Available online at www.sciencedirect.com
ScienceDirect
Information Fusion 9 (2008) 354–369
INFORMATION FUSION
www.elsevier.com/locate/inffus

A scalable correlation aware aggregation strategy for wireless sensor networks

Yujie Zhu [a,*], Ramanuja Vedantham [a], Seung-Jong Park [b], Raghupathy Sivakumar [a]

[a] *School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332, USA*
[b] *Department of Computer Science Louisiana State University, Baton Rouge, LA 70803, USA*

Received 8 September 2005; received in revised form 4 September 2006; accepted 23 September 2006
Available online 13 November 2006

Abstract

Sensors-to-sink data in wireless sensor networks (WSNs) are typically characterized by correlation along the spatial, semantic, and/or temporal dimensions. Exploiting such correlation when performing data aggregation can result in considerable improvements in the bandwidth and energy performance of WSNs. In this paper, we first identify that most of the existing upstream routing approaches in WSNs can be translated to a correlation-unaware data aggregation structure – the shortest-path tree. Although by using a shortest-path tree, some implicit benefits due to correlation are possible, we show that explicitly constructing a correlation-aware structure can result in considerable performance improvement. Toward this end, we present a simple, 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.

Keywords: Sensor networks; Data aggregation; Information fusion; Correlation; Data gathering

1. Introduction

Wireless sensor networks (WSNs) have gained tremendous importance in recent years because of their potential use in various fields. However, the devices used for sensing and communication in these networks are usually small, cheap and low-powered and hence, have limited resources for computation as well as for communication. This has spurred a need for efficient protocols tailored specifically towards sensor network environments.

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 cost 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 the different sensors are *correlated*. Such correlation of data being collected can be leveraged by appropriately fusing the data in the network to the best extent possible. Hence, the specific problem addressed in this work can be stated as: *How should an energy-efficient data gathering structure be constructed to leverage any existing correlation between data reported by the sensors?*

Many research work have proposed solutions to construct correlation-aware structures [1–3]. 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.

* Corresponding author.
E-mail addresses: zhuyujie@ece.gatech.edu (Y. Zhu), ramv@ece.gatech.edu (R. Vedantham), sjpark@bit.csc.lsu.edu (S.-J. Park), siva@ece.gatech.edu (R. Sivakumar).

1566-2535/$ - see front matter
doi:10.1016/j.inffus.2006.09.002

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 does not require any knowledge of the number of sources or their locations. The SCT approach, with its highly manageable structure, ensures low maintenance overhead of the aggregation structure, eliminates the need for global synchronization among sensors for aggregation of sensor-data, while also addressing the other challenges described in Section 4 including load balancing and node failures. Through simulations and analysis, we establish the message costs incurred by SCT for a variety of network conditions, and compare them with an ideal correlation-aware and a correlation unaware approach. We show that SCT, though simple in its realization, can achieve substantial performance benefits.

The rest of the paper is organized as follows: Section 2 defines the problem and Section 3 discusses existing data gathering structures and analyzes their characteristics. Section 4 identifies the different challenges in designing a practical, efficient correlation-aware structure. Section 5 presents the key design principles in the SCT approach and describes how it addresses the corresponding challenges. Section 6 explains the SCT approach in detail. Section 7 evaluates the performance of SCT in comparison to ideal structures and practical implementation of shortest path tree (SPT). Section 8 discusses the issues pertaining to the SCT approach while Section 9 concludes the paper.

2. Problem definition

We consider a multi-hop WSN with one sink at the center and n sensors distributed randomly in a circular field according to a Poisson process.[1] The sink sends a query and k of the n sensors respond to the query. We refer to these sensors that have information to send as sources in the rest of the paper. 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 cost* as the total number of transmissions required for responses from all k sources to reach the sink. Our primary goal is to minimize message cost when there is correlation present between data from different sources.

The following two types of correlations are considered in this paper:

- *Spatial correlation:* This refers to the correlation of the data reported by multiple sensors sensing the same event or phenomenon. For example, consider the query: what is the temperature in the region defined by the rectangle $(x1, y1, x2, y2)$? Given the typical dense deployments of sensors in WSNs, it is likely that the sensing regions of two different sensors within the rectangular region overlap. Consequently, the data reported by these sensors are spatially correlated. If the two sensors are very close to each other, the data reported by both sensors is practically the same, which implies that the sensors are perfectly correlated.
- *Semantic correlation:* This refers to the correlation of data reported by multiple sensors due to the semantics of the query. The messages from different sensors reporting different events or phenomena and hence the content of these messages may not be spatially correlated. However, if the query imposed is not about the specific details of each event, but about certain statistics over the entire event region, it is likely that the messages generated by each source can still be aggregated. For example, consider the query: *Is the total number of cars in the rectangular region $(x1, y1, x2, y2)$ greater than K?* In this case, even if the sensors are reporting data about *different* cars, the information reported is correlated as it is only required to find the total number of cars and consequently determine if it is greater than K. Responses to statistical queries such as *min*, *avg*, *max*, usually fall under this category.

Characterizing the correlation existing between sensor data is a fairly complicated task, since the nature of correlation differs with the type of applications considered. Even for a simple correlation model, the mathematical representation becomes difficult when multiple distributed sources are involved. For simplicity, we adopt the same correlation model used in reference [1], where each raw data packet is assumed to bring a fixed amount of new information into the aggregated data packet. Specifically, if ρ is defined to be the correlation degree, and m to be the sizes of raw data packets generated by sensor nodes, then after aggregation of two data packets, the message size becomes $m + (1 - \rho)m$. Similarly, for n sources, the aggregated data packet has a size of $m + (n - 1)(1 - \rho)m$. Correlation degree $\rho = 1$ means that two messages are perfectly correlated (i.e. semantic correlation) therefore can be reduced to one message of the same size. Correlation degree $0 < \rho < 1$ indicates that two messages are partially correlated (i.e. spatial correlation), while $\rho = 0$ implies that two messages are independent of each other.

3. Related work

3.1. Correlation unaware approaches

We now consider two of the popular choices used in the design of routing protocols in sensor networks: (i) using the

[1] Note that the assumptions about the shape of the sensor field and the location of the sink are made for better illustration of the proposed approach and are not essential to the solution. We will discuss the implications of different network shapes and sink locations in Section 8.

query paths to construct the sensors-to-sink routes and (ii) using the location information of sensors and the sink to forward messages to the sink. Most of the contemporary routing approaches [5–7] fall under one of the two categories in terms of basic routing design principle. We consider directed diffusion [5] and GPSR [6] as representative examples of (i) and (ii), respectively, and show that the structure generated by these approaches can be translated to a Shortest Path Tree (SPT).

In directed diffusion [5], sink requests data by sending interests for named data. The interests are propagated through the network wherein every node delivers the interest to all its neighbors through a local broadcast. During the diffusion process, all nodes set up gradients towards their neighbors from which the interest was received. While there are different possible criteria for the reinforcement of the gradient, the most common practice results in a shortest path tree rooted at the sink [1,8].

GPSR is a geographic routing protocol that eliminates the need to maintain states while performing routing. In this approach, every source in the sensor field knows the geographical location of the sink, and addresses the message for sink with the specific location. For each hop along the path to sink, a node chooses the node closest in geometric distance to the sink as the next hop destination and forwards the message to it. Since GPSR delivers a vast majority of packets in the optimal number of hops [6], the data gathering tree again approximates shortest path tree.

From the previous discussion we establish that two of the representative routing schemes in sensor networks are approximations of shortest path trees. Since the primary goal of this structure is to minimize delay, 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. A correlation-aware structure, on the other-hand, would be able to cut down message cost by explicitly facilitating data aggregation in the network.

3.2. Correlation-aware approaches

Several related work have been proposed in the context of explicit aggregation [1–3,8]. We categorize them into two general classes: correlation-aware structures assuming complete global source knowledge and incomplete source knowledge.

3.2.1. Structures built with complete source knowledge

When the full knowledge about source location is known, the Steiner tree over all sources, sink and non-source nodes gives the optimal message cost when the degree of correlation is close to 1. However, the computation of Steiner tree is an NP-hard problem [9].

In [1,8], the authors propose simple heuristics that approximate the Steiner tree to perform efficient aggregation when messages are correlated. For any given correlation factor $0 < \rho \leqslant 1$, the authors describe (i) leaves deletion heuristic and (ii) balanced SPT/multiple Traveling Salesman Problem (TSP) tree as simple alternatives of the Steiner tree.

In [10], the authors first identify that the message cost can be modeled as a concave-cost function for any correlation factor $0 < \rho < 1$, and propose an algorithm that constructs approximation trees simultaneously good for all concave cost functions.

However, these approaches require complete information regarding the number of sources and their location to be available at the sink and cannot work for the cases when the information is incomplete. While such information can be made available via queries and responses, the overheads involved in acquiring such information both in terms of message cost and delay could be potentially prohibitive. Moreover, they are centralized approaches hence do not scale well with increasing node densities typical to WSN environments. Finally, these approaches try to solve the problem of efficient aggregation from a theoretical perspective and do not consider the practical challenges that we identify in Section 4.

3.2.2. Structures built with incomplete source knowledge

[2,3] address the more general problem of building aggregation structure with optimal expected cost when the knowledge of sources is incomplete.

The problem considered by both work is a network with a root node and a collection of N client nodes in the network. Each client, i, may choose to contact the root independently from others with some probability p_i along a path from itself to the root. If a client chooses to contact the root, the edges on this path become active. The goal is to minimize the expected number (or cost) of active network edges over a random choices of the clients. This problem is a stochastic version of the deterministic Steiner tree problem.

The stochastic Steiner tree problem is an NP-complete problem. Therefore, the focus of this work is on developing constant-factor approximation algorithms. In [2], the authors observe that the optimum solution is invariably a tree, and the optimum tree consists of a central "hub" area within which all edges are almost certainly used, together with a fringe of "spokes" in which multiple clients contribute independently to the cost of the solution. To set up a good approximation structure, their solution leverages a facility location algorithm to identify a good set of "hubs" to which clients route messages at independent costs, and the set of hubs are connected using a Steiner tree algorithm. In [3], the authors design a similar structure constructed in two stages: during the first stage, a subset of nodes D is chosen by picking each node independently from the network with probability proportional to p_i, then a minimum spanning tree is built over D; later revealed clients requesting services are connected to the existing structure with an augmentation algorithm. Using the above principle, the authors propose a two-approximate algorithm for the stochastic Steiner tree problem.

Both papers propose good constant-factor approximation algorithms. However, they are not distributed solutions and are not tailored for sensor network environments. For example, [2] requires building a Steiner tree over "hubs", and [3] requires building a minimum spanning tree to form a first-stage backbone. Both need centralized computation with high complexity. Also, they assume a fixed probability for nodes responding to a query and do not handle the situation where the source density is variable. Finally, they do not address all the practical challenges we identified in Section 4.

4. Challenges

The main goal of this work is to design an efficient aggregation structure that minimizes the message cost. To realize this goal, we first identify the following important challenges and list the desirable properties of a solution that addresses the challenges.

4.1. Construction

The foremost consideration in building an aggregation structure is the manner in which the aggregation structure is constructed. We have already seen that existing approaches in WSNs are correlation unaware and approximate a shortest path tree (SPT). In a SPT, aggregation is not efficient even if the data from the sources are highly correlated because the primary concern is to minimize the delay. The ideal solution for the aggregation structure would be a Steiner tree or a stochastic Steiner tree [2], which depends on whether complete source knowledge is available at the time of construction. However, both problems are NP-hard.

Even the approximation algorithms for constructing a Steiner tree impose requirements such as the information regarding the number of sources and the location of them to be available at the sink *a priori*. This information could 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 location 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 cost of the data transmitted. However, the overhead involved is $O(k)$, where k is the number of sources. This overhead might be comparable or even exceed the message cost of the aggregated messages. Another drawback of such a two-phase approach is the delay incurred in determining required information.

Moreover, the types of queries and responses sent may influence the nature of the ideal solution required for aggregation. The queries and responses can be categorized as (1) single query and one-time responses from the sources, (2) single query and multiple continuous responses from the same set of sources and (3) single query and multiple responses from a varying set of sources. We will explain each of these classifications briefly and present the ideal solution for each classification:

- *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 cost, since the message cost of the first phase will be comparable to the message cost of the aggregated responses. 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 comprise multiple packets. However, the set of sensors responding to the query remains the same over all the packets. If the number of continuous responses is large, the cost incurred in determining the number of sources in a two-phase approach could be amortized over the message costs involved in aggregation. In this case, the network Steiner tree is the optimal aggregation solution.
- *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 location of sources or the number of sources. Therefore, the optimal solution is neither a network Steiner tree nor a Stochastic Steiner tree problem as the set of sources and their probability can vary for different packets. 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 cost of the aggregation structure on the other hand, and ensure that it is reasonably efficient across all query and response paradigms.

4.2. Maintenance

An aggregation structure may be modified or reconstructed after a certain period of time to accommodate load balancing, node failures or for any other reasons.

Load balancing is to ensure that the energy consumed by all the nodes is fairly even over a certain period of time. In an aggregation structure, aggregation nodes take the responsibility of receiving, computing and transmitting of aggregated messages, hence consume more energy than non-aggregation nodes. If the role of aggregation node is performed by a certain node for a long time, this node may fail much earlier than other nodes, impacting the connectivity of the network. To address this issue, we would like to spread the roles of aggregation nodes among all nodes so that the network does not become disconnected prematurely.

The aggregation structure should also be resilient to node failure, which is very common in sensor networks. 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.

One way to reconstruct the structure is where the nodes send explicit beacons to the central coordinator requesting a structure modification, and the coordinator reconstructs the structure partially or globally according to the number of nodes requesting such changes. However, this approach requires central coordination and may incur large overheads. In a more desirable approach, nodes should be able to reconstruct the aggregation structure in a distributed fashion with low overheads and delay.

4.3. 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 as synchronization requirements. In the absence of such timing requirements, 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 cost 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 timer at every node and wait for the expiration of the timer based on a waiting function, similar to the one described in [11], before performing aggregation. In this approach, each node waits for a time corresponding to $MAX - i \times \Delta$ after the reception of the first response, where MAX is the maximum wait time proportional to the depth of the aggregation tree constructed, i is the hop count of the node from the sink and Δ is the average degree of a sensor node. When the timer expires, it is assumed that all the messages from sources downstream have arrived at this node and the messages are aggregated. One of the obvious problems of this approach is that the nodes use the average degree of the network as opposed to the degree of that particular node. This makes the timer value not accurate and may cause imperfections in aggregation. On the other hand, the problem with incorporating a fine-grained timer is the higher message overhead of time synchronization, and complicated computation at each sensor node.

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.

4.4. Other considerations

An aggregation approach should also be reasonably efficient in terms of message cost when the degree of correlation varies ($0 < \rho \leqslant 1$). In addition, it should be able to perform efficient aggregation irrespective of the distribution of sources. As we will see in Section 7, the proposed approach takes into account these considerations and performs reasonably well for a wide range of network scenarios.

5. The SCT design basis

The design of SCT is predicated on two important elements:

- An aggregation backbone facilitating the generation of efficient aggregation trees
- A fixed structure independent of source distribution and density.[2]

These two design elements address the challenge of efficient construction, 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 present in Section 6.

5.1. Motivation for a ring-and-sector division

Our target problem of data collection with variable source distributions and densities can be thought of as a generalized version of the stochastic Steiner tree problem. The stochastic Steiner tree gives the optimal message cost when there is a given source probability while our goal is to find the efficient aggregation tree for variable source probabilities. In [2,3], the authors have presented centralized constant-factor approximations to the stochastic Steiner tree problem. In this subsection, we will provide motivation of the structure we used to approximate the generalized stochastic Steiner tree, adopting similar arguments provided by the above two related work.

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

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

[2] Ratio of the number of sensors that send data packets to the sink to the total number of sensor nodes in the network.

This cost function captures the tradeoffs in the characteristics of data transmission and correlation in sensor networks as follows: when more sources use one particular edge, the total communication cost for this edge increases; on the other hand, the cost per message decreases due to the correlation between multiple messages. Since the 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. The sources can then connect opportunistically to closest aggregation nodes using edges with a higher cost. With this structure, the total expected cost of the aggregation tree for a certain source probability distribution can be minimized.

Based on these observations, a good aggregation structure should have the following features: (i) 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; (ii) Each node in the backbone is responsible for aggregating messages from sources within a certain sub-area.

There are many ways to construct the aggregation backbone. In this paper, we design a uniform division of the network to assist in the construction of the backbone, based on the following criteria:

- The division of the network should facilitate the distributed selection of aggregation nodes within the network with no additional message overhead.
- The division should result in energy-efficient data aggregation irrespective of distribution of sources.
- The interconnection of aggregation nodes to form the backbone should be achieved in a distributed fashion with low overhead.
- The division of the network should ensure easy connectivity of all non-backbone nodes to the aggregation backbone.

In SCT, one such efficient division of the network is adopted, where the network is divided into rings and sectors. The division not only addresses *all of the above* desired criteria, but is also characterized by several impressive practical benefits, including *lack of global synchronization requirement* for aggregation of sensor-data, *ability to perform load balancing* with no additional message overhead, *ability to address node failures*, etc. *The ring-sector division is also characterized by the unique invariant that a message sent by any source always propagates towards the sink at each hop, because of the symmetrical structure of the division*. We elaborate more on the motivations of ring-sector division in SCT later in this section and in Section 6.

As illustrated in Fig. 1, the network is divided into m concentric rings with the same width (R/m). Each ring is in turn divided into sectors of the same size such that on average each sector contains about n_0 nodes. For each sector, an aggregation node is chosen as a member of the aggregation backbone, and each 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 form the backbone aggregation tree. Each aggregation node is responsible for collecting messages from all sources in the sector it belongs to.

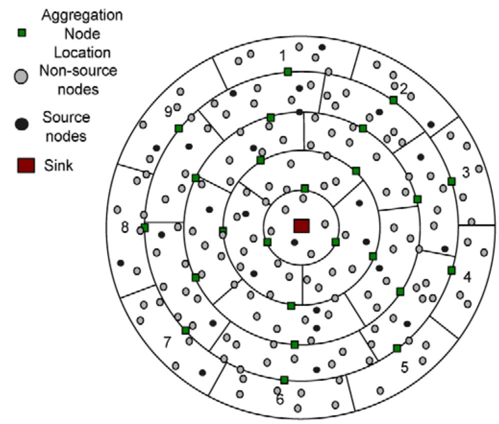

Fig. 1. SCT structure: rings, sectors and aggregation nodes.

As we will see in the Section 6, this structure can facilitate the realization of a 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. Therefore, 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 optimal backbone aggregation nodes increases. This can be explained intuitively as follows: 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 backbone with higher cost, because of the additional transmissions required by the aggregation nodes. Therefore, fewer aggregation nodes are more desirable.

However, as source number increases, the probability that two or more messages from sources being 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 reduced high-transmission cost edges used due to early aggregations. Hence, as source density increases, more aggregation nodes are desirable.

As we identified above, a good approximation of optimal structure should adapt with different source densities: the higher the source density, the more nodes are involved in the backbone. This indicates that the proposed ring-sector structure should also adapt to source densities in order to approximate the optimal solution. However, in the next subsection, we will show that a fixed structure satisfying certain properties is reasonably efficient for a wide range

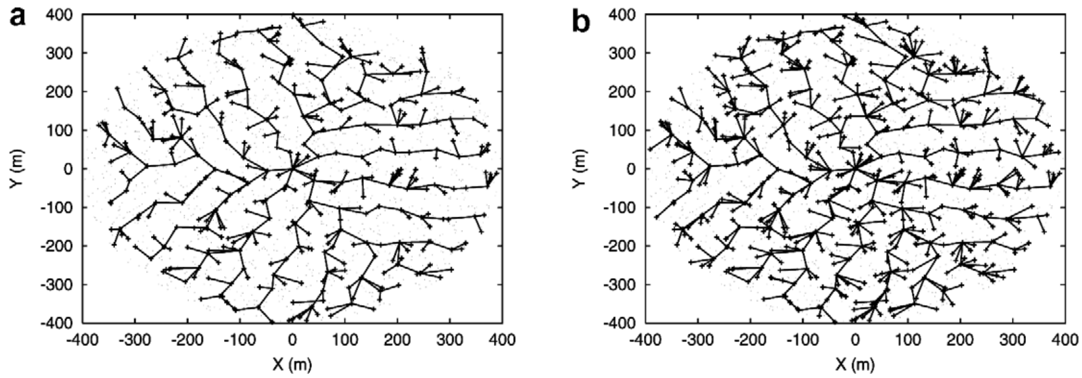

Fig. 2. The optimal structures when $n = 2000$.

of source densities, and hence motivate a relatively stable aggregation structure with low maintenance overhead.

5.2. Motivation for a source-independent aggregation structure

In the previous sub-section we pointed out that for the ring-sector structure proposed, the optimal number of aggregation nodes increases with the source density. In this sub-section, we will show that when the source density is beyond a certain point, the optimal structure stays the same because of a "saturation" phenomenon. Thus, by choosing a fixed aggregation structure, we are able to do efficient aggregation for a large range of source densities while incurring very little construction overhead.

The optimal sector size generally reduces with increasing source density. But this reduction is not always desirable. Consider a certain threshold source number k_0 at 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 transmission. Reducing sector beyond the saturation size will not help increasing aggregation efficiency. Furthermore, the introduction of additional aggregation nodes increases the message cost. Thus, when the number of sources, k, is increased beyond k_0, decreasing the sector size results in increased message cost. Therefore, when $k > k_0$, the optimal aggregation structure should remain the same. Fig. 2(a) and (b) are visualizations of the aggregation trees when each sector reaches saturation size. These figures show the SCT aggregation tree constructed over all source nodes and aggregation nodes. The scattered points on the background are non-source, non-aggregation nodes. Fig. 2(a) is the optimal aggregation tree when $n = 2000$ and $k = 250$, and Fig. 2(b) is the optimal structure when $n = 2000$, and $k = 500$. From the two figures we can observe that the "backbone" of the aggregation trees are the same, only the sources increase in the latter case. This is because saturation is already achieved when $k = 250$. Therefore, even if k increases, the optimal structure remains the same. Notice that the above "saturation" phenomenon is not specific to the ring-sector structure we proposed, but is true for the optimal aggregation structure in general due to the fixed transmission ranges of all sensor nodes.

The effect of saturation on message cost is also substantiated by other simulation results. Fig. 3(a) shows for a certain n and k how the message cost of aggregation tree varies with varying sector size n_0. When n_0 is small, the message costs of the aggregation trees decrease as n_0

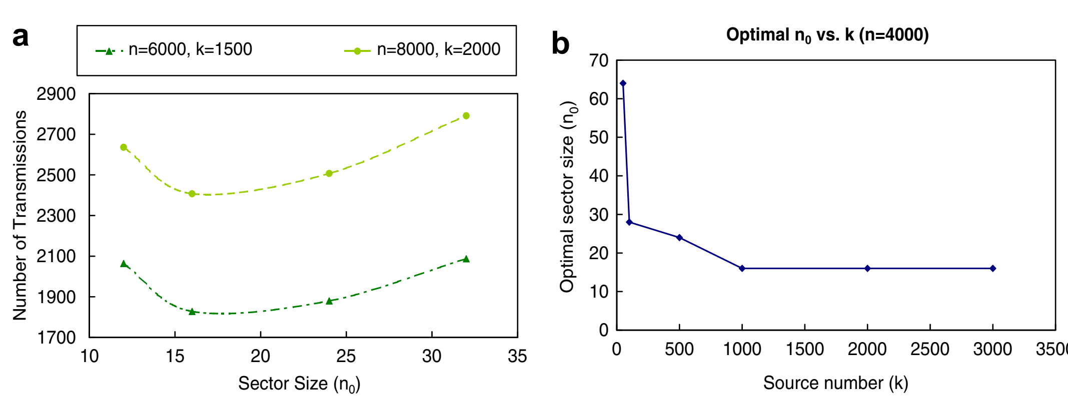

Fig. 3. Find the optimal sector size.

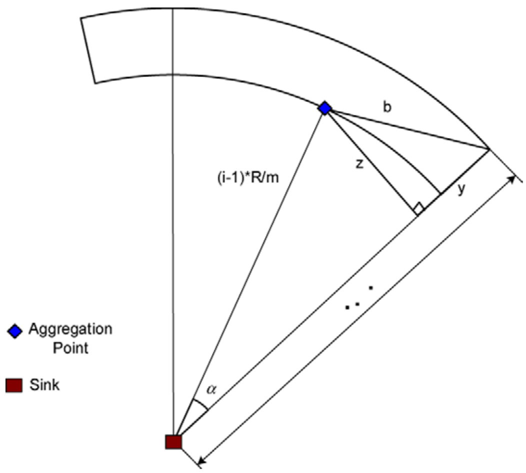

Fig. 4. Maximum travel distance within a sector.

increase. However, after $n_0 = 16$, costs increase with increasing n_0, indicating saturation for both cases.

Fig. 3(b) shows how the optimal n_0 varies with k/n. For a wide range of source densities, the optimal structure is the same. This implies that a fixed aggregation structure is good enough most of the time in our application. Simulation results for other node densities give similar results.

We now describe how the optimal values for m[3] and n_0[4] can be estimated at the sink. Both parameters are chosen such that each sector in the network reaches the saturation size. The choice of m and n_0 is explained next.

The choice of m determines the depth of the aggregation tree. Since the downstream aggregation nodes are sources for the aggregation nodes one ring closer to the sink, to achieve saturation size, level-i aggregation nodes should be within 1-hop distance to level-$i+1$ aggregation nodes. Therefore, m 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 β is a constant used to accommodate the fact that aggregation nodes are not distrubted on a straight line. We choose a empirical value of 1.32 for β based on experimental results.

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. Refer to Fig. 4, the furthermost possible node within this sector is located at the corner of this sector, we indicate the distance of this node to aggregation point as b. The distance b is calculated in the right-angled triangle as follows:

[3] The total number of rings in network, or the aggregation levels.

[4] The average number of sensor nodes in each sector, determines the sector size.

Table 1
Comparison of computed and experimental m^* and n_0^*

Node density	Computed m^*	Experimental m^*	Computed n_0^*	Experimental n_0^*
$n = 2000$	8.54	9	13.9	16
$n = 4000$	11.6	12	14.9	16
$n = 6000$	14.1	15	15.5	16
$n = 8000$	16.1	17	16.0	16

$$z = \frac{(i-1)R\sin\alpha}{m} \tag{3}$$

$$y = \frac{iR}{m} - \frac{(i-1)R\cos\alpha}{m} \tag{4}$$

$$b = \sqrt{z^2 + y^2} \tag{5}$$

$$= \frac{R}{m}\sqrt{(4i(i-1)\sin^2\frac{\alpha}{2}) + 1} \tag{6}$$

The angle α for the ith ring is given by:

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

When $\alpha \to 0$, $sin\frac{\alpha}{2} \to \frac{\alpha}{2}$, and Eq. (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 $\sigma < 1$ is a constant. From Eqs. (8) and (2), and we can derive the optimal n_0^* for large k as:

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

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 Eqs. (2) and (9) and those obtained from simulations. Table 1 shows that the theoretical values match experimental optimums closely.

6. The SCT approach

In this section, we explain the SCT approach in detail. We present the different phases of the data aggregation process as well as provide insights for the design choices.

6.1. Division of the network

During the setup phase, the sink propagates the following information to the entire network: (i) location of itself, (X_s, Y_s), (ii) the total number of nodes, n (iii) the radius of the network, R and (iv) the computed values for m and 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 the packet, it first computes the distance between itself and the sink. This determines the ring, i, to which it belongs to. For example, any node at a distance d from the sink, such that $(i-1)\frac{R}{m} < d \leqslant i\frac{R}{m}$, belongs to the ith ring. Each node can calculate the number of sectors per ring $s(i)$ as: $s(i) = \lceil \frac{2(i-1)n}{n_0 m^2} \rceil$. Given the locations of the node with respect to the sink and the number of sectors within the ring, any node can determine the sector number to which it belongs.

6.2. Determination of aggregation nodes

Once the network has been divided into sectors and rings, the aggregation node for each sector has to be selected. In the SCT approach, the aggregation nodes are selected by leveraging the fixed, geometric division of the sensor field.

For each sector in a given ring, the geometric center of the lower arc bounding the sector is defined as the ideal location of the aggregation node for this sector. The node closest to this ideal location is chosen as the aggregation node. Given the value of m and n_0, each source not only knows the sector and ring numbers to which it belongs but can also determine the boundaries of the sector. If we are to adopt polar coordinates and if α and β are the bounding angles of a sector corresponding to the ith ring, the location of the aggregation node is given by $\left((i-1)\frac{R}{m}, \frac{\alpha+\beta}{2}\right)$.

Each source within this sector routes messages to this location of the aggregation point. During the query forwarding phase, each node piggybacks the coordinates of itself along with the query and other information sent by the sink. In this fashion, a node can learn the locations of its immediate neighbors during the query delivery phase. When a source wants to send it's message to the ideal location of the aggregation point, it sends the packet to the node closest to the ideal location. Once a node receives this packet, it then does a local broadcast declaring itself as the aggregation node. Since most of the nodes within any sector are in a one-hop region of the aggregation node, the other source nodes will then forward their packets to this aggregation node. In this way, the node closest to this ideal location becomes the aggregation node where messages are aggregated. In a few cases, it is possible that two sources may elect two different nodes as aggregation nodes simultaneously. However, such instances are rare and do not increase the message cost considerably because of the following two reasons: (i) most of the nodes in a sector are within a 1-hop region of each other, and (ii) once a node is elected as the aggregation node, the local broadcast will enable other sources to identify this node as the aggregation node.

Thereafter, these aggregation nodes act as sources for the sectors in the next ring closer to the sink, and send aggregated messages to aggregation nodes of those sectors. With this approach messages are combined step by step as they progress towards the sink, until the last ring is reached, for which the sink is the aggregation node. Fig. 5 is a illustration of this aggregation process.

The aggregation nodes are implicitly elected by the routing protocol when the sources first send their data to the location of the aggregation nodes. To ensure that an aggregation nodes is 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 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 farthest from the ideal location of the aggregation node. (ii) These corner nodes take on 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 nodes.

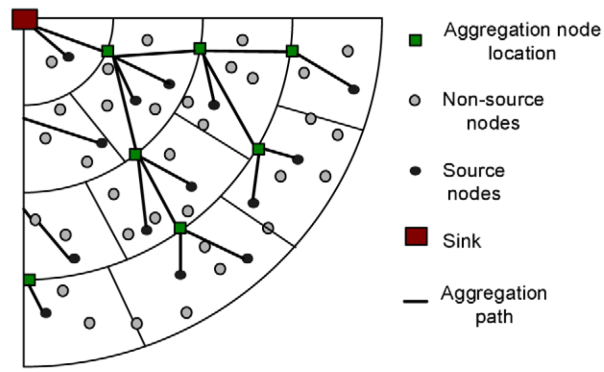

Fig. 5. A Subsection of the SCT aggregation structure.

This geometric election of the aggregation nodes facilitates independent and distributed computation of aggregation point locations by each node. It also enables a location based routing approach to be used to send packets from the sources to the aggregation nodes or between aggregation nodes. The geometric structure construction and aggregation nodes election allow each node to have a consistent view of the entire network to help set up the SCT structure in a distributed fashion.

6.3. Event-driven data collection

To achieve maximum aggregation of the source data at the aggregation nodes, it is also necessary to ensure that these nodes wait for an optimum delay value. In Section 4, we identified the drawbacks of using a fine-grained aggregation timer to trigger the aggregation process. In SCT we use a more desirable alternative where there are only coarse-grained timers and the aggregation process is mainly event-driven. This approach is motivated by the fact that each aggregation node knows the exact number of children that are also aggregation nodes. 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 notifications 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 achieved in an event-driven fashion without the need for explicit delay timers at each 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. Even if there are no sources within a sector, the aggregation node sends out a small message to the upstream sector indicating the absence of sources. 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*. For the case when there are no nodes in a downstream sector, there will not be any downstream aggregation node in that sector. In these rare cases, the aggregation nodes use the coarse-grained aggregation timer before aggregating the messages from other downstream aggregation nodes and sources. Note that the probability of occurrence of such empty sectors is extremely low as we will discuss in Section 8.

6.4. Load balancing

Load balancing schemes, as we have identified in Section 4, are important to ensure that the resources of all non-source nodes are utilized to roughly the same extent over a period of time. We propose two simple schemes to distribute the roles of aggregation nodes to different sets of nodes over a certain period of time:

(1) *Location 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 $0 \cdots \frac{R}{mr}$. The same offset is applied to 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) *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, θ, can be incremented according to the relation, $\theta = \frac{c}{s(i)}$ where $s(i)$ is the number of sectors in the *ith* ring and c is a small integer dependent on the query identifier. This again assures that different nodes are chosen as aggregation nodes over several query floods.

6.5. Aggregation reliability and node mobility

The failure of any non-aggregation node will not impact the correctness or efficiency of the SCT approach. Therefore, here we only discuss how the aggregation node failures are addressed in SCT. Recall that we require an aggregation node to announce itself to its neighbors once it receives the first message from a source. In this way, aggregation node failures before the setup of the SCT structure can be identified by the lack of announcement from the particular node, and another node closet to the ideal location can announce itself as the aggregation node of this sector. If an aggregation node fails during the information collection phase, the lack of ACK messages from this node to sources can inform them of its failure. In this case, the retransmission of the first packet enables the election of a new aggregation node, which in turn broadcasts an announcement upon receiving the retransmitted message. After the re-election, aggregation proceeds as usual. Notice that the election of a new aggregation node may delay the entire aggregation process due to the retransmissions and announcement necessary, but the correctness and efficiency of the aggregation process will not change since it is event-driven.

In the case of node mobility, the proposed solution needs to be modified to accommodate mobility in (i) sources, (ii) aggregation nodes and (iii) other nodes. If the sources are mobile but the aggregation nodes are fixed, sources will forward to the closest aggregation node given its current location. If aggregation nodes are mobile, the node failure handling mechanism can be leveraged to elect a new aggregation node for that sector. Mobility of other nodes does not affect the SCT approach.

7. Performance evaluation

In this section we evaluate the performance of the SCT approach under different network configurations and compare it with two centralized schemes: minimum Steiner Tree, SPT, and one decentralized scheme: DSPT (Decentralized Shortest Path Tree). We vary the node density, source density, source distribution, as well as correlation coefficient (ρ) and evaluate the message cost of the four structures under different scenarios.

7.1. Simulation environment

- We use a discrete event simulator based on the LECS simulator for all evaluations. The simulation topologies are largely similar to those used in general sensor networks: 2000–8000 nodes are uniformly distributed within a circular field of radius 400 m. The number of sources that generate messages for one specific query varies from $\frac{1}{10}$, $\frac{1}{6}$, $\frac{1}{4}$ to $\frac{1}{2}$ of the total number of nodes in the network.
- We evaluate the SCT approach using two metrics: message cost and data gathering latency. For message cost, we measure the total number of transmissions required for all responses to reach the sink for one round of data collection, and for data gathering latency, we measure the time interval from the time when all sources start to send messages, to the last message reaches the sink.
- To focus on the comparison of aggregation efficiency of different structures, we assume a perfect MAC layer that avoid collisions of data packets.

- All the simulation results are derived after averaging results over 10 random seeds and are presented within 95% confidence intervals.
- Since minimum Steiner tree is the optimal solution when sources are fixed, we compare SCT with an approximation of minimum Steiner tree (MST) generated using Prim's algorithm. We also compare the SCT with SPT generated by Dijkstra's algorithm because it is representative of correlation-unaware structures. To highlight the benefit of SCT as a distributed solution, a decentralized version of the shortest path tree (DSPT) is also included in the evaluation. In DSPT, GPSR routing protocol is used to approximate SPT in a distributed fashion because the routes generated by GPSR closely approximate SPT, especially when node density is high [6].

7.2. Perfect correlation ($\rho = 1$) scenarios

7.2.1. Different node densities

We first compare the performance of the decentralized SCT with that of the DSPT. In this scenario, we assume that data from all sources are correlated perfectly ($\rho = 1$). We will address the ($\rho < 1$) case later in this section.

Fig. 6(a)–(c) show the cost of two decentralized schemes as a function of the number of nodes for different numbers 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 ensure fair comparison, we assume DSPT uses an explicit mechanism to achieve perfect aggregation, therefore the message cost is a measure of the aggregation structure efficiency only (we will present results related to delay performance of both schemes later). 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 the SCT cost as the number of nodes increases. The DSPT cost also increases faster than the SCT cost as node number increases. This is expected since more 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 a more scalable approach. Furthermore, it is observed that the difference between the two schemes increases as the ratio of the number of sources to the number of nodes, $\frac{k}{n}$, decreases because more sources increase the probability of aggregation for DSPT. As the ratio $\frac{k}{n}$ approaches *1*, both schemes converge into n transmissions.

7.2.2. Decentralized vs. centralized schemes

We compare the performance of the decentralized SCT and DSPT schemes with the centralized schemes they approximate. Fig. 7 shows the cost of the proposed scheme and the centralized schemes as a function of the node number. To evaluate the cost of the centralized schemes, we assume perfect aggregation for both SPT and MST. From the figures we can see that DSPT's message cost approaches closely that of SPT while SCT's message cost approaches closely the cost of MST. Furthermore, although SCT is a decentralized scheme without perfect aggregation, it still outperforms the centralized SPT, since SCT explicitly aggregates sensor data, while SPT just leverages aggregation opportunistically. We also observe that as the 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 an ideal structure.

7.3. Different correlations ($0 < \rho < 1$)

In the extreme case of $\rho = 1$, minimum Steiner tree is the optimal aggregation structure, while for $\rho = 0$ (no correlation), SPT is the optimal aggregation structure. For the general case $0 < \rho < 1$, no optimal solution exists. And the problem is classified as NP-complete in [1].

In Fig. 8, we characterize the message complexity of SCT when the correlation coefficient varies from 0.2 to 0.9. Notice that in this graph, the number of transmissions is normalized to a unit message size. For example, if after aggregation, a node transmits a message of size 1.5 times the unit message size, it is counted as 1.5 transmissions. From this figure, we can see that for both DSPT and SCT, message cost reduces as ρ increases, since the two schemes have either implicit or explicit mechanisms to

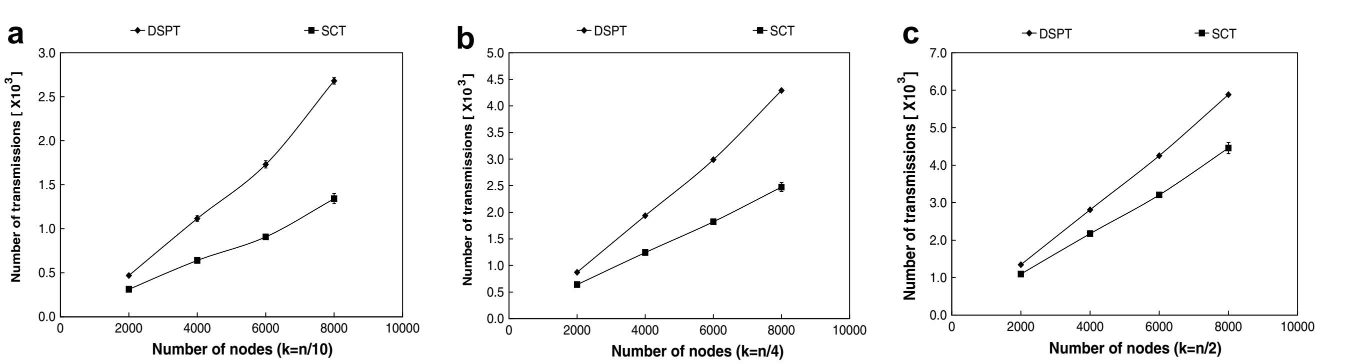

Fig. 6. Performance comparison between SCT, and DSPT: Figure (a), (b), and (c) show the number of transmissions as a function of number of nodes for different source densities k.

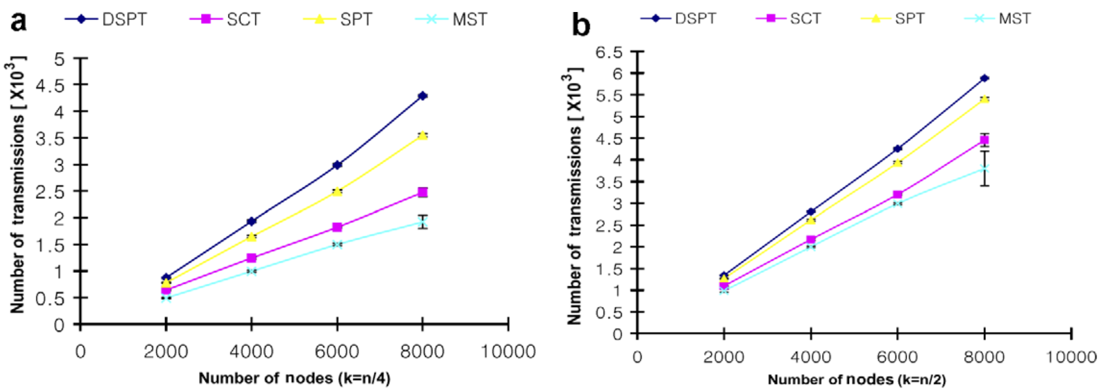

Fig. 7. Performance comparison between SCT and centralized schemes, SPT and MST: Figure (a) and (b) show the number of transmissions as a function of number of nodes for different number of sources k.

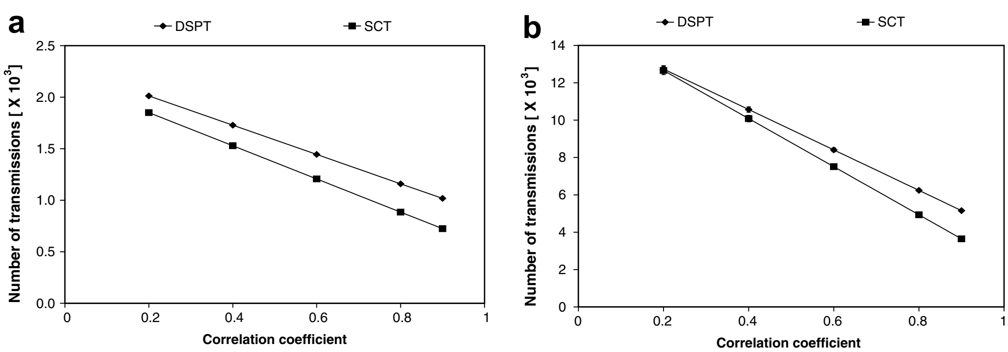

Fig. 8. Performance comparison between SCT and DSPT in case of different correlation factor ρ.

leverage the correlation. However, the message cost of SCT reduces faster than DSPT because it facilitates aggregation at an earlier stage of packet forwarding, hence can reduce packet transmission cost more effectively. Since DSPT is the optimal structure when $\rho = 0$, we expect that the two curves will cross each other at a certain small correlation factor. For $n = 8000$, the cross point occurs at $\rho = 0.2$, while for $n = 2000$, the cross point is expected to occur at $\rho < 0.2$. As we analyzed earlier, DSPT aggregates less efficiently when n becomes large, hence it does not lose much when ρ decrease, therefore it performs relatively better for a smaller ρ when n is large.

7.4. Delay sensitivity

To evaluate a data gathering scheme, latency is another important metric besides the message cost. In this part, we study the data gathering latency of SCT and at the same time characterize the message cost-latency trade off for SPT.

Since SCT is an event-driven approach, none of the aggregation nodes on the tree needs a timer, and each aggregation node can forward messages as soon as it receives responses from downstream aggregation nodes. Therefore, there is very little overhead for perfect aggregation. While for SPT, in order to achieve perfect aggregation, each node on the tree has to set a timer and wait for a certain amount of time before aggregation.

Since SPT itself does not include any mechanism for aggregation timing, we implement a simple scheme [11] that sets aggregation timer for each node based on its hop distance to the sink. Each node on the tree knows its distance (in hop count) as well as the maximum distance D among all nodes on the tree to the sink (depth of the tree). Given a maximum delay MAX_DELAY, per-hop delay Δ is calculated as MAX_DELAY/D. A node that is i hops away from the sink will wait for $MAX\_DELAY\text{-}i * \Delta$ time before aggregating and forwarding data it has received. In this way, if Δ is large enough to accommodate transmission and contention delay at each hop, perfect aggregation can be achieved.

Fig. 9 shows the number of transmissions as a function of maximum delay MAX_DELAY for $n = 2000$ and $n = 4000$ cases. It is shown that SPT achieves perfect aggregation when MAX_DELAY is more than 8.0 s. However, the cost of SPT increases as MAX_DELAY approaches 5.0 s since smaller MAX_DELAY increases the possibility of late arrivals of data before aggregation. Once a packet misses its aggregation deadline at one of the intermediate hops, the probability of it missing deadlines at later hops becomes higher, which explains why SPT performance aggravates quickly as MAX_DELAY decreases. On the other hand, since SCT is an event driven data aggregation structure, its message cost remains the same irrespective of different delays, which explains the flat curves in both figures.

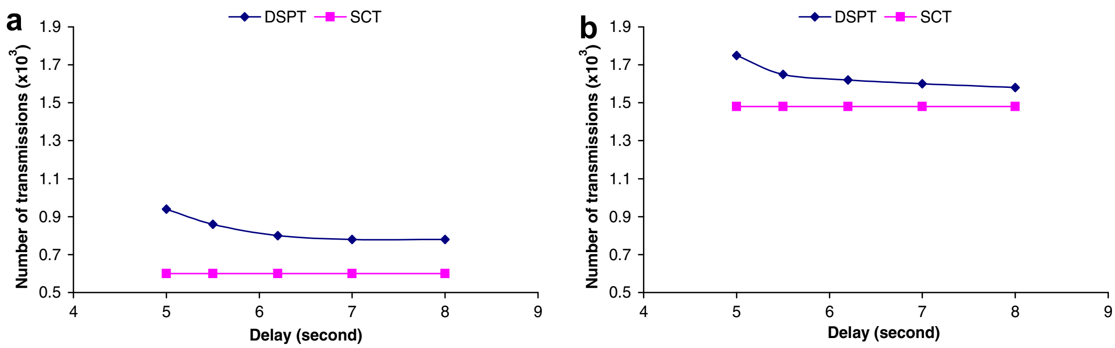

Fig. 9. Performance comparison between SCT and SPT in case of different delays.

7.5. Localized events

Until now, we have assumed that sources are deployed independently in a sensor field. This is typical when considering semantic correlation, where each source reports information about a discrete event. However, spatial correlation usually generates non-uniformly distributed sources, where multiple sources are reporting the same event. In this case, sources are clustered around one or more spots in the network. In this section, we investigate how SCT performs when sources are distributed in such a fashion.

In the simulation, we consider $n = 2000$ and $n = 8000$ scenarios, each has a total of $k = \frac{n}{4}$ sources in the network. The number of events (spots) in the network is varied from $\frac{k}{100}, \frac{k}{50} \cdots$ to $\frac{k}{5}$. With this configuration, the total number of sources does not change, but the distribution varies from a highly concentrated scenario to close to uniform distribution.

Fig. 10(a) and (b) show the simulation results for such spot topologies. When there are very few events and source distributions are highly concentrated, DSPT achieves good aggregation at early stage and aggregated messages reach the sink along only a few paths. But SCT also performs well because SCT does explicit aggregation within each sector that contains sources, while other sectors without sources or not on the aggregation paths to the sink do not take part in the aggregation process, therefore does not incur any extra cost. From the figure, we can see that at $n = 2000$, when the number of events is small, DSPT and SCT have similar cost. But as the number of events increases, sources are less concentrated as in the previous case, and the probability of DSPT paths overlapping reduces, hence the higher message cost. However, for SCT the only extra cost is the introduction of more aggregation nodes into the aggregation tree, which is limited by the total number of sectors in the network. Therefore, the increase of message cost is much less compared with DSPT. For $n = 8000$ case we observe a similar trend, and even for the least event number, SCT outperforms DSPT, because as we analyzed earlier, the chance of DSPT paths overlapping decreases as node density increases.

8. Issues and discussions

8.1. Empty sectors

In Section 6, we assumed that each sector contains at least one node which can serve as an aggregation point and ensure that the tree is connected. However, due to the irregularity of Poisson node distribution, it is possible that some of the sectors may not contain any node. Here, we investigate the implications of having an empty sector. In this case, downstream aggregation nodes of the empty sector won't find a path to aggregate, and hence will not be able to forward the aggregated information. The design of SCT has a simple back up strategy in this case: if an aggregation node is not able to find an aggregation node upstream, it will simply forward the aggregation message

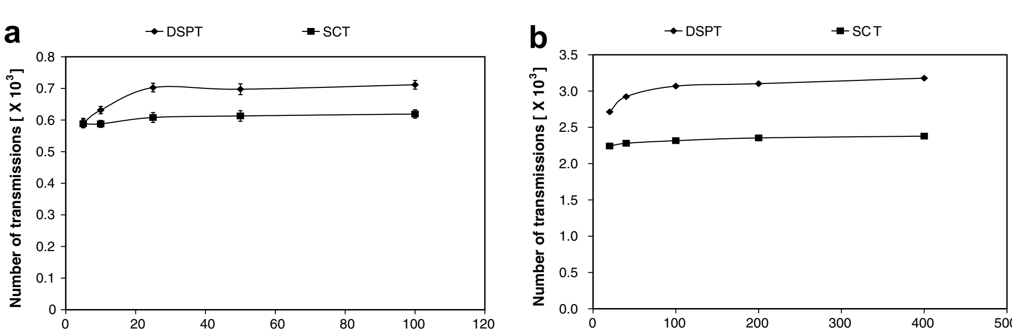

Fig. 10. Performance comparison between SCT and DSPT for Different numbers of localized events in case of 2000 and 8000 nodes, respectively.

to a neighbor in the adjacent sector. After that, the aggregated message will be delivered via a connected aggregation path.

Although this is a less efficient aggregation path, we argue that the probability of empty sectors is extremely low for the target environment.

Consider a network with n nodes and S total number of sectors of the same size, the probability that no empty sector exists in the network is:

$$P = \frac{S^n - \sum_{i=1}^{S-1} \binom{s}{i} (S-i)^n (-1)^{i+1}}{S^n} = 1 - \sum_{i=1}^{S-1} \binom{s}{i} \left(1 - \frac{i}{S}\right)^n (-1)^{i+1}$$

where

$$S = \sum_{i=1}^{m} s(i) = \sum_{i=1}^{m} \left\lceil \frac{(2i-1)n}{mn_0} \right\rceil \quad (10)$$

For the n_0^* and m^* we chose for the fixed structure, the probabilities of non-empty sectors for various node densities are shown in Table 2.

8.2. Comparison with clustering approaches

Structure-wise, SCT approach bears a few similarities with clustering algorithms proposed for ad-hoc and sensor networks [12,13]. In a clustering approach, the entire network is partitioned into clusters of similar size, with a head node elected in each cluster. Non-cluster nodes report messages via a short path to their corresponding cluster heads, where messages are aggregated and transmitted to the sink, and the total transmission cost is reduced through the decreased total transmission distance and smaller message size. With this analogy, a natural question to ask is: *How is SCT different from these clustering algorithms?*

In terms of cluster construction, there are two different solutions from existing works. One approach is for each node to broadcast in a certain area its properties (id, node degree, residual energy etc.), following which an election process is executed to choose the cluster head [13,14]. This approach generally assures regular cluster size and full node coverage, but at the cost of high communication overhead. Using the approach proposed in [13], the clustering algorithm is triggered at periodic intervals to select new cluster heads. At each interval, the clustering process requires a certain number of iterations to finally select the desired cluster head. If the minimum probability of a node becoming a cluster head is p, it takes $N \leqslant \lceil \log_2 \frac{1}{p} \rceil + 1$ steps for the election algorithm to terminate ($N = 6 - 15$ iterations for average scenarios). During each iteration, a tentative cluster head generates broadcasting messages, resulting in a total message cost (for setting up the cluster structure) to be of order $N*n$, where n is the total number of nodes in the network. This translates to significant energy consumption, given that the election process is invoked repeatedly to achieve load balancing. In contrast SCT does not require the overhead of message exchange in either the initial set up phase or in the later maintenance phase, where structure modifications are performed.

Another approach, such as the one used in [12], is to specify a certain probability for each node to become a cluster head, and the node which turns out to be cluster head announces itself through a limited-scope flooding. This approach incurs lower message overhead, but cannot guarantee a uniform cluster head distribution and full coverage of all non-cluster nodes. Therefore, the *orphan* nodes not covered by any cluster heads have to transmit their messages directly to the sink, significantly increasing the total message cost. On the contrary, in SCT, aggregation nodes are chosen implicitly without any message exchange, since the location of the node is the criteria for head selection. Therefore, in terms of clustering forming overhead, it is comparable to the probability-based approach. On the other hand, the clustering structure formed in SCT is organized such that each node is guaranteed to be covered by an aggregation node, and every cluster has similar size. In this sense, a low message cost is achieved without incurring any overhead for additional message exchanges.

In terms of cluster manageability, SCT benefits from its highly regular structure. Since cluster heads consume more energy in terms of communication and computation, cluster head rotation is a essential part of any clustering scheme. For most of the clustering schemes, rotation involves re-election of cluster heads, which incurs a lot of overhead. While with SCT, since the structure is symmetric along all radii, a simple rotation of the SCT rings and sectors moves the roles of cluster heads to a different set of nodes in the network. With carefully designed rotation sequence, the role of cluster heads can be evenly distributed to all sensors in the network.

Table 2
The probability of non-empty sectors

Node density	Probability (%)
$n = 2000$	99.97
$n = 4000$	99.99
$n = 6000$	99.94
$n = 8000$	99.89

8.3. Impact of network shape and sink location

In the previous sections, we assume that the sensor network has a circular shape and the sink is located at the center of the network. These assumptions are made for ease of discussion and presentation of SCT. The approach itself does not impose any constraint on the shape of the network and the location of the sink. For example, consider a network with rectangular shape and the sink located at one of the four corners of the rectangle. In this case, the SCT structure can still be constructed by fitting the entire network into a polar coordinate, with sink being the root of the SCT and the diagonal of the rectangle being the radius. Load balancing can be achieved as well by shifting the rings and sectors periodically as described before. Although the efficiency of the SCT structure for irregular network shapes may not be as high as that of circular network shape, the correctness of SCT is not compromised.

8.4. Other related work

In [15], a TTDD approach is proposed to provide scalable and efficient data delivery to multiple mobile sinks in sensor fields. Each data source in TTDD pro-actively builds a grid structure which enables mobile sinks to continuously receive data on the move by flooding queries within a local cell only. This approach is similar to SCT in that they both construct virtual geometric structures (a grid structure in TTDD and a ring-sector structure in SCT) to facilitate data delivery. However, TTDD addresses the problem of communication between sensors and mobile sinks in the sensor field. The focus of TTDD is point-to-point communication in wireless sensor networks, while our work focuses on data collection in many-to-one communication (i.e. data aggregation). Furthermore, the primary challenge addressed in [15] is to maintain connectivity between senders and receivers when one of them is mobile, while we consider mainly static sensor networks and focus on the energy-efficiency of the data aggregation process.

EWSN [16] proposes combining passive clustering scheme with directed diffusion to restrict the flooding of interests and exploratory data at the initial stage of direct diffusion. Passive clustering is proposed in [17] to control the exchange of flooded messages in sensor networks. Instead of constructing a cluster structure pro-actively, this scheme piggybacks all control information on data messages to save the cost required for setting up clusters, at the cost of less up-to-date clustering structures. A simple cluster-head selection approach, where the first sensor nodes declaring itself as cluster-head wins, is used to reduce the overhead of complicated clustering approach. The application of passive clustering to directed diffusion can effectively reduce the flooding overhead of interest and exploratory message propagation. However, it does not address the problem of improving the aggregation efficiency of directed diffusion, which is the problem considered in this work.

In [18], the authors describe an approach to set up connections among a given set of cluster heads to ensure a connected backbone among cluster heads. In this approach, a random distributed algorithm is proposed to provide connectivity with high probability. The focus of this work is ensuring connectivity of cluster heads (aggregation nodes in our context), while our focus is to select a subset of nodes as aggregation nodes to minimize the total message cost of data gathering. Therefore, the contributions of these two works are orthogonal to each other.

9. Conclusions

In this paper, we propose a novel solution to aggregate correlated information from a subset of sensors to the sink. The proposed scheme is scalable, distributed, requires minimal computation and is highly-manageable compared with existing solutions. The proposed scheme is assessed both intuitively and analytically. Through simulations, we compared the proposed scheme with ideal, centralized data structures as well as a distributed structure. Simulation results show that as a correlation-aware structure, SCT performs significantly better than correlation-unaware structures in terms of message cost and data gathering latency.

References

[1] R. Cristescu, B. Beferull-Lozano, M. Vetterli, On network correlated data gathering, in: Proceedings of The 23th Conference on Comuter Communications (INFOCOM'04), Hong Kong, 2004, pp. 2571–2582.

[2] D.R. Karger, M. Minkoff, Building Steiner trees with incomplete global knowledge, in: Proceedings of the 41th Annual IEEE Symposium on Foundations of Computer Science (FOCS'00), Redondo Beach, CA, 2000, pp. 613–623.

[3] A. Gupta, M. Pal, R. Ravi, A. Sinha, Boosted sampling: approximation algorithms for stochastic optimization, in: Proceedings of The 34th Annual ACM Symposium on Theory of Computing (STOC'04), Chicago, IL, 2004, pp. 417–426.

[4] V. Mhatre, K. Rosenberg, Design guidelines for wireless sensor networks: communication, clustering and aggregation, J. Ad Hoc Networks 2 (1) (2004) 45–63.

[5] C. Intanagoniwawat, R. Govindan, D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, in: Proceedings of The 6th Annual International Conference on Mobile Computing and Networking (MOBICOM'00), Boston, MA, 2000, pp. 56–67.

[6] B. Karp, H. Kung, GPSR: Greedy perimeter stateless routing for wireless networks, in: Proceedings of The 6th Annual International Conference on Mobile Computing and Networking (MOBICOM'00), Boston, MA, 2000, pp. 243–254.

[7] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, S. Shenker, GHT – A geographic hash-table for datacentric storage, in: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and their Applications (WSNA'02), Atlanta, GA, 2002, pp. 78–87.

[8] R. Cristescu, M. Vetterli, Power efficient gathering of correlated data: optimization, NP-completeness and heuristics, in Poster Session: The 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'03), Annapolis, MD, 2003.

[9] H. Takahashi, A. Matsuyama, An approximate solution for the Steiner tree problem in graphs, Math. Japonica 6 (24) (1980) 573–577.

[10] A. Goel, D. Estrin, Simultaneous optimization for concave cost: single sink aggregation or single source buy-at-bulk, in: Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithm (SODA'03), Baltimore, MD, 2003, pp. 499–505.

[11] W. Yuan, S.V. Krishnamurthy, S.K. Tripathi, Synchronization of multiple levels of data fusion in wireless sensor networks, in: Proceedings of the 46th IEEE Global Telecommunicaions Conference (GLOBECOM'03), Sanfrancisco, CA, 2003, pp. 221–225.

[12] S. Bandyopadhyay, E.J. Coyle, An energy efficient hierarchical clustering algorithm for wireless sensor networks, in: Proceedings of

The 22th Conference on Computer Communications (INFOCOM'03), Sanfrancisco, CA, 2003, pp. 1713–1723.

[13] O. Younis, S. Fahmy, HEED: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks, IEEE Trans. Commun. 3 (4) (2004) 366–379.

[14] A.D. Amis, R. Prakash, D. Huynh, T. Vuong, Max-min D-cluster formation in wireless ad hoc networks, in: Proceedings of The 19th Conference on Comuter Communications (INFOCOM'00), Tel-Aviv, Israel, 2000, pp. 32–41.

[15] H.Y. Luo, F. Ye, J. Cheng, S.W. Lu, L.X. Zhang, TTDD – A two-tier data dissemination for large-scale sensor networks, in: Proceedings of The 6th Annual International Conference on Mobile Computing and Networking (MOBICOM), Atlanta, GA, 2002, pp. 148–159.

[16] V. Handziski, A. Köpke, H. Karl, C. Frank, W. Drytkiewicz, Improving the energy efficiency of directed diffusion using passive clustering, in: Proceedings of the 1st European Workshop on Wireless Sensor Networks, Berlin, Germany, 2004, pp. 172–187.

[17] T.J. Kwon, M. Gerla, Efficient flooding with passive clustering in ad hoc networks, ACM SIGCOMM Comput. Commun. Rev. 32 (1) (2002) 44–56.

[18] I. Chlamatac, A. Farago, A new approach to the design and analysis of peer-to-peer mobile networks, Wirel. Network 5 (1) (1999) 149–156.