Reliable Data Gathering in Wireless Sensor Network Using LLHC

Lalit Patil, Archana Augustine, Dr. Shrikant Charhate

Pillai HOC College of Engineering and Technology, Rasayani, Raigad, 410207, India

Abstract - Data gathering is a fundamental operation in various applications of wireless sensor networks (WSNs), where sensor nodes sense information and forward data to a sink node via multi-hop wireless communications. Typically, data in a WSN is relayed over a tree topology to the sink for effective data gathering. A number of tree-based data gathering schemes have been proposed in the literature, most of which aim at maximizing network lifetime. A specific characteristic of sensor network applications is that the major traffic consists of data collection from various sensor source nodes to a sink via a unidirectional tree. Since the reliability of a link is highly related to its signal to interference plus noise ratio (SINR), the SINR of all the currently used links on the data gathering tree should be greater than a threshold to guarantee high reliability. We formulate the joint problem of tree construction, Our simulation results show that the proposed algorithms achieve much lower data gathering latency than existing data gathering strategies while guaranteeing high reliability. Moreover, the algorithms also have a comparable energy efficiency and network lifetime to other algorithms.

Keywords - Wireless Sensor Networks (WSNs), Data Gathering, Latency, LLHC, Network Life.

1. Introduction

Recently, wireless sensor networks (WSNs) have exhibited great potentials as a new information-gathering approach for many applications, such as structure monitoring, security surveillance, and wildlife preservation. Besides sensing interested information, the paramount task in WSNs is how to gather data from scattered sensors. Typical approaches for data gathering are to forward sensed data to a static data sink via a few selected relaying nodes or dynamic routing [1]. In more complex approaches, sensed data are aggregated or compressed at relaying nodes by exploring spatiotemporal correlation [2]–[4], which introduces extra delay and may not be applicable to all applications to employ a mobile collector that roams around the sensing field by moving sufficiently close to sensors so as to collect data from them via short-range or direct communications. Though this approach can significantly reduce the energy consumption of multi-hop relaying, the limit on the velocity of mobile collectors makes it difficult to complete data gathering timely in large-scale WSNs.

The WSN acts as the data source for the cloud and mobile users are the data requesters for the cloud. With just a simple client on their mobile devices, mobile users can have access to their required sensory data from the cloud, whenever and wherever there is network connection. Evolving as the concept of “sensor-cloud”, the integrated WSN-MCC is “an infrastructure that allows truly pervasive computation using sensors as an interface between physical and cyber worlds, the data-compute clusters as the cyber backbone and the internet as the communication medium.”

2. Related Work

Distributed protocol for constructing a data gathering tree was presented in [1], where each node maintains its own data gathering links based on local information. In [2], sensors are organized into clusters based on the correlation of their sensed data, to maximize the data aggregation level and minimize the number of packets to be sent to the sink. In addition, some work assumed that an intermediate node is capable of aggregating all received packets and its own sensed data into a single packet. Based on such an assumption, the problem of finding a maximum lifetime data gathering tree in WSNs was proved to be NPhard in [3] and an approximation...
algorithm was provided. In [4], an optimal algorithm was presented to find a maximum lifetime tree from all shortest path trees rooted at the sink for data gathering by showing that it is equivalent to solving a semi-matching problem between nodes from any two adjacent levels. However, the assumption of perfect data aggregation in these schemes may not hold in general WSNs. More importantly, the reliability of links on the tree was not considered in these schemes. As discussed earlier, a TDMA-based MAC mechanism can be used to ensure link reliability, which divides the MAC time into time slots and schedules a subset of links to transmit in each time slot. There has been some work on link scheduling for data gathering in WSNs.

The problem of finding the minimum length schedule for data gathering without aggregation was proved to be NP-hard in, by reducing a well-known NP-complete partition problem to it. In , a data collection scheme was proposed for WSNs, where sensor nodes determine the time slot and path for each data packet, so as to minimize the overall data gathering time. Moreover, a distributed link scheduling algorithm that requires at most 3N time slots for an N-node WSN was presented in. However, the interference model used in the above studies is inaccurate, as it determines a binary interfering relation for two links solely based on whether they are within a fixed interference range.

In reality, the interference among multiple links may be too severe for all links to transmit data simultaneously even though any two of them are interference free based on the interference model. This is because that interference from multiple transmitters is accumulated at each receiver. Consequently, links scheduled by these schemes may have poor reliability. In a multi-channel scheme and a joint link scheduling and power assignment scheme are proposed for data gathering in WSNs over an arbitrary minimum spanning tree. However, how to construct the spanning tree to achieve the lowest data gathering latency is not studied.

3. Proposed Method

The SINR-based data gathering problem in a WSN can be formally described as follows. Given a wireless sensor network consisting of a set of sensor nodes, N, the set of traffic demands for all nodes, D, and a set of directed links among them, L, find a subset of links from L to form a tree T rooted at the sink node, allocate transmitting time slots for links on T, and assign transmitting power to active links in each time slot, such that the traffic demand of all nodes is satisfied, and the number of allocated time slots is minimized. The reliability of data gathering is guaranteed, as the SINR of all active links is always above the threshold λ.

In LLHC algorithm, the data gathering tree is constructed iteratively and a new node is added to the tree in each step. To determine the next node to join the tree and which tree node it should connect to, we define a weight to reflect the distribution of traffic load if a new node joins, as well as the incompatibility among links that are already on the tree and the potential links the new node may introduce. For any sub-tree on the data gathering tree, we define its traffic load as the aggregated traffic demands of all nodes on the subtree.

Then we define the maximum sub-tree traffic $D_{max}$ sub(T) of tree $T$ as the maximum traffic load of all sub-trees whose ancestor is the sink node. In this way, we could reduce the data gathering latency by minimizing the maximum subtree traffic of the tree, as the traffic load will be balanced among all subtrees whose ancestor is the sink. In addition, we define the incompatibility degree $int(T)$ as the summed incompatibility of all links on the tree. Then the weight of the tree is the sum of maximum subtree traffic and the incompatibility degree,

$$W(T) = D_{max} \text{sub}(T) + int(T) \quad (1)$$

For each new node $u$ to join the tree, we consider the additional traffic demands and incompatibility introduced by $u$ and its 1-hop neighbours in the network that are not on the tree yet. Here any node $v$ can be regarded as a 1-hop neighbour of $u$ if there exists a link from $v$ to $u$ in $L$, since $u$ might be responsible for relaying the traffic of $v$. Because in the worst case, all 1-hop neighbours of node $u$ need to connect to the tree through $u$.

The additional traffic load from node $u$ and its 1-hop neighbours is the sum of their traffic demands, regardless of how they are connected to $u$. On the other hand, the incompatibility introduced by node $u$ and its 1-hop neighbours is related to the structure of the subtree among node $u$ and its neighbours. In most cases, neighbours of node $u$ can join the tree by connecting to other tree nodes, hence we will only use the spanning tree where each neighbours is directly connected with $u$ to determine the potential incompatibility introduced by $u$ and its neighbours. We use $Su$ to denote such a spanning tree from $u$ to its 1-hop neighbours.
Fig. 1. An example of WSN where nodes are not evenly distributed. Node \( a \) is the sink, while both nodes \( e \) and \( f \) have to relay the traffic from the nearby hotspot. Solid lines stand for links that are already on the data gathering tree.

Fig. 2. A data gathering tree generated by LLHC algorithm in a WSN, where 100 sensor nodes are randomly deployed in a \( 400 \times 400m^2 \) field. The green node is the sink node and the red nodes are sensor nodes.

After obtaining a data gathering tree by LLHC algorithm, the next step is to allocate time slots and transmitting power to links on the tree to minimize data gathering latency. When scheduling links for a new time slot, links with high traffic load should be considered first as they need more time slots to transmit the traffic than other links. On the other hand, a link whose corresponding vertex on \( I \) has a large degree should be given high priority also, since it prevents the scheduling of all other links incompatible with it in the same time slot. Based on these observations, we propose a maximum weight first (MWF) algorithm for link scheduling and transmitting power assignment, where the weight of a link is defined as the sum of the remaining traffic load over the link and the degree of the corresponding vertex of the link on \( I \).

Fig. 3. Simulation configuration, results

In this simulation uses different traffic rate which is shown in Fig 3. Flow start at 2 and traffic for sending data packets are low.

Fig. 4. In this simulation uses different timeslot which is shown in Fig 4. Which describe that time slot start at 50 in x direction provide better result than time slot used in different nodes.

4. Conclusion

In this paper, we have studied tree-based data gathering in WSNs. Our objective is to gather data from all sensors with low latency and high reliability, by carefully
constructing a data gathering tree, scheduling links on the tree, and assigning transmitting power levels to active links in each time slot. We have formulated the problem into an optimization problem and proved it is NP-hard. We then divided the problem into two subproblems and provided a heuristic algorithm for each subproblem. We have conducted extensive simulations to evaluate the proposed algorithms. The results demonstrate that the proposed algorithms can significantly reduce the data gathering latency under various node densities, SINR thresholds, and traffic demands, while guaranteeing that the SINR of each active link is above the threshold. In addition the algorithms can distribute the relaying traffic well onto the data gathering tree, and have low requirement on the buffer size on sensor nodes. Furthermore, the simulation results show that the proposed algorithms achieve such low latency and high reliability without sacrificing the energy efficiency and lifetime of the network compared to other algorithms.

References


