Fast Data Retrieval Scheduling in Tree Based Wireless Sensor Networks

B.Lakshmi Prasanna*, T.Sunitha**

*(M.Tech Final Year student, Department of CSE, QIS College of Engineering and Technology, Ongole)

**(Assoc. Professor, Department of C.S.E, QIS College of Engineering and Technology, Ongole).

Abstract - We address the problem of efficiently gathering correlated data from a wireless sensor network. In a multi-hop wireless network, a conventional way of defining interference neighbors is to prohibit a node from using the same slot/code as those of its 1-hop and 2-hop neighbors. However, for data collection in a wireless sensor network, since the set of communication nodes is limited and the transmission directions are toward the sink, we show that a less strict set of interference neighbors can be defined. Based on this observation, we develop an efficient distributed wake-up scheduling scheme for data collection in a sensor network that achieves both energy conservation and low reporting latency.

I. Introduction

Data collection from a set of sensors to a common sink over a tree-based routing topology is a fundamental traffic pattern in wireless sensor networks (WSNs). This many-to-one communication paradigm in which data flows from many nodes to a single node is known as converge cast. Collecting sensing data is an important function of a wireless sensor network (WSN). It involves a subset of nodes, \( R \), each requested to report its sensory data via a data collection tree to the sink. Two main technical issues are power saving and latency. The former is to prolong network lifetime, while the latter concerns the freshness of data. To simultaneously address these two issues, we define a wake-up scheduling problem, where nodes can periodically switch between sleep and active modes. A node, if involved in the data collection tree, will receive an active slot. During its active slot, a node must wake up to collect data from its children. Then, it can go to sleep. Note that a node also needs to wake up to cooperate with its parent’s active slot. On the other hand, for latency concern, data forwarding along the tree should be bounded. Besides, interference among these transmission activities should be avoided. To avoid interference, a conventional way is to avoid a node from using the same slot as those used by its neighbors within two hops. However, in the data collection scenario in WSNs, since communication only involves partial nodes and the communication directions are always toward the sink along a data collection tree, the definition of interference can be relaxed. Motivated by this observation, this paper shows how to define the tightest set of interference neighbors when assigning active slots to nodes. Based on this definition, we then design an efficient distributed wake-up scheduling scheme for data collection in a WSN to meet the interference-free and low-latency requirements. Several efforts have focused on data collection in a WSN. In \([1][2]\), nodes of the same depth in the data collection tree will have the same wake-up time. Work \([1]\) proposes a staggered wake-up scheme, while \([2]\) extends \([1]\) to a multiparent scheme such that a node can choose one parent with the earliest wake-up time to relay data. Unfortunately, \([1][2]\) are not compatible with ZigBee and nodes of the same depth may suffer from interference. The work \([3]\) proposes a ZigBeecompatible scheduling for convergecast, but it involves all nodes to report their data. It is a special case of our work and still adopts the conventional interference definition. Based on a TDMA model, \([4]\) shows how to assign transmission slots to nodes to avoid interference. Reference \([5]\) further improves \([4]\) by reducing the latency when collecting data along the tree. Although a less strict definition of interference is used in \([4]\) and \([5]\), interference actually happens at the receivers’ side. In this paper, we analyze the data collection problem under the restriction that any data collection protocol can directly utilize only second-order marginal or conditional
probability distributions – in other words, we only directly utilize pairwise correlations between the sensor nodes. There are several reasons for studying this problem. First off, the entropy function typically obeys a strong diminishing returns property in that, utilizing higher-order distributions may not yield significant benefits over using only second-order distributions. Second, learning, and utilizing, second-order distributions is much easier than learning higher-order distributions (which can typically require very high volumes of training data). Finally, we can theoretically analyze the problem of finding the optimal data collection scheme under this restriction, and we are able to develop polynomial-time approximation algorithms for solving it. The above restriction leads to what we call compression trees. Generally speaking, a compression tree is simply a directed spanning tree T of the communication network, in which, the parents are used to compress the values of the children. More specifically, given a directed edge (u, v) in T , the value of Xv is compressed using the value of Xu (i.e., we use the value of Xu = xu to compute the conditional distribution p(Xv|Xu = xu) and use this distribution to compress the observed value of Xv (using say Huffman coding)). The compression tree also specifies a data movement scheme, specifying where (at which sensor node) and how the values of Xu and Xv are collected for compression. The compression tree-based approach can be seen as a special case of the approach presented by one of the authors in prior work [11]. There the authors proposed using decomposable models for data collection in wireless sensor networks, of which compression trees can be seen as a special case. However, that work only presented heuristics for solving the problem, and did not present any rigorous analysis or approximation guarantees.

II. Related Work

Presenting preliminary background on data compression in sensor networks, discuss the prior approaches, and then introduce the compression tree-based approach. A. Notation and Preliminaries We are given a sensor network modeled as an undirected, edge-weighted graph GC(V = {1, · · · , n},E), comprising of n nodes that are continuously monitoring a set of distributed In the rest of the paper, we denote this by Xv|Xu attributes X = {X1, · · · ,Xn}. The edge set E consists of pairs of vertices that are within communication radius of each other, with the edge weights denoting the communication costs. Each attribute, Xi, observed by node i, may be an environmental property being sensed by the node (e.g., temperature), or it may be the result of an operation on the sensed values (e.g., in an anomaly-detection application, the sensor node may continuously evaluate a filter such as “temp > 100” on the observed values). If the sensed attributes are continuous, we assume that an error threshold of e is provided and the readings are binned into intervals of size 2e to discretize them. In this paper, we focus on optimal exploitation of spatial correlations at any given time t; our approach can be generalized to handle temporal correlations in a straightforward manner. We are also provided with the entropy rate for each attribute, H(Xi) and the conditional entropy rates, over all pairs of attributes. More generally, we may be provided with a joint probability distribution, p(X1, ...,Xn), over the attributes, using which we can compute the joint entropy rate for any subset of attributes. However accurate computation of such joint entropies for large subsets of attributes is usually not feasible. We denote the set of neighbors of the node i by N(i) and let N(i) = N(i)||i|| and deg(i) = |N(i)|. We denote by d(i, j) the energy cost of communicating one bit of information along the shortest path between i and j. We focus on the wireless communication model (WL) in this paper; specifically we assume that when a node transmits a message, all its neighbors can hear the message (broadcast model). We further assume that the energy cost of receiving such a broadcast message is negligible, and we only count the cost of transmitting the message. In the extended version of the paper [12], we discuss how our approach generalizes to wired communication networks, and to unicast or multicast models. Prior Approaches Given the entropy and the joint entropy rates for compressing the sensor network attributes, the key issue with using them for data compression is that the values are
generated in a distributed fashion. The naive approach to using all the correlations in the data is (a) to gather the sensed values at a central sensor node, and (b) compress them jointly. However, even if the compression itself was computationally feasible, the data gathering cost would typically dwarf any advantages gained by doing joint compression. Prior research in this area has suggested several approaches that utilize a subset of correlations instead. Several of these approaches are illustrated. As discussed in the introduction, in practice, we are likely to be limited to using only low-order marginal or conditional probability distributions for compression in sensor networks. In this paper, we begin a formal analysis of such algorithms by analyzing the problem of optimally exploiting the spatial correlations under the restriction that we can only use second-order conditional distributions (i.e., two-variable probability distributions). A feasible solution under this restriction is fully specified by a directed spanning tree \( T \) rooted at \( r \) (called a compression tree) and a data movement scheme according to \( T \). In particular, the compression tree indicates which of the second-order distributions are to be used, and the data movement scheme specifies an actual plan to implement it. More formally, let \( p(i) \) denote the parent of \( i \) in \( T \). This indicates that both \( X_i \) and \( X_{p(i)} \) should be gathered together at some common sensor node, and that \( X_i \) should be compressed using its conditional probability distribution given the value of \( X_{p(i)} \) (i.e., \( p(X_i|X_{p(i)} = x_{p(i)}) \)). The compressed value is communicated to the base station along the shortest path, resulting in an entropy rate of \( H(X_i|X_{p(i)}) \). Finally, the root of the tree, \( r \), sends its own value directly to the base station, resulting in an entropy rate of \( H(X_r) \). It is easy to see that the base station can reconstruct all the values. The data movement plan specifies how the values of \( X_i \) and \( X_{p(i)} \) are collected together for all \( i \). In this paper, we address the optimization problem of finding the optimal compression tree that minimizes the total communication cost, for a given communication topology and a given probability distribution over the sensor network variables (or the entropy rates for all variables, and the joint entropy rates for all pairs of variables). We note that the notion of compression trees is quite similar to the so-called Chow-Liu trees [18], used for approximating large joint probability distributions.

III. Greedy Algorithm

We next present a generic greedy framework that helps us analyze the rest of the problems. Suppose node \( p(i) \) is the parent of node \( i \) in the compression tree \( T \). Let \( l_i,p(i) \) denote the node where \( X_i \) is compressed using \( X_{p(i)} \). We note that this is not required to be \( i \) or \( j \), and could be any node in the network. This makes the analysis of the algorithms very hard. Hence we focus on the set of feasible solutions of the following restricted form: \( l_i,j \) is either node \( I_o \) or \( j \). The following lemma states that the cost of the optimal restricted solution is close to the optimal cost. Lemma 1: Let the optimal solution be \( OPT \) and the optimal restricted solution be \( g OPT \). We have cost( \( g OPT ) \leq cost( OPT ) \). Furthermore, for WL-SG model, cost( \( g OPT ) \leq 2cost( OPT ) \). Our algorithm finds what we call an extended compression tree, which in a final step is converted to a compression tree. An extended compression tree \( T \) corresponding to a compression tree \( T \) has the same underlying tree structure, but each edge \( e(i, j) \in T \) is associated with an orientation specifying the raw data movement. Basically, an extended compression tree naturally suggests a restricted solution in which an edge from \( i \) to \( j \) in \( T \) implies that \( i \) ships its raw data to \( j \) and the corresponding compression is carried out at \( j \). We note that the direction of the edges in \( T \) may not be

![Diagram](image_url)

Fig1 (i) connected dominating set of the sensor network is indicated by the shaded nodes, (ii) the corresponding comparison tree.

The same as in \( T \) where edges are always oriented from the root to the leaves, irrespective of the data movement. In the following, we refer the parent of node \( i \) to be the parent in \( T \), i.e., the
node one hop closer to the root, denoted by \( p(i) \).
The main algorithm greedily constructs an extended compression tree by greedily choosing subtrees to merge in iterations. We start with a empty graph \( F_1 \) that consists of only isolated nodes. During the execution, we maintain a forest in which each edge is directed. In each iteration, we combine some trees together into a new larger tree by choosing the most (or approximately) cost-effective tree star corresponding data movement is added to our solution. The algorithm terminates when only one tree is left which will be our extended compression tree ~\( T \). Let \( r \) be the center of \( TS \) and \( S \) We now discuss the final data movement scheme and how the cost of the final solution has been properly accounted in the tree stars that were chosen. Suppose in some iteration, a treestar \( TS \) is chosen in which the center node \( r \) sends its raw information to each \( v_j (v_j \in T_j, j \in S) \) (S is the set of indices of leaf-trees in \( TS \)). The definition of the cost function suggests that \( X_v \) is compressed using \( X_r \) at \( v_j \), and the result is sent from \( v_j \) to BS. However, this may not be consistent with the extended compression tree ~\( T \). In other words, some \( v_j \) may later become the parent of \( r \), due to latter treestars being chosen, in ~\( T \) which implies that \( r \) should be compressed using \( v_j \) instead of the other way around. Suppose some leaf \( v_p (v_p \in T_p, p \in S) \) is the parent of \( r \) in ~\( T \). The actual data movement scheme is determined as follows. We keep the raw data movement induced by \( T \) unchanged, i.e., \( r \) still sends \( X_r \) to each \( v_j (j \in S) \). But now, \( X_r | X_p \) instead of \( X_p | X_r \) is computed on node \( v_p \) and sent to the base station. Other leaves \( v_j (j \neq p) \) still compute and send \( X_v | X_r \). It is easy to check this data movement scheme actually implements the extended compression tree ~\( T \). For instance, in Figure 3, node 3 is initially the parent of node 1, but later node 4 becomes the parent of node 1, and in fact node 1 ships \( X_1 | X_4 \) to the base station (and not \( X_1 | X_3 \)). Node 1 now being the parent of node 3 also compresses \( X_3 \) and sends \( X_3 | X_1 \) to BS. The pseudocode for constructing ~\( T \) and the corresponding communication scheme is given in Generic Greedy Algorithm.

\[
\mathcal{F}_i = \bigcup_{j=1}^{n} \{\{X_i\}\}; \\
i \rightarrow 1; \\
\textbf{while } \mathcal{F}_i \text{ is not a spanning tree do} \\
\quad TS_i = \text{Mce-Treestar}(\mathcal{F}_i); \\
\quad \text{Let } E(\mathcal{F}_i) \text{ be leaf-edges of } TS_i \text{ and } r \text{ is the center of } TS_i; \\
\quad \mathcal{F}_{i+1} = \mathcal{F}_i + E(\mathcal{F}_i); \\
\quad T_r = T_r + IC(\mathcal{F}_i); \\
\quad i = i + 1; \\
\quad \bar{T} = \mathcal{F}_i; \\
\textbf{for each directed edge } e(i,j) \in E(\bar{T}) \text{ do} \\
\quad \text{if } i \text{ is the parent of } j \text{ then} \\
\quad \qquad \text{Compute } X_j | X_i \text{ at } j \text{ and send it to BS}; \\
\quad \text{else} \\
\quad \qquad \text{Compute } X_i | X_j \text{ at } j \text{ and send it to BS};
\]

easy to see \( ' \) must be smaller than \( n \), the algorithm terminates. We assume in each iteration, Mce-Treestar is guaranteed to find an approximate most cost-effective treestar. We assume further the bounded conditional entropy parameter is We conducted a comprehensive simulation study over several datasets comparing the performance of several approaches for data collection. Our results illustrate that our algorithms can exploit the spatial correlations in the data effectively, and perform comparably to the DSC lower bound. Below we present results over a few representative settings. Wireless sensor networks have been a very active area of research in recent years (see [24] for a survey). Due to space constraints, we only discuss some of the most closely related work on data collection in sensor networks here. Directed diffusion [25], TinyDB [26], LEACH [27] are some of the general purpose data collection mechanisms that have been proposed in the literature. The focus of that work has been on designing protocols and/or declarative interfaces to collect data, and not on optimizing continuous data collection. Aside from the works discussed earlier in the paper [7], [8], [9], the BBQ system [23] also uses a predictive modeling-based approach to collect data from a sensor network. However, the BBQ system only provides probabilistic, approximate answers to queries, without any guarantees on the correctness. Scaglione and Servetto [14] also consider the interdependence of routing and data compression, but the problem they focus on (getting all data to all nodes) is different from.
the problem we address. In seminal work, Gupta and Kumar [28] proved that the transport capacity of a random wireless network scales only as $O(npn)$, where $n$ is the number of sensor nodes. Although this seriously limits the scalability of sensor networks in some domains, in the kinds of applications we are looking at, the bandwidth or the rate is rarely the limiting factor; to be able to last a long time, the sensor nodes are typically almost always in sleep mode. Several approaches not based on predictive modeling have also been proposed for data collection in sensor networks or distributed environments. For example, constraint chaining [29] is a suppression-based exact data collection approach that monitors a minimal set of node and edge constraints to ensure correct recovery of the values at the base station.

IV. Conclusion

In this paper we have discussed Generic Greedy Algorithm that can optimally exploit the strong spatial correlations typically observed in a given sensor network remains an open problem. In this paper, we considered this problem with the restriction that the data collection protocol can only utilize second-order marginal or conditional distributions. We analyzed the problem, and drew strong connections to the previously studied weakly connected dominating set problem. This enabled us to develop a greedy framework for approximating this problem under various different communication model or solution space settings. Although we are not able to obtain constant factor approximations, our empirical study showed that our approach performs very well compared to the DSC lower bound. We observe that the worst case for the problem appears to be when the conditional entropies are close to zero, and that we can get better approximation bounds if we lower-bound the conditional entropies. Future research directions include generalizing our approach to consider higher-order marginal and conditional distributions, improving the approximation bounds by incorporating lower bounds on the conditional entropy values, and also understanding how to apply such approximation algorithms in practice in presence of node and communication link.

References


B.Lakshmi Prasanna was born in Ongole, Prakasam Dt, Andhra Pradesh, India. She received MCA from university of JNTU,Kakinada ,India. Presently, she is pursuing M.Tech in C.S.E from QIS College of Engineering & Technology, vengamukkapalem, Ongole, Prakasam Dt, Andhra Pradesh, India. Her Research interest includes Computer Networks, Data Mining.

T.Sunitha was born in Ongole, Prakasam Dt, Andhra Pradesh, India. She received B.Tech from university of JNTUH, Hyderabad, and M.Tech from Andhra University, Andhra Pradesh, India. Presently, she is working as a Associate Professor in C.S.E from QIS College of Engineering & Technology, vengamukkapalem, Ongole, Prakasam Dt, Andhra Pradesh, India. Her Research interest includes Computer Networks and Data Mining.