.H. Spencer, *The Probabilistic Method*, J. Wiley and Sons, 1992.
21. B. Karp and H. T. Kung, "Gpsr: Greedy perimeter state routing for wireless networks," in *MOBICOM*, Boston, USA, Aug. 2000, pp. 243–254.