# A Scalable Correlation Aware Aggregation Strategy for Wireless Sensor Networks

Yujie Zhu, Ramanuja Vedantham, Seung-Jong Park and Raghupathy Sivakumar
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, USA
{zhuyujie, ramv, sjpark, siva}@ece.gatech.edu

## Abstract

*Sensors-to-sink data in wireless sensor networks (WSNs) are typically correlated with each other. Exploiting such correlation when performing data aggregation can result in considerable improvements in the bandwidth and energy performance of WSNs. In order to exploit such correlation, we present a scalable and distributed correlation-aware aggregation structure that addresses the practical challenges in the context of aggregation in WSNs. Through simulations and analysis, we evaluate the performance of the proposed approach with centralized and distributed correlation aware and unaware structures.*

## 1. Introduction

Wireless sensor networks (WSNs) have gained tremendous importance in recent years. One of the key tasks performed by any WSN is the collection of sensor data from the sensors in the field to the sink for processing. This task is also referred to as *data gathering*. An important challenge associated with data gathering is to reduce the message complexity to minimize the bandwidth usage of the network and the energy consumption of the sensor nodes. In this paper, we consider the problem of efficient data gathering in environments where the data from different sensors are *correlated*.

Many research work have proposed solutions to construct correlation-aware structures [1, 2, 3]. However, these approaches are either centralized and require complete knowledge regarding the number and location of sources, or do not address several important practical challenges for WSNs, such as ease of construction, maintenance and synchronization requirements. Therefore, those approaches are not suitable for a real-life sensor network environment.

In this context, we present a simple, scalable, and distributed approach called $SCT$ (Semantic/Spatial Correlation-aware Tree) that does not require any centralized coordination while still achieving potential cost benefits due to efficient aggregation. The SCT structure is instantaneously constructed during the course of a single query delivery and is a fixed structure that is efficient for wide range of source densities and source distributions. The SCT approach, with its highly manageable structure, ensures low maintenance overhead of the aggregation structure, while also addressing the other challenges described in Section 3.

The rest of the paper is organized as follows: Section 2 defines the problem and discusses existing data gathering structures and analyzes their characteristics. Section 3 identifies the different challenges in designing a practical, efficient correlation aware structure. Section 4 presents the key design principles in the SCT approach and describes how it addresses the corresponding challenges. Section 5 presents the SCT approach in detail. Section 6 evaluates the performance of SCT with ideal structures and practical implementation of shortest path tree (SPT) while Section 7 concludes the paper.

## 2. Problem Definition and Related Work

We consider a multi-hop WSN with one sink and $n$ sensors distributed uniformly in a sensor field. The sink sends a query and $k$ of the $n$ sensors respond to the query. We assume that all the sensors have the same fixed transmission range equal to $cr_0$, where $c$ is a small constant ($c > 1$) and $r_0$ is the minimum connectivity transmission range [4]. As a measure of the energy efficiency of a data gathering structure, we define its message complexity as $H$, where $H$ represents the total number of transmissions required for responses from all $k$ sources to reach the sink. Our primary goal is to minimize $H$ when there is correlation present between the data from different sources, defined by correlation degree $\rho$. Correlation degree $\rho = 1$ means that two messages are perfectly correlated, $0 < \rho < 1$ indicates that two messages are partially correlated, while $\rho = 0$ implies that two messages are independent with each other.

### 2.1. Correlation Unaware Approaches

Two popular routing approaches proposed in the context of sensor networks can be classified as correlation unaware approaches. Directed diffusion [5] uses the query paths to construct the sensors-to-sink routes, and it is shown that data gathering tree constructed by directed diffusion resembles shortest path tree. GPSR [6] uses the location information of sensors and the sink to forward messages to the sink, and the data gathering tree generated using GPSR is also approximations of shortest path tree. Since the primary goal of this structure is to minimize delay, the shortest path tree is not a correlation-aware data gathering structure. Even though opportunistic aggregation may still occur when different paths overlap with each other, this structure does *not* maximize the aggregations possible in the network.

### 2.2. Correlation Aware Approaches

There are several related works that have been proposed in the context of explicit aggregation [1, 2, 3]. When the full knowledge about source locations is known, the Steiner Tree over all sources, sink and some of the non-source nodes gives the optimal message complexity when the degree of correlation is very high. However, the computation of Steiner Tree is a NP-Hard problem [7]. In [1], the authors propose simple heuristics that approximate the Steiner tree to do efficient aggregation when messages are perfectly correlated. For any given $\rho$, $0 < \rho \leq 1$, the authors describe two heuristics as simple alternatives that approximate the performance of Steiner tree. In [8], the authors first identify that the message complexity can be modelled as a concave-cost function for any correlation factor, and propose an algorithm that constructs a good approximation structure that is close to optimal for all concave cost functions. However, these approaches require complete information regarding the number of sources and their locations to be available at the sink and cannot work for the cases when there is incomplete information. Also, they are centralized approaches and do not scale well with increasing node densities typical to WSN environments and do not address the practical challenges identified in Section 3. [2] and [3] address the more general problem of building aggregation structure with optimal expected cost when there is incomplete knowledge of sources. They identify that the problem can be considered as a Stochastic version of the deterministic Steiner tree problem. Since this problem is NP-complete, the authors focus on developing constant-factor approximation algorithms. In [2], the authors propose a *hub and spoke* model to construct a good approximation aggregation structure to the Stochastic Steiner tree. In [3], a 2-stage approximation algorithm is proposed. However, those approaches are neither distributed nor scalable, thereby are not tailored for sensor network environments. Both approaches need centralized computation of high complexity and assume each node responding to a query with a fixed probability. The scenario where there is a varying source density are not handled by those approaches.

## 3. Challenges

The main goal of this work is to design an efficient aggregation structure that minimizes the message complexity. In order to realize this goal, we identify the following important challenges and elucidate the desirable properties of a solution that addresses the challenge.

### 3.1. Construction

The foremost consideration in building an aggregation structure is the manner in which the aggregation structure is constructed. Even the approximation algorithms for constructing a Steiner tree impose some requirements such as the information regarding the number of sources and the locations of them to be available at the sink *a priori*. This information could potentially be obtained if the sink adopts a two-phase querying procedure, where the query is sent in the first phase and the responses collected reveal the locations of the sources interested in responding to that query. In the second phase, the sink initiates the construction of the approximation of a Steiner tree to optimize the message complexity of the data sent. However, this solution is not desirable across all types of queries and responses as shown below:

- *One-shot queries and responses*: In this category, the sensors send a one-time query and the corresponding responses from the sources are also one-time responses. In this case, the two-phase procedure that we had discussed above will be clearly infeasible both in terms of delay and message complexity, as the message complexity and delay of the fist phase is comparable to the second phase. In this case, if the probability distribution of sources is given, the stochastic Steiner tree is the optimal aggregation structure.
- *Single queries and multiple responses from the same set of sources*: Here, the sink sends one-time queries but the responses from each sensor may be comprised of multiple packets. However, the set of sensors responding to the query remains the same over all the packets. While in this case, it may seem that the two-phase approach may provide low message complexitys, the delay incurred in determining the number of

sources could potentially limit its application. However, the network Steiner tree is still the optimal aggregation solution in terms of low message complexity since the number and locations of sources is known after the first packets from all sensors reach the sink.

- *Single query and multiple responses from a varying set of sources*: Here, the responses to the one-time queries may comprise multiple messages but the responding source sets may vary with time. For this case, it is desirable to have a solution that is independent of the locations of sources or the number of sources. In this case, the optimal solution is neither a network Steiner tree nor a Stochastic Steiner tree. We define this problem as a generalized Stochastic Steiner tree problem.

In summary, a desirable practical solution should consider the tradeoffs between the overhead involved in the construction process itself on the one hand, and the message complexity of the aggregation structure on the other hand, and ensure that it is reasonably efficient across all query and response paradigms.

### 3.2. Maintenance

Once the structure is constructed, it may be required that the structure be modified or reconstructed after a certain period of time to accommodate load balancing, node failures or any other reasons. Ideally, the maintenance overhead of the structure should be negligible in terms of both message complexity and delay.

We will now consider two important reasons for reconstructing the aggregation structure and discuss the preferred characteristics for accommodating those considerations:

- *Load Balancing*: This is to ensure that the energy consumed by all the nodes is fairly even over a certain period of time. In any aggregation structure, aggregation nodes take the responsibility of receiving, compacting and transmitting aggregated messages, and hence on average consume more energy than non-aggregation nodes. It is therefore likely that these nodes fail much earlier than the other nodes, impacting the connectivity of the network. To address this issue, we would like to spread the role of aggregation node among all nodes so that the network does not become disconnected prematurely.

- *Node Failures*: Once the structure is set up it should also be resilient to node failures, which are common in sensor networks [5]. Otherwise, it is likely that some messages will never reach the sink even though there may be alternate paths available. Therefore, when node failure occurs, the structure should have the ability to adapt and form a different, near-efficient structure.

### 3.3. Synchronization Requirements

One of the main considerations for any aggregation scheme is the time each node has to wait before it aggregates the messages received from all sources downstream of it. We refer to these timing requirements necessary for aggregation as synchronization requirements. In the absence of such synchronization requirements, it is conceivable that messages from some downstream sources may arrive after aggregation at a particular aggregation node and hence need to be transmitted separately. This will increase the message complexity despite the existence of an efficient aggregation structure.

Ideally, a scheme should enable an aggregation node to wait until the arrival of messages from all sources downstream before aggregation is performed. One way to do this is by having a fine-grained timer at every node and waiting for the expiry of the timer based on a waiting function, similar to the one described in [9], before performing aggregation. However, it is difficult to set aggregation timers accurately, especially when the network topology changes. This timer inaccuracy may in turn causes imperfections in the aggregation. Also, the computational overhead in maintaining fine-grained timers makes it less desirable. On the other hand, if a coarse-grained timer is used, the possible aggregation cannot be maximized.

In order to address this problem, an ideal aggregation structure should facilitate event-driven aggregation and should rely on timing requirements only sparingly. In this case, the timers can be made coarse as they will not be used often.

## 4. The SCT Design Basis

The design of SCT is predicated on two key elements [1]:

- An aggregation backbone facilitating the generation of efficient aggregation trees

- A fixed structure independent of source distribution and density.

These two design elements address the challenge of efficient construction and maintenance, and incorporate the characteristics and requirements of sensor networks. In this section, we establish and justify these two design elements. The details of SCT approach will be presented in section 5.

1 Without loss of generality, we consider a circular sensor network with sink located at the center of the field from now on. However, this requirement is only used for easy of analysis and presentation, and is not required for the proper functioning of SCT

### 4.1. Motivation for a Ring-and-Sector Division

As mentioned in Section 3, the optimal aggregation structure from sources with varying source distributions and densities is a generalized version of stochastic Steiner tree. In [2, 3], the authors have presented centralized constant-factor approximations to the stochastic Steiner tree problem. In this subsection, we use similar arguments to motivate the structure we propose to approximate the generalized stochastic Steiner tree.

Consider the network model in which each sensor, $i$, has a probability, $p_i$, to become a source for a given query. If an edge, $e$, between two sensor node is used by a set of source nodes $D$ to transmit data to sink, then the cost of this edge, $c_e$, can be defined as:

$$c_e = Pr[e\ is\ active] = 1 - \prod_{i \in D}(1 - p_i) \quad (1)$$

This cost function captures the tradeoffs in the characteristics of data transmission and correlation in sensor networks as follows: when several sources use a particular edge, the communication cost increases; at the same time, the cost per message decreases when a large number of sources use the same edge due to the correlation between different messages. Since $c_e$ is a concave function, if the number of sources using one edge is beyond a certain value, adding more sources causes only a minimal increase in the total cost. Therefore, it pays to place a certain number of aggregation nodes in the network with the property that all the edges between those aggregation nodes are highly utilized [2].

Given the aggregation backbone structure, the next question is what is the optimal number of the aggregation nodes. An immediate observation is that with increasing number of sources, the number of backbone aggregation nodes required increases, since, when the number of sources is small, probability of aggregating two or more messages from different sources at an aggregation node is relatively small. Moreover, having more aggregation nodes translates to a larger number of nodes, which will increase the message complexity because of the additional transmissions required by the aggregation nodes.

However, as source number increases, the probability that two or more messages getting aggregated at each aggregation node increases. Therefore, addition of aggregation nodes helps in aggregating the messages from different sources as early as possible. In this case, the additional cost incurred by introducing extra aggregation nodes can be offset by the reduction in the number of edges with high-transmission cost because of early aggregations. Hence, with increasing source density, it is desirable to have increasing number of aggregation nodes.

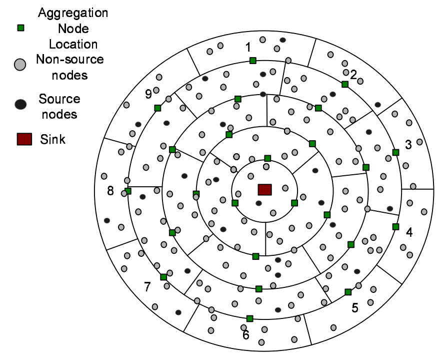



**Figure 1. SCT structure**

Based on these observations, we propose a ring-sector division of the network with the following features:

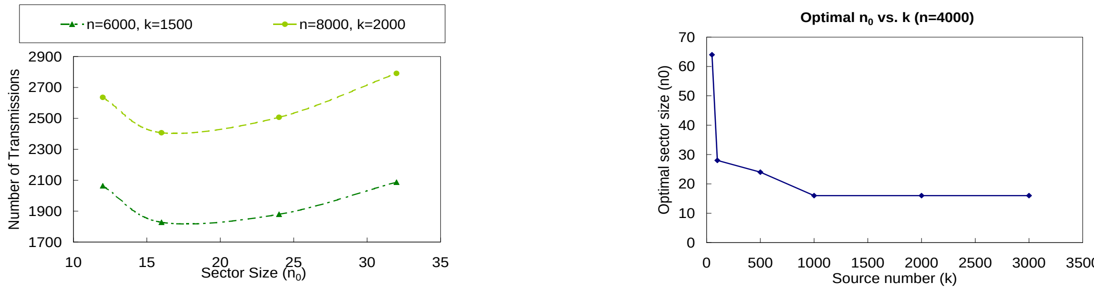
- A subset of nodes is chosen as aggregation nodes and a spanning tree is built on top of these nodes to form a “backbone” for aggregation.
- Each node in the backbone is responsible for aggregating messages from sources within a certain sub-area

As illustrated in Fig. 1, the network is divided into m concentric rings with the same width ($R/m$). Each ring is in in turn divided into sectors of the same size such that approximately $n_0$ nodes are distributed within each sector. For each sector, an aggregation node is chosen as a member of the aggregation backbone, and an aggregation node in $ith$ ring is connected to it’s upstream aggregation node in $(i-1)th$ ring via shortest path. The collection of all aggregation nodes and shortest paths forms the backbone aggregation tree. Each aggregation node is responsible for collecting messages from all sources in the sector it belongs to.

As we will see in Section 5, this structure can facilitate the realization of desirable aggregation backbone in a distributed fashion. But the problem is only partially addressed because our goal is an optimal aggregation structure for variable source densities. This implies that the ring-sector structure proposed should adapt to source densities in order to approximate optimal solution. However, in the next subsection, we will show that a fixed structure satisfying certain properties is reasonably efficient for a wide range of source densities, and hence motivate a relatively stable aggregation structure with low maintenance overhead.

### 4.2. Motivation for a Source-Independent Aggregation Structure

In previous sub-section we point out that for the ring-sector structure proposed, the optimal number of aggregation nodes increases with the source density. In this subsection, we will show that when the source density is




(a) Message Complexity Variation

(b)Optimal Section Size Variation

**Figure 2. Determination of $n_0^*$**

beyond a certain value, the optimal structure no longer changes because of a "saturation" phenomenon.

In general, the optimal sector size reduces with increasing source numbers. However, this reduction is not always desirable. Consider a certain threshold source number $k_0$ for which the optimal sector size is small enough such that aggregation node just falls into transmission range of every other node in the sector. We call the size of the sectors at this point the *saturation size*. In this case, every source message can reach aggregation node with 1-hop. Reducing sector beyond the saturation size will not help in increasing the aggregation efficiency. Furthermore, the introduction of additional aggregation nodes increases the message complexity. Thus, when the number of sources, $k$, is increased beyond $k_0$, decreasing the sector size results in increased message complexity. Therefore, when $k > k_0$, it is desirable to maintain the same aggregation structure.

The effect of saturation on message complexity is also substantiated by simulation results. Figure 2 (a) shows for a certain $n$ and $k$, how the message complexity of aggregation tree varies with sector size $n_0$. Figure 2 (b) shows how the optimal $n_0$ varies with $k/n$. From this figure we observe that for a wide range of source densities, the optimal structure is the same. This suggests that a fixed aggregation structure is efficient when the number of sources is reasonably high.

Based on this observation, we choose $m$ and $n_0$ such that each sector in the network reaches the saturation size. The optimal values of $m$ and $n_0$ ($m^*$, $n_0^*$) can be estimated at the sink as explained next. The choice of $m$ actually determines the depth of the aggregation tree. To achieve saturation size, level-i aggregation nodes should be within 1-hop distance to level-i+1 aggregation nodes. Therefore, the optimal value of $m$ - $m^*$ that minimize the total aggregation tree cost can be determined by the following formula:

$$m^* = \frac{R}{r}\beta \tag{2}$$

where $R$ is the radius of the network, $r$ is the transmission range of the nodes, and $\beta$ is a constant determined empirically as $1.32$.

Determination of $n_0$ is based on the requirement that every node within this sector is less than 1-hop away from the aggregation node. The furthermost possible node within this sector is located at the corner of this sector and we indicate the distance of this node to aggregation point as $b$. The distance $b$ in figure 3 is calculated in the right-angled triangle as follows:

$$\begin{aligned} z &= \frac{(i-1)R\sin\alpha}{m} && (3)\\ y &= \frac{iR}{m} - \frac{(i-1)R\cos\alpha}{m} && (4)\\ b &= \sqrt{z^2+y^2} && (5)\\ &= \frac{R}{m}\sqrt{(4i(i-1)\sin^2\frac{\alpha}{2})+1} && (6) \end{aligned}$$

The angle $\alpha$ for the $i$th ring is given by:

$$\alpha = \frac{m^2 n_0 \pi}{2(2i-1)n} \tag{7}$$

When $\alpha \to 0$[2], $sin\frac{\alpha}{2} \to \frac{\alpha}{2}$, and equation 6 reduces to

$$b = \frac{R}{m}\sqrt{\frac{1}{2}(\frac{m^2 n_0 \pi}{2n})^2+1} \tag{8}$$

To make sure $b$ is less than transmission range, we let $b = \sigma r$, where $\sigma$ is a constant. From equations 8 and 2, and we can derive the optimal $n_0^*$ for large $k$ as:

$$n_0^* = 0.428(\frac{r}{R})^2\sqrt{\sigma\beta^2-1} \tag{9}$$

Through empirical studies, we determine $\sigma = 0.836$. To verify the accuracy of the choice of constants, we compare optimal $m$ and $n_0$ derived from equation 2 and 9 and those obtained from simulation. Figure 4 shows that the theoretical values closely match experimental optimums. These results further substantiate that 1-hop transmission model is the optimal aggregation model.

2 This is true for most rings that are not close to sink.

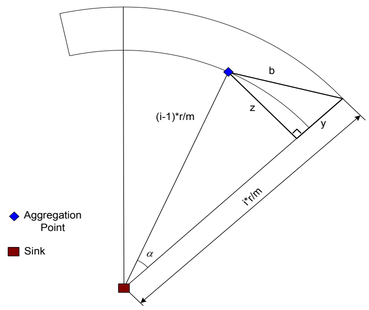



**Figure 3.** **Maximum Distance Travelled**

| n | Comp. $m^*$ | Exp. $m^*$ | Comp. $n_0^*$ | Exp. $n_0^*$ |
|---|---|---|---|---|
| 2000 | 8.54 | 9 | 13.9 | 16 |
| 4000 | 11.6 | 12 | 14.9 | 16 |
| 6000 | 14.1 | 15 | 15.5 | 16 |
| 8000 | 16.1 | 17 | 16.0 | 16 |

**Figure 4.** **Computed and Experimental** $m^*$ **and** $n_0^*$

# 5. The SCT Approach

In this section, we present an overview of the SCT approach and explain the event-driven aggregation in detail.

## 5.1. SCT Structure

During the setup phase, the sink sends the location of itself, $(X_s, Y_s)$, the total number of nodes, $n$, the radius of the network, $R$ , the number of rings $m$, and average number of nodes within each sector $n_0$ to all the nodes in the network.

Each node in the network is assumed to know it's own geographical location. When a node receives these parameters, it first computes the distance between itself and the sink. This determines the ring, $i$, and the sector to which it belongs. Using the same set of parameters, every node can also determine the boundaries of the sector in which it is contained. This is used to compute the ideal location of the aggregation node in this sector, which is defined as the geometric center of the lower arc bounding that sector. If we are to adopt a polar coordinates and if $\alpha$ and $\beta$ are the bounding angles of a sector corresponding to the $i$th ring, the location of the aggregation node is given by $((i-1)\frac{R}{m}, \frac{\alpha+\beta}{2})$.

Once the location of the aggregation node is determined, a source transmits its message to the geometric location of the aggregation node or sink (if the source is within the first ring) using a one-hop location based routing protocol. Note that usually it is not possible that there is a node right at the ideal location of the aggregation point. However, the aggregation node can be chosen based on the proximity to the ideal location of the aggregation point when the first source sends the information to the aggregation point. After that, each aggregation node can determine the number of downstream aggregation nodes that are dependent on it for forwarding information to the sink. The information at the aggregation node is aggregated once it has received from all the downstream aggregation nodes. Once the aggregation process is done in one section, the aggregation node act as source in the next upstream ring closer to the sink. This process is repeated till all the messages reach the sink.

## 5.2. Event-driven Data Collection

To ensure perfect aggregation of the source data at the aggregation nodes, it is also necessary that these nodes wait for an optimum delay value. In Section 3, we identified some of the drawbacks in using a fine-grained aggregation timer to trigger the aggregation process. SCT uses a more desirable alternative where there are only coarse-grained timers and the aggregation process is mainly event-driven. When an aggregation node receives information from all children that are aggregation nodes, it is assured that the data from all sources within the sector are also received by an aggregation node. This is because the sources transmit their data at the beginning of each message collection round while the aggregation nodes wait for the notification from all the downstream aggregation nodes. The arrival of messages from all downstream aggregation nodes is used as the trigger to merge and propagate the information collected, upstream towards the sink. Thus, synchronization is done in an event-driven fashion without the need for explicit delay timers at each aggregation node.

To ensure that the aggregation nodes are elected at every sector irrespective of the presence or absence of sources in that sector, we adopt the following procedure in the last ring: (i) During the query forwarding phase, some nodes in the last ring use the approach proposed in [10] to identify themselves as a corner nodes within each sector. The corner nodes are located at the periphery of the upper arc bounding a sector and are farthest from the ideal location of the aggregation node. (ii) These corner nodes take up the responsibility to identify the physical aggregation node for this sector by forwarding a dummy packet to the geographical location of the aggregation node. Note that this procedure needs to be done only for sectors in the last ring as these aggregation nodes act as sources when communicating to the upstream aggregation node.

The aggregation node identification procedure in the last ring also helps in determining the time to aggregate and forward messages to the upstream aggregation node. The arrival of messages from all the downstream aggregation nodes, is used to trigger the aggregation process in that aggregation node. In this way, the aggregation process is mainly *event driven*. Mechanisms that address the rare cases of empty sectors are also available in SCT. However, due to lack of space, we do not present them in this paper.

### 5.3. Load balancing and Node Failures

We propose two simple load balancing schemes to distribute the roles of aggregation nodes to different sets of nodes over a certain period of time:

1. *Varying the locations of the rings*: In the current SCT description, the different rings are of width $\frac{R}{m}$, where $R$ is the radius of the network and $m$ is number of rings. To do load balancing, the location of the first ring can be shifted by a distance $\frac{R}{m} - rc$, where $r$ is the one-hop transmission range and $c$ is a small integer that is varied from $0 \ldots \frac{R}{mr}$. The offset is the same for every ring so that the width of the ring is still maintained to be $\frac{R}{m}$ for all rings except the first and last.
2. *Changing the orientation of the sectors*: In a similar way, we can choose the offset angle for a sector to be different across multiple queries. The offset angle, $\theta$, can be incremented according to the relation, $\theta = \frac{c}{s(i)}$ where, $s(i)$ is the number of sectors in $ith$ ring and $c$ is a small integer dependent on the query identifier.

The design of SCT also addresses all possible sensor node failures. However, we do not present the details here due to lack of space.

## 6. Performance Evaluation

In this section we evaluate the performance of the SCT approach under different network configurations and compare it with centralized and decentralized schemes.

### 6.1. Simulation Environment

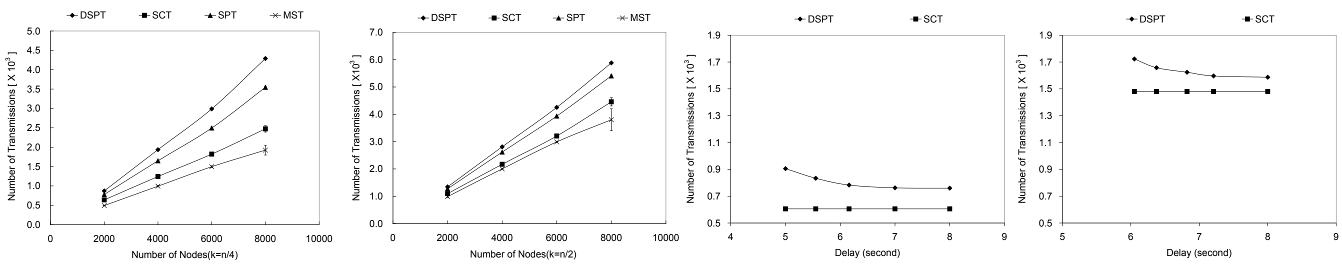
- We use a discrete event simulator for all evaluations. The simulation topologies are sensor networks with 2000 to 8000 nodes uniformly distributed within a circular field of radius 400m. The transmission range of all scenarios are $2r_0$ to ensure connectivity in case of node failures, where $r_0$ is the transmission range that guarantees minimum connectivity.
- We evaluate SCT approach using two metrics: message complexity (the total number of transmissions required for all responses to reach the sink) and data gathering latency (the time elapsed to send all the messages from the sources to the sink).
- All the simulation results are derived after averaging results over 10 random seeds and are presented with 95% confidence intervals.
- We compare SCT with an approximation of minimum Steiner tree (MST) generated using Prim's algorithm. We also compare the SCT with SPT generated with Dijkstra's algorithm because it is representative of correlation-unaware structures. To highlight the benefit of SCT as a distributed solution, we also compare it with a decentralized version of the shortest path tree (DSPT).

### 6.2. Message Complexity

We first compare the performance of the decentralized SCT with that of the centralized SPT, MST and decentralized DSPT. In this scenario, it's assumed that data from all sources are correlated perfectly ($\rho = 1$). In these simulations, we choose the total number of nodes $n$ as $2000$, $4000$, $6000$, and $8000$; and the number of sources $k$ as $\frac{n}{4}$ and $\frac{n}{2}$ respectively. To ensure fair comparison, we assume DSPT, SPT and MST use explicit mechanisms to achieve perfect aggregation, therefore the message complexity is a measure of aggregation structure efficiency only. Figure 5 (a) and (b) shows the cost of the proposed scheme and other schemes as a function of the number of nodes. It can be seen that SCT outperforms DSPT scheme under all situations. Interestingly, we observe that the cost of DSPT is up to 200% of SCT cost as the number of nodes increases, and the cost of DSPT increases faster than that of the SCT approach as node number increases. This is expected since more number of nodes reduces the efficiency of aggregation in DSPT as the paths chosen by different sources are less likely to overlap. Therefore, SCT can be considered as a more scalable approach as node number increases.

It can be observe from the figure that DSPT's message complexity is close to that of SPT while SCT's message complexity is close to the cost of MST. Furthermore, although SCT is a decentralized scheme without perfect aggregation, it still outperforms the centralized SPT since SCT does explicit data aggregation, while SPT just leverages possible aggregation implicitly. We also observe from the figures that as $k/n$ ratio increases, the difference between both decentralized schemes and their approximated centralized schemes decreases, because as k increases, both schemes can achieve better aggregation and approach the performance of ideal centralized structure.

We have also studied the message complexity of SCT when the correlation coefficient $0 < \rho < 1$. The results indicate that for both DSPT and SCT, message complexity reduces as $\rho$ increases. However, the message complexity of SCT reduces faster than that of DSPT because it facilitate aggregation at an earlier stage of packet forwarding, hence




(a) $k = n/4$ (b) $k = n/2$ (c) $n = 2000$ (d) $n = 4000$

**Figure 5. Performance Comparison Between SCT, SPT, MST and DSPT**

can reduce packet transmission cost more effectively. Since DSPT is the optimal structure when $\rho = 0$, when $\rho$ is relatively small ($\rho < 0.2$), DSPT performs better than SCT. Simulation results are omitted for lack of space.

### 6.3. Delay Sensitivity

To evaluate a data gathering scheme, latency is another important metric apart from the message complexity. Since SCT is an event-driven approach, aggregation nodes can forward message as soon as they get response from downstream aggregation nodes. For DSPT, we implement a existing scheme [9] that sets aggregation timer for each node based on their hops distance to the sink. Figure 5 (c),(d) shows the number of transmissions as a function of maximum delay (which is proportional to the depth of the aggregation tree) for $n = 2000$ and $n = 4000$ cases. It is shown that DSPT achieves perfect aggregation when maximum delay is more than 8.0 second. However, the cost of DSPT increases as maximum delay approaches 5.0 second since smaller maximum delay can increase the possibility of late arrivals of data for aggregation. Once a packet misses aggregation deadline at one of the intermediate hops, the probability of it missing deadlines at later hops is also high, which is the reason that DSPT message complexity increases quickly as maximum delay decreases. On the other hand, since SCT is an event driven data aggregation structure, its message complexity remains the same irrespective of different delays, which explains the flat curves in both figures.

## 7. Conclusions

In this paper, we propose a novel solution for aggregating correlated information from a subset of sensors to the sink. The proposed scheme is scalable, distributed, requires minimal computation and is highly-manageable compared to existing solutions. Simulation results show that SCT performs significantly better than correlation-unaware structures in terms of message complexity and delay performance.

## References


[1] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, “On Network Correlated Data Gathering,” in *INFOCOM*, Hong Kong, Mar. 2004.

[2] 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.

[3] Anupam Gupta, Martin Pal, R. Ravi, and Amitabh Sinha, “Boosted Sampling: Approximation Algorithms for Stochastic Optimization,” in *STOC*, 2004.

[4] Vikram Mhatre and Katherine Rosenberg, “Design Guidelines for Wireless Sensor Networks: Communication, Clustering and Aggregation,” in *Elsevier Science Journal*, Aug. 2003.

[5] C. Intanagoniwawat, R. Govindan, and D. Estrin, “Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks,” in *MOBICOM*, Boston, MA, USA, Aug. 2000.

[6] B. Karp and H. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks,” in *In Mobile Computing and Networking*, 2000.

[7] H. Takahashi and A. Matsuyama, “An Approximate Solution for the Steiner Tree Problem in Graphs,” in *Math. Japonica 24*, 1980, vol. 6, pp. 573–577.

[8] 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 Algorithms*, 2003.

[9] W. Yuan, S. V. Krishnamurthy, and S. K. Tripathi, “Synchronization of Multiple Levels of Data Fusion in Wireless Sensor Networks,” in *IEEE GLOBECOM*, 2003.

[10] Tian He, Chengdu Huang, Brian Blum, John A Stankovic, and Tarek Abdelzaher, “Range-Free Localization Schemes for Large Scale Sensor Networks,” in *MOBICOM*, 2003.