Gathering Correlated Data in Sensor Networks

Pascal von Rickenbach
Computer Engineering and
Networks Laboratory
ETH Zurich, Switzerland
pascalv@tik.ee.ethz.ch

Roger Wattenhofer
Computer Engineering and
Networks Laboratory
ETH Zurich, Switzerland
wattenhofer@tik.ee.ethz.ch

ABSTRACT

In this paper, we consider energy-efficient gathering of correlated data in sensor networks. We focus on *single-input coding* strategies in order to aggregate correlated data. For *foreign coding* we propose the MEGA algorithm which yields a minimum-energy data gathering topology in $O(n^3)$ time. We also consider *self-coding* for which the problem of finding an optimal data gathering tree was recently shown to be NP-complete; with LEGA, we present the first approximation algorithm for this problem with approximation ratio $2(1+\sqrt{2})$ and running time $O(m + n \log n)$.

Categories and Subject Descriptors:
F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems—*computations on discrete structures*;
G.2.2 [Discrete Mathematics]: Graph Theory—*graph algorithms*;
G.2.2 [Discrete Mathematics]: Graph Theory—*network problems*

General Terms: Algorithms, Theory.

Keywords: Sensor networks, data gathering, data aggregation, distributed algorithms, energy efficiency.

1. INTRODUCTION

Recent advances in wireless networking and microelectronics have led to the vision of sensor networks consisting of hundreds or even thousands of cheap wireless nodes—each equipped with some memory, a processor, a power unit, and a short-range radio—covering a wide range of application domains [1, 6, 11]. These networks of the future can be readily deployed in physical environments to collect information from an area of interest in a robust and autonomous manner. Because of the requirement of unattended operation in remote or even hostile environments and due to the nodes being operated by a fugacious battery or a feeble solar cell, energy efficiency is a major concern in sensor networks.

DIALM-POMC'04, October 1, 2004, Philadelphia, Pennsylvania, USA.

We focus on one generic type of applications for sensor networks, namely monitoring, in which all nodes periodically produce relevant information by sensing an extended geographic area that is eventually transmitted to an information sink for processing. Example scenarios that fall in this category include monitoring of continuous environmental conditions such as temperature, humidity, or seismic activity. Because of power and transmission range limitations, multi-hop routing techniques are applied to transmit the data from all nodes to the information sink. Since different sensor nodes partially monitor the same spatial region, data is often correlated. In order to account for this circumstance and to save energy data should be already processed as it flows from the information source to the sink. This technique is commonly referred to as (in-network) data aggregation. Thereby, a sensor node uses a so called aggregation function to encode the data available at that node before forwarding it to the sink. Several coding strategies were proposed in recent research that can be classified as follows. On the one hand there are the so called multi-input coding strategies [14, 27], where aggregation is performed at a node only if all input information from multiple nodes is available in order to exploit correlation among several nodes. On the other hand, there also exist single-input coding strategies [10] where the encoding of a node's information only depends on the information of one other node.

In this paper, we only consider conditional coding where data from one node can be compressed in the presence of data from other nodes. We therefore only refer to conditionally encoded data as encoded data and speak about raw data otherwise even though a node may also apply an encoding scheme to its gathered data in the absence of side information. In particular, we focus on single-input coding strategies because they feature several advantages compared to multi-input coding strategies. The most important one is certainly the ability to apply single-input coding also in asynchronous networks where no timing assumptions can be made. Using multi-input coding on the other hand, certain timing assumptions have to be made since packets cannot be delayed for an indefinite time at intermediate nodes while waiting for belated information [24]—data freshness at the sink should not suffer too much from data aggregation. We distinguish two classes of single-input coding, namely self-coding and foreign coding. Using self-coding, data is only allowed to be encoded at the producing node and only in the presence of side information from at least another node. With foreign coding in contrast, a node is only able to encode raw data originating at another node as it is routed

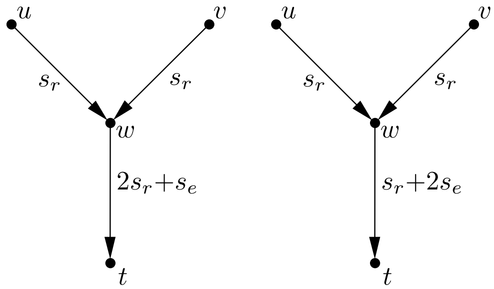

Figure 1: Simple network example. Raw data has size s_r while encoded data is of size s_e, with $s_e < s_r$. On the left hand side self-coding is applied while foreign coding is used on the right. For this example foreign coding is more energy efficient than self-coding.

towards the sink using its own data.

Figure 1 depicts a simple network example where three sensor nodes (u, v, w) want to send their raw data of size s_r to a sink t. Communication links only exist between nodes u and w, v and w, as well as w and t, respectively. Therefore, packets from u and v have to be relayed at w to reach the sink t. If a node is able to encode data due to side information the data size reduces to s_e, with $s_e < s_r$. The configuration on the left depicts the usage of self-coding. Since u and v do not have any side information the both send a packet of s_r bit towards t. Because of side information from u, and v, respectively, node w is able to encode its data such that the corresponding packet has size s_e and therefore $2s_r + s_e$ bits have to be sent over the link (w, t). On the right hand side of Figure 1 the same network is shown if foreign coding is applied. As with self-coding, nodes u and v send their packets of size s_r to w. However, w encodes the raw data of u and v using its own data before it relays it the sink t. Thus, only $s_r + 2s_e$ bits are sent over (w, t). Using $s_e < s_r$, foreign coding transmits less bits over (w, t) than self-coding and is thus more energy efficient for this configuration.

In case of self-coding, [10] shows that already for a very restrictive model where raw data is of size s_r and a node can encode its data to use only s_e bits, with $s_e < s_r$, in the presence of any side information, the problem of finding a minimum-energy data gathering tree is NP-complete. To the best of our knowledge, we are the first that provide an approximation algorithm for this problem with approximation ratio $2(1+\sqrt{2})$. Algorithm LEGA is based on the *shallow light tree* (SLT) introduced in [2, 3, 19] that unifies the properties of the *minimum spanning tree* (MST) and the *shortest path tree* (SPT).

Considering foreign coding, we introduce the algorithm MEGA that results in a minimum-energy data gathering topology under the assumption that the topology of the network and the correlation structure of the nodes are known. MEGA obtains an optimal solution for the data gathering problem by reducing it to the problem of finding a MST in a directed graph which is known to be computable in polynomial time. We then introduce a distributed version of MEGA which works in a slightly more restrictive model.

After discussing related work in the following section, we state the network model for this paper in Section 3. Focusing on foreign coding we present an algorithm computing a minimum-energy data gathering topology in Section 4. In Section 5, dedicated to the self-coding strategy, an approximation algorithm for the optimal data gathering tree is introduced. Section 6 concludes the paper.

2. RELATED WORK

Already at the outset of the design of data gathering protocols for sensor networks researcher have identified the importance of data aggregation in order to improve energy efficiency. In [17] Directed Diffusion is proposed, a protocol in which sensors create gradients of information in their respective neighborhoods. The sink node requests data by broadcasting interests. If interests fit gradients, paths of information flow are formed and in order to reduce communication costs, data is aggregated on the way. The key idea in [15] is to reduce the number of nodes communicating directly with the sink by forming randomized clusters. Each cluster-head encodes data arriving from nodes in its cluster, and sends an aggregated packet to the sink. However, the main drawback of the protocol in [15] is the requirement that all nodes must be able to directly communicate with the sink.

In [14] the problem of data gathering is addressed by using concave, non-decreasing cost functions to model the aggregation function applied at intermediate nodes. The authors propose a hierarchical matching algorithm resulting in a aggregation tree that simultaneously approximates all such cost functions up to a logarithmic factor. However, in their model only the number of nodes providing data to an aggregating node decides on its aggregation performance regardless of the correlation among the available data. That is, the impact of data correlation is not explicitly considered. This too simplistic assumption that an intermediate node can aggregate multiple incoming packets into a single outgoing packet albeit their correlation is also required by other work [18, 21, 22].

Based on signal processing techniques, papers [7, 10] tackle the problem of minimum-energy data gathering by applying Slepian-Wolf coding. In their model the correlation among nodes in known a-priori and is used in order to optimize the rate allocation of a distributed compression algorithm which obviates the need for the sensor nodes to exchange their data among each other in order to strip their common redundancy.

The work which is most related to the problem we consider in this paper is the one involving the concept of self-coding [10]. The authors prove that already for a restricted model where nodes are only allowed to encode their own data in the presence of side information the problem of finding minimum-energy data gathering trees is NP-complete, by applying a reduction to set cover. Moreover, [10] proposes a heuristic based on a combination of a shortest path tree augmented by travelling salesperson paths. We continue their work by establishing a strict classification of coding strategies. Furthermore, we provide an approximation algorithm in case of self-coding and an optimal one for the foreign coding strategy.

3. NETWORK MODEL

Sensor networks are commonly modeled by graphs. A graph $G = (V, E)$ consists of a set of n nodes $V \subset \mathbb{R}^2$ in the euclidian plane and a set of m edges $E \subset V^2$. Nodes in V correspond to sensor nodes, whereas edges represent links between these nodes. In order to prevent already basic communication between directly neighboring nodes from becoming unacceptably cumbersome [23], it is often required that a message sent over a link can be acknowledged by sending a corresponding message over the same link in the opposite direction. In other words, only *undirected* (symmetric) edges are considered. Let $t \in V$ denote a particular node called the sink node where the data from all nodes in V should be gathered. We refer to the process of gathering information on a certain time interval from each node as a round. Therefore, at each round the data from all nodes in V has to be send to t, where it is further processed.

Sensor nodes are considered to be able to adjust their radio signal strength, in order to save energy. Therefore, the weight $w(e)$ for an edge $e = (v_i, v_j) \in E$ is defined to be the cost of transmitting one bit of data from node v_i to node v_j, or vice versa. That is, we use an energy metric for the graph G. We do not consider a specific radio model such as the popular first order radio model presented in [15], since we want our results to be independent from the applied radio model. Therefore, the radio model is abstracted in the edge weights of the graph G.

4. FOREIGN CODING

In this section we first introduce a model for the data correlation in a network. Based on this model an algorithm is presented that solves the minimum-energy data gathering problem. We then propose a distributed version of MEGA which works in a slightly more restrictive model.

4.1 Correlation Model

Since sensor networks are often used to sense real world phenomena, each sensor node continuously produces information as it monitors its vicinity. Thus, we assume that each node $v_i \in V$ generates one data packet p_i of size s_i bits per round that describes the measured information sample losslessly. Note that data packets from different nodes need not have equal size.

Distributed sensor data is often correlated and it is therefore often possible to perform in-network aggregation. Data aggregation can potentially take place at any intermediate node as a data packet is routed towards the sink node. However, once a packet is encoded at a node it is not possible to alter the packet again; hence, recoding is not possible. In other words a node v_i can use its data to encode packets containing correlated data that are routed through v_i on their paths to the sink node t. A packet from node v_i that is encoded at a node v_j is denoted by p_i^j; its corresponding size is s_i^j. The compression rate depends on the data correlation between the involved nodes v_i and v_j, denoted by the correlation coefficient $\rho_{ij} = 1 - s_i^j/s_i$. Encoding at a node v_i only depends on the data collected by v_i and not on other data also routed through v_i—recording is not possible. However, it must be guaranteed that encoding does not result in cyclic dependencies that cannot be resolved while decoding at the sink t. Such an encoding strategy does not depend on timing assumptions in the encoding nodes, and therefore it is also applicable to asynchronous networks.

MEGA

Input: Graph $G = (V, E)$ and sink $s \in V$
T_{SP} = shortest path tree in G rooted at t
$\widetilde{G} = (V, \widetilde{E})$ = complete directed graph
for all $(v_i, v_j) \in \widetilde{E}$ **do**
    $w'(v_i, v_j) = s_i \left(w(v_i, v_j) + w(v_j, t)(1 - \rho_{ij}) \right)$
end for
T' = DMST on $\widetilde{G}^T$
$T = (V, E_T) = T'^T$
for all $v_i \in V$ **do**
    consider v_j such that $(v_i, v_j) \in E_T$
    set v_j as encoding relay for p_i
end for
Output: Minimum-energy data gathering topology for G

Then the *minimum-energy data gathering* problem for a given graph $G = (V, E)$ is defined as follows. Find a routing scheme and a coding scheme to deliver data packets from all nodes in V to a sink node $t \in V$, such that the overall energy consumption is minimal. Let $f(e)$ with $e \in E$ denote the number of bits transmitted over edge e. In order to minimize the overall energy consumption our goal is to minimize the following cost function:

$$C(G) = \sum_{e \in E} w(e) f(e). \tag{1}$$

4.2 Algorithm

In the following we present the *minimum-energy gathering algorithm* (MEGA). The resulting topology of the algorithm is a superposition of two tree constructions. A directed minimum spanning tree of a directed graph whose edges are specified later in this section is used to determine the encoding nodes for all data packets. Once a packet is encoded it is routed on a shortest path towards the sink in order to save energy. Given a graph $G = (V, E)$ as described in Section 3 and a sink node $t \in V$, MEGA first computes a *shortest path tree* (SPT) of G rooted at t (e.g. using Dijkstra's algorithm [12]). Since the weight of an edge in E corresponds to the energy spent to transmit one bit of information from one incident node to the other, the SPT comprises an energy-minimal path from each node in V to the sink.

In a second step, the algorithm computes for each node v_i a corresponding node v_j that encodes the packet p_i using its own data. Since cyclic dependencies in the encoding must not occur in order to guarantee decoding this results in a so-called *coding tree*. In order compute this coding tree, we make use of an algorithm solving the *directed minimum spanning tree* (DMST) problem (also known as the *minimum weight arborescence* problem). Consider a directed graph $G = (V, E)$ with V and E being the set of nodes and edges, respectively, and a weight $w(e)$ associated with each edge e. The problem is to find a rooted directed spanning tree such that the sum of $w(e)$ over all edges e in the tree is minimized provided that all nodes are reachable from the root. Chu and Liu[8], Edmonds [13], and Bock [4] have independently given efficient algorithms for finding the MST on a directed graph. Tarjan [25] gives an efficient implementation (see also [5]). Edmonds algorithm is also described in [20]. Furthermore, a distributed algorithm is given by Humblet [16].

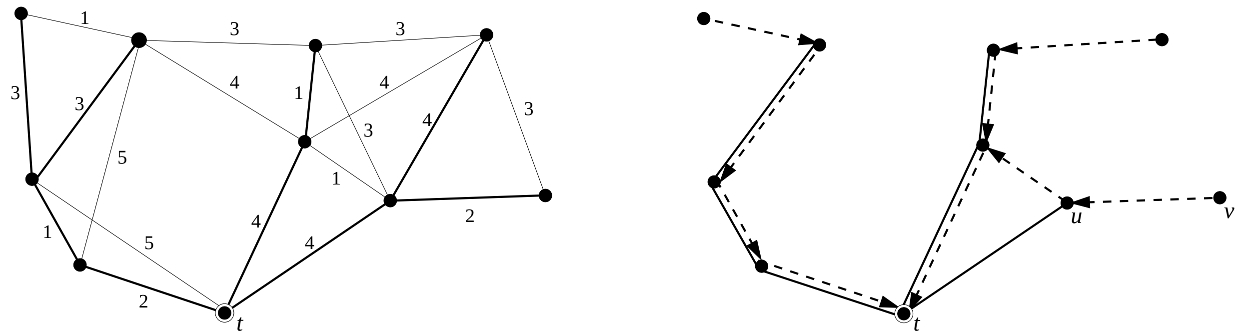

Figure 2: The figure on the left depicts an example graph G where all nodes should send their gathered data to the sink node t. Each edge e in the given graph G is labeled according to the energy consumed to send one bit of data over e. The SPT rooted at t is indicated by bold edges. On the right hand side the resulting minimum-energy data gathering topology obtained by MEGA is shown. The coding tree (dashed arrows) determines for each node its corresponding encoding node. Encoded data is sent on the SPT (solid lines) towards the sink t.

In the following we propose the directed graph on which MEGA computes the DMST with one of the above-mentioned algorithms. First, MEGA builds a complete directed graph $\widetilde{G} = (V, \widetilde{E})$. The weight $\widetilde{w}(e)$ for a directed edge $e = (v_i, v_j)$ in $\widetilde{E}$ is defined as

$$\widetilde{w}(e) = s_i \left(\sigma(v_i, v_j) + \sigma(v_j, t)(1 - \rho_{ij})\right), \qquad (2)$$

whereas $\sigma(v_i, v_j)$ denotes the weight of a shortest path from v_i to v_j in G, that is, the sum of the edge weights on that path. The weight of an edge in $\widetilde{G}$ therefore stands for the total energy consumption in order to route a data packet p_i to the sink using node v_j as an encoding relay. This also depends on the correlation coefficient of the involved nodes. The DMST is by definition a minimum-weight directed tree with edges directed off the root node (e.g. the sink t) but we aim at a directed tree with edges pointing towards t. Therefore, MEGA does not apply a DMST algorithm to $\widetilde{G}$ but to the transposed graph[1] $\widetilde{G}^T$. Then, the edges in $\widetilde{G}$ corresponding to the ones in the DMST of $\widetilde{G}^T$ form a tree that defines the encoding relays for a nodes in V. The resulting topology of MEGA comprises for each node v_i all edges on a shortest path from v_i to its encoding relay v_j found by the above described DMST construction and all edges on the path from v_j to t on the SPT. It can be seen that if the data is pairwise independent for all nodes the resulting topology of MEGA is the SPT—this is the minimum-energy data gathering topology for uncorrelated data.

On the left hand side of Figure 2 an example graph G is depicted. The information from all nodes in G should be gathered at the sink node t. To simplify matters it is assumed that all nodes send a packet of equal size s_r. For each edge e in G the edge weight $w(e)$ corresponding to the energy consumption of sending one bit of data over e is depicted in Figure 2. Data correlation among nodes v_i and v_j is defined to be inverse proportional to their Euclidean distance. That is, $\rho_{ij} = \frac{1}{1+d(v_i,v_j)}$. Furthermore, a path loss exponent of 2 is assumed and thus $d(v_i, v_j) = \sqrt{w(e)}$, with $e = (v_i, v_j)$. Algorithm MEGA first computes a SPT rooted at t—bold lines on the left of Figure 2. The coding tree established by MEGA is depicted with dashed arrows on the right hand side of Figure 2. The encoding relay of each node is thereby determined by its outgoing arrow. Once a packet is decoded it is send towards the sink on the SPT. One can see that different data packets at a node are not always sent to the same neighboring node. For example, at node u the packet received from node v is first encoded and then forwarded directly to the sink t whereas u's packet is sent to an intermediate node for encoding in order to circumvent the costly edge (u, t).

The running time of the algorithm is $\mathrm{O}(n^3)$ since solving the all-pair shortest path problem on G takes $\mathrm{O}(n^3)$ time and the running time for the computation of the DMST on $\widetilde{G}^T$ is $\mathrm{O}(n^2)$.

4.3 Analysis

In order to show MEGA to be optimal we first establish some properties of an optimal data gathering topology. In an optimal solution for the graph G each packet p_i is routed along a distinct path from node v_i to the sink t since multiple paths would unnecessary increase the total energy consumption. The packet p_i is encoded at no more than one node v_j on its path towards the sink. Note that it is also possible that p_i is sent to t without any encoding. In this case t is considered to be the encoding relay for node v_i. Along the lines of MEGA we can therefore establish a directed graph $\widetilde{G}_{opt} = (V, \widetilde{E}_{opt})$ for the optimal solution where each node v_i in V apart form t has exactly one outgoing edge in $\widetilde{E}_{opt}$ pointing towards the encoding relay of the packet p_i. It follows that $|\widetilde{E}_{opt}| = n-1$. In order to guarantee the decoding of all data at the sink node it is required that the encoding does not lead to cyclic dependencies among the encoded packets. By the construction rules of the directed graph $\widetilde{G}_{opt}$ such cyclic dependencies would be reflected in cycles in $\widetilde{G}_{opt}$ and therefore $\widetilde{G}_{opt}$ must be cycle-free. Consequently, the above elaborated properties of the directed graph $\widetilde{G}_{opt}$, namely

- $|\widetilde{E}_{opt}| = n - 1$,

[1]The transpose of a directed graph $G = (V, E)$ is $G^T = (V, E')$ with $(v_i, v_j) \in E'$ if and only if $(v_j, v_i) \in E$. That is, the direction of each edge in G is reversed in the transposed graph.

- every node v_i, $v_i \in V \setminus \{t\}$, has an edge leading out from it,

- and $\widetilde{G}_{opt}$ does not contain cycles,

characterize a directed tree pointing into node t. Thus, we obtain the following theorem:

THEOREM 1. *Given a graph $G = (V, E)$ and a sink $t \in V$, algorithm MEGA computes a minimum-energy data gathering topology for G.*

PROOF. The path for a packet p_i in the optimal solution can be divided into two parts. First, p_i is routed on a path from v_i to its encoding node v_j and in a second step the encoded packet p_i^j is routed from v_j to the sink t. In the optimal topology both sub-paths are minimum-energy paths—And thus shortest paths in G—in order to minimize the overall cost function $C(G)$ as defined in Equation (1). In Equation (1) the total energy consumption is computed by summing up the load of each edge in E times its corresponding weight. Another way to compute the total energy consumption is to charge each node v_i the energy the packet p_i spends during $p_i's$ way to the sink. For each node v_i and its corresponding encoding relay v_j this account is summing up to $\widetilde{w}(v_i, v_j)$ as defined in Equation (2). Consequently, the total energy consumption is defined to be the sum of all edge weights in $\widetilde{G}_{opt}$. MEGA computes exactly this account values for all possible encoding relays of a node v_i and assigns them to the corresponding edges in $\widetilde{G}$. Using a DMST algorithm on the transposed graph a directed tree pointing into t is obtained that minimizes the sum of all edge weights. Since the optimal solution G_{opt} also corresponds to a directed tree in $\widetilde{G}_{opt}$, MEGA also minimizes $C(G)$. □

4.4 Distributed Computation

So far, the proposed centralized algorithm MEGA requires total knowledge about the correlation among all nodes and the topology of the network. In this section we consider the well studied *Unit Disk Graph* (UDG) model [9] where all nodes have the same limited transmission ranges. Additionally, we restrict the raw-data packets of all nodes to have equal size s_r. Data correlation in sensor networks is often assumed to be regional. Thus, in the following the data correlation between two nodes v_i and v_j is modeled to be inverse proportional to their Euclidean distance $d(v_i, v_j)$.

Using the distributed algorithm described in [16] to compute the DMST, MEGA can be implemented in a distributed way. In a setup phase, the sink t starts building a SPT rooted at t using e.g. Dijkstra's algorithm. Thus, each node in V is able to determine the energy consumption to send t one bit of information. Then, each node v_i in the graph G broadcasts a sample packet p_i. Upon receipt of a packet p_j from a neighboring node v_j, v_i encodes p_i using p_j in order to compute the correlation coefficient between the two neighbors. Additionally, v_i can determine the energy cost of a transmission to v_j by using a Received Signal Strength Indicator (RSSI) [26]. Node v_i establishes an directed edge (v_i, v_j) whose weight is set according to Equation (2). The graph $\widetilde{G}$ then consists of edges between direct neighbors in G only. In order to guarantee that MEGA still results in an optimal topology we have to show that in an optimal solution each node and its corresponding encoding relay are only one hop away from each other. That is, they are neighbors in the graph G.

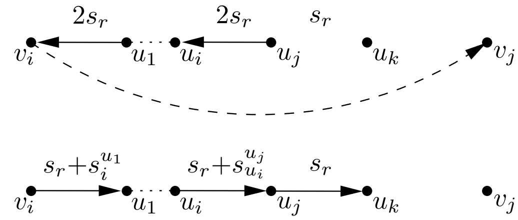

Figure 3: Configuration of the graph $\widetilde{G}$ if the encoding relay v_j of node v_i is more than one hop away from v_i (top) and a configuration using only one-hop relays that results in less energy consumption (bottom).

Assume for the sake of contradiction that the encoding relay v_j for a node v_i in a minimum-energy data gathering topology G_{opt} is more than one hop away from v_i. Then the packet p_i is routed along a path $p(v_i, v_j) = (v_i, u_1 \dots u_i, u_j, u_k \dots v_j)$ to its encoding relay v_j. Since v_i and v_j are no neighbors in G and G is a UDG, it follows that $d(v_i, v_j) > d(v_i, u_1)$ and consequently $\rho_{v_i,v_j} < \rho_{v_i,u_1}$—data correlation is inverse proportional to the Euclidean distance. It follows that $s_i^j > s_i^{u_1}$. Thus if (u_1, v_i) is not in $\widetilde{G}_{opt}$, that is v_i is not the encoding relay for u_1, we choose u_1 to be the encoding node of v_i and obtain a topology with less energy dissipation which contradicts the assumption. If (u_1, v_i) is in $\widetilde{G}_{opt}$ we are in a configuration as it is depicted in Figure 3 at the top. Node v_i has an edge to v_j in $\widetilde{G}_{opt}$ (dashed arrow) and each node on $p(v_i, v_j)$ up to u_j has an edge to its predecessor on the path. Since $\widetilde{G}_{opt}$ is cycle-free, there is at least one node on the path (u_k in Figure 3) that does not point to its predecessor. In Figure 3 all edges on the path are labeled according to their load caused by all raw data packets from nodes on the path. However, by changing the edges in $\widetilde{G}_{opt}$ subject to the configuration at the bottom of Figure 3—and thus also the encoding relays— and due to the assumption that all raw-data packets have equal size s_r edge (u_i, u_j) has a load of at most $s_r + s_{u_i}^{u_j}$ since the packet p_{u_i} is sent to u_j and the encoded packet $p_{u_i}^{u_j}$ possibly back to u_i on its way to the sink. The same holds for all other edges on the path for which the corresponding edge in $\widetilde{G}_{opt}$ is reversed. Since $s_{u_i}^{u_j} < s_r$ for all direct neighbors in G, it follows that we can decrease the energy consumption of G_{opt} by applying the transformation shown in Figure 3 which leads to a contradiction. It is therefore adequate to restrict the directed graph $\widetilde{G}$ to comprise only edges connecting neighboring nodes in G in order to obtain an optimal topology. This consequently allows for the distributed computation of the minimum-energy data gathering topology of G.

5. SELF-CODING

In this section we first determine a lower bound for the energy consumption of an optimal data gathering topology using self-coding. Then, an algorithm is presented that approximates an optimal topology up to a constant factor.

5.1 Correlation Model

In this section we consider the problem of constructing efficient data gathering trees in the model of explicit communication introduced in [10]. In this model nodes can only encode their own raw data in the presence of other raw data routed through them in contrast to the model introduced in Section 4 where the inverse restriction is assumed. Thus, the reduction in data size at a node v_i is due to the direct availability of side information locally at v_i. If no side information is available at node v_i the packet size of p_i is s_r bits. However, if raw data is routed on their way to the sink t through v_i, the node can encode its data such that the size of p_i reduces to s_e bits with $s_e < s_r$. That is, different from the correlation model in Section 4 the correlation between data is equal and therefore $\rho_{ij} = 1 - s_e/s_r$ for all $v_i, v_j \in V$ with $i \neq j$. Consequently, if a node encodes its data using some other data, the encoded data will have exactly s_e bits. It is shown in [10] that already for this restricted correlation model the problem of finding minimum-energy data gathering trees is NP-complete, by applying a reduction to set cover. Moreover, [10] proposes a heuristic based on a combination of a shortest path tree augmented by travelling salesperson paths.

5.2 Lower Bound

Given a graph $G = (V, E)$ and a sink node $t \in V$, we present an approximation algorithm that guarantees a data gathering tree for which the cost $C(G)$ defined in Equation (1) is only a constant factor higher than the cost of an optimal topology. We first give a lower bound on the cost, that is, the energy consumption of the optimal topology.

LEMMA 2 (LOWER BOUND). *The cost of an optimal topology c_{opt} is bounded from below by $c_{opt} \geq max(s_e \cdot c_{SSP}, s_r \cdot c_{MST})$, where c_{SSP} is the sum of the costs of all the shortest paths to the sink t, and c_{MST} is the cost of the minimum spanning tree of all nodes in V.*

PROOF. Nodes in the graph can either send their raw data directly to the sink, or use the raw data of other nodes to encode their data, and then send their data to the sink. Let B be the set of nodes sending their raw data to the sink t without encoding. Let the nodes who encode their data using the raw data of node $u \in B$ be the set S_u. The set B and the sets S_u for all $u \in B$ form a partition over all nodes in V, that is: $V = B \cup \sum_{u \in B} S_u$.

After deciding how V will be partitioned, the optimal algorithm will use the shortest paths (SP) to deliver the encoded data of all nodes in $V \setminus B$ to the sink since this minimizes the total energy consumption. Therefore, nodes in S_u need to encode their data using the raw data of node u, u being a node in set B. On the other hand, the sink t needs to decode the encoded data of nodes in S_u; to do so, t also requires the raw data of u. The optimal topology to distribute the raw data of u is given by the Steiner tree (ST) where the nodes in S_u, node u, and t are terminal nodes. Summing up, the cost of the optimal topology is therefore

$$c_{opt} = \sum_{u \in B} \left(s_r \cdot \mathrm{ST}(S_u, u, t) + \sum_{v \in S_u} s_e \cdot \mathrm{SP}(v, t) \right).$$

We can lower-bound this equation in two ways. By definition $\mathrm{SP}(v_i, v_j) = \mathrm{ST}(v_i, v_j)$ and any additional terminal node in the Steiner tree increases the cost of the tree. Furthermore, since $s_e < s_r$ it follows that

$$\begin{aligned} c_{opt} &= \sum_{u \in B} \left(s_r \cdot \mathrm{ST}(S_u, u, t) + \sum_{v \in S_u} s_e \cdot \mathrm{SP}(v, t) \right) \\ &\geq \sum_{u \in B} s_e \cdot \mathrm{SP}(u, t) + \sum_{u \in B} \sum_{v \in S_u} s_e \cdot \mathrm{SP}(v, t) \\ &= \sum_{u \in V} s_e \cdot \mathrm{SP}(u, t) = s_e \cdot c_{SSP}. \end{aligned}$$

On the other hand, all nodes in B send and all nodes in $V \setminus B$ receive at least one packet containing raw data. Thus, raw data is distributed at least on a spanning tree. Since the minimum spanning tree (MST) is the cheapest possible spanning tree, the cost of the optimal algorithm is also bounded from below by the cost of the MST, used to transmit raw data. The lemma follows immediately. □

5.3 Algorithm and Analysis

In the following we present the *low energy gathering algorithm* (LEGA), an approximation algorithm that is optimal up to a constant factor. The algorithm is based on the *shallow light tree* (SLT), a spanning tree that approximates both the MST and the SPT for a given node (e.g. the sink). The SLT was introduced in [2, 3]. Given a graph $G = (V, E)$ and a positive number γ, the SLT has two properties:

- Its total cost is at most $1 + \sqrt{2}\gamma$ times the cost of the MST of the graph G;
- The distance on the SLT between any node in V and the sink is at most $1 + \sqrt{2}/\gamma$ times the shortest path from that node to the sink.

For more details on the construction of the shallow light tree (SLT) we refer to [19].

The algorithm is as follows: First the SLT spanning tree is computed, the sink node t being the root of the SLT. Then, t broadcasts its packet p_t to all its one-hop neighbor nodes in the SLT. When node v_i is receiving a packet p_j consisting of raw data of a neighboring node v_j, v_i encodes its locally measured data p_i using p_j, and transmits the packet p_i^j to the sink t on the path given by the SLT. Then node v_i broadcasts its packet p_i to all its one-hop neighbors but v_j; in other words to all its children but its parent in the SLT. The sink t has its own data p_t available locally (or it can use the data of one of its first-hop neighbors), and thus can perform recursive decoding of the gathered data, based on the encoded data that it receives from all other nodes in V.

THEOREM 3. *LEGA achieves a $2(1+\sqrt{2})$-approximation of the optimal data gathering topology.*

PROOF. The total cost of LEGA is given by

$$c_{LEGA} = s_r \cdot c_{SLT} + \sum_{v_i \in V} s_e \cdot |path_{SLT}(v_i, t)|.$$

The first term follows from the fact that each node sends its raw data to all its children in the SLT. The second term corresponds to the sum of the shortest paths from all nodes in V to the sink node t in the SLT. Using the SLT properties and setting $\gamma = 1$ we obtain

$$\begin{aligned} c_{LEGA} &= s_r \cdot (1 + \sqrt{2}) c_{MST} + s_e \cdot (1 + \sqrt{2}) c_{SSP} \\ &\leq 2(1 + \sqrt{2}) c_{opt}. \end{aligned}$$

The second equation follows directly from Lemma 2. □

Since [19] provides an algorithm constructing the SLT of a graph G that runs in $O(m + n \log n)$ time, the running time of LEGA is also $O(m + n \log n)$.

6. CONCLUSION

In this paper we investigate the problem of gathering correlated data in sensor networks. In contrast to most of the related work we provide algorithms for two important coding strategies that explicitly consider data correlation. In case of self-coding, for which the problem is known to be NP-complete, we present LEGA, a $2(1+\sqrt{2})$-approximation algorithm. In addition, for the foreign coding strategy, algorithm MEGA is proposed that results in an minimum-energy data gathering topology.

The two considered coding strategies—self-coding and foreign coding—complement each other. It is intriguing that one is optimally solvable in polynomial time while the other is in NP. This leads to the fascinating question whether the general problem (allowing both self-coding and foreign coding at the same time) will be in P or NP.

7. REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002.

[2] B. Awerbuch, A. Baratz, and D. Peleg. Cost-Sensitive Analysis of Communication Protocols. In *Proc. of the* 9^{th} *Symposium on Principles of Distributed Computing (PODC)*, 1990.

[3] K. Bharath-Kumar and J. M. Jaffe. Routing to Multiple Destinations in Computer Networks. *IEEE Trans. on Communications*, 31(3):343–351, 1983.

[4] F. Bock. An algorithm to construct a minimum spanning tree in a directed network. In *Developments in Operations Research*, pages 29–44, 1971.

[5] P. M. Camerini. A note on finding optimum branchings. *Networks*, 9:309–319, 1979.

[6] C.-Y. Chong and S. P. Kumar. Sensor Networks: Evolution, Opportunities, and Challenges. *Proceedings of the IEEE*, 91(8):1247–1256, 2003.

[7] J. Chou, D. Petrovic, and K. Ramchandran. A Distributed and Adaptive Signal Processing Approach to Reducing Energy Consumption in Sensor Networks. In *Proc. of the* 22^{nd} *Annual Joint Conference of the IEEE Computer and Comunications Societies (INFOCOM)*, 2003.

[8] Y. J. Chu and T. H. Liu. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400, 1965.

[9] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit Disk Graphs. *Discrete Mathematics*, 86:165–177, 1990.

[10] R. Cristescu, B. Beferull-Lonzano, and M. Vetterli. On Network Correlated Data Gathering. In *Proc. of the* 23^{rd} *Conference of the IEEE Comunications Society (INFOCOM)*, 2004.

[11] D. Estrin et al. *Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers*. The National Academies Press, Washington DC, 2001.

[12] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[13] J. Edmonds. Optimum branchings. *J. Research of the National Bureau of Standards*, 71b:233–240, 1967.

[14] A. Goel and D. Estrin. Simultaneous Optimization for Concave Costs: Single Sink Aggregation or Single Source Buy-at-Bulk. In *Proc. of the* 14^{th} *Symposium on Discrete Algorithms (SODA)*, 2003.

[15] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Proc. of the* 33^{rd} *Hawaii International Conference on System Sciences (HICSS)*, 2000.

[16] P. A. Humblet. A distributed algorithm for minimum weighted directed spanning trees. *IEEE Trans. on Communications*, 31(6):756–762, 1983.

[17] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proc. of the* 6^{th} *Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2000.

[18] K. Kalpakis, K. Dasgupta, and P. Namjoshi. Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks. In *Proc. of the IEEE International Conference on Networking (ICN)*, 2002.

[19] S. Khuller, B. Raghavachari, and N. Young. Balancing Minimum Spanning and Shortest Path Trees. In *Proc. of the* 4^{th} *Symposium on Discrete Algorithms (SODA)*, 1993.

[20] E. Lawler. *Combinatorial Optimization: Networks and Matroids*. Saunders College Publishing, Fort Worth, 1976.

[21] S. Lindsey, C. Raghavendra, and K. Sivalingam. Data Gathering in Sensor Networks using the Energy*Delay Metric. In *Proc. of the* 15^{th} *International Parallel & Distributed Processing Symposium (IPDPS)*, 2001.

[22] S. Lindsey and C. S. Raghavendra. PEGASIS: Power-Efficient Gathering in Sensor Information Systems. In *Proc. of the IEEE Aerospace Conference*, 2002.

[23] R. Prakash. Undirectional Links Prove Costly in Wireless Ad-Hoc Networks. In *Proc. of the* 3^{rd} *International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, 1999.

[24] I. Solis and K. Obraczka. The Impact of Timing in Data Aggregation for Sensor Networks. In *Proc. of the IEEE International Conference on Communications (ICC)*, 2004.

[25] R. E. Tarjan. Finding Optimum Branchings. *Networks*, 7:25–35, 1977.

[26] Theodore S. Rappaport. *Wireless Communications: Principles & Practice*. Prentice Hall, Upper Saddle River NJ, 2nd edition, 2001.

[27] Y. Yu, B. Krishnamachari, and V. Prasanna. Energy-Latency Tradeoffs for Data Gathering in Wireless Sensor Networks. In *Proc. of the* 23^{rd} *Conference of the IEEE Comunications Society (INFOCOM)*, 2004.