Optimizing a Data Routing Protocol’s Influence on Communication Connectivity

Md Ashiqur Rahman and Sajid Hussain
Jodrey School of Computer Science
Acadia University
Wolfville, Nova Scotia, Canada
{082654r, sajid.hussain}@acadiau.ca

Abstract—A global energy efficient wireless sensor network data routing protocol typically uses few suitably located sensors to relay other sensors’ data. These overused sensors reasonably consume more energy and become inactive earlier. Some other sensors then take their role of relaying other sensors’ data. At some point in this cycle, a sensor with sufficient energy to sense and communicate can be disconnected from the network for not having any active sensor within its transmission range. If the used data routing protocol is modified such that a sensor is connected as long as it has sufficient energy then the actual energy efficiency of the data routing protocol will be compromised. In this paper, our proposed approach modifies an existing data routing protocol that minimizes the total energy consumption of all sensors. Our modification fulfills the connectivity requirement without compromising the existing energy efficiency of the data routing protocol.

I. INTRODUCTION AND RELATED WORK

There is little agreement about what precisely a Wireless Sensor Network (WSN) should contain. Even though a WSN is constraint to have some sensing device or sensor, there is no restriction that it cannot have other heterogeneous components. For this reason, section II first formally defines a WSN’s architecture. A WSN can have other form than the proposed one, but we will limit our focus only on the proposed version.

After the elements and the basic activities of a WSN are defined, section IV modifies a previously documented WSN data routing protocol to achieve better connectivity. The original protocol minimizes sensors’ total energy consumption without assuming sensors’ limited transmission range [1]. Our modification will simply make this existing protocol more useful in real application. Section V also evaluates our approach using a method described in section III.

Our work has close connection with those in [2], [3], and [4]. [2] solves the problem of minimizing energy consumption of each sensor locally while ensuring two global properties: communication connectivity, and sensing coverage. Since the optimization is performed locally, it is not globally optimal. [3] presents a group communication scheme for collaborative information processing in WSNs. This scheme ensures efficient communication among a subset of mobile sensors in a collaboration group. Unicast and multicast communication are special cases of this scheme where the subset contains one and all group members, respectively. [4] addresses the problems of achieving communication connectivity and sensing coverage first separately and then jointly. The paper also proposes a mechanism to dynamically adjust coverage by scheduling different sleep intervals for different sensors. Unlike these work, ours improves an existing globally energy efficient data routing protocol to achieve better connectivity without compromising its actual energy efficiency. Our work also differs in the aspect that it integrates heterogeneous components within the WSN to improve connectivity.

II. A GENERIC MULTIPURPOSE WSN ARCHITECTURE

The top most element of the WSN is the Base Station (BS). The BS is the ultimate storage and processor of all sensed data. It is assumed to have sufficient energy source to ignore its energy consumption. All other elements are powered locally with limited energy source which may not be replenishable.

An element of the WSN that resides within the deployment field is called a node. A node can be either a sensor or a proter. A sensor is a device that can sense its vicinity. A sensor sends its sensed data to the BS if queried. The sensed data is sent as part of a data packet. All nodes including the BS can receive and transmit data packets using wireless radio channel.

A proter, unlike a sensor, cannot sense. However, it can have additional features such as caching, extra energy, more processing power etc.

How a data packet will travel from a sensor to the BS is determined by a pre-specified data routing path. At any round, all data routing paths resemble a data routing tree that is rooted at the BS and spans all active sensors and zero or more active proters. The BS constructs sufficient data routing trees for the WSN’s expected lifetime and disseminates them to all nodes before the first round. The BS uses the initial energy and location information of the nodes to construct these trees. It also uses a sensor model and a radio model to estimate nodes’ energy consumption due to sensing and data routing,

1The deployment field is the region that is meant to be monitored by the WSN.
2A sensed data contains information about an event’s occurrence (e.g. an object’s presence) or specifies a variable’s value (e.g. environmental humidity).
3The word ‘proter’ is a combination of processor and router.
4A round is a timeslot in the WSN’s lifetime.
respectively. The expected lifetime is valid only if all sensors are queried at every round\(^3\).

**A. In Depth**

1) **Query:** A query is a request from the BS to a sensor to send its sensed data. There are three types of queries: continuous, manual, and self-generated. A continuous query specifies all or a group of sensors to send their data to the BS for some consecutive rounds. A manual query asks one or more sensor to send their sensed data at an arbitrary round. In a self-generated query, a sensor itself sends its sensed data to the BS whenever the sensed data validates some condition. This paper will only consider a default continuous query where all active sensors will send their sensed data to the BS at every round until they are all inactive.

2) **Data packet:** A data packet has two parts: (i) preamble, and (ii) data. The preamble contains sender-receiver authentication information. The data contains either a sensor’s sensed data or a query or maintenance information from the BS.

3) **Round:** A WSN’s lifetime is divided into equal rounds. The duration of each round is same and the activities within a round are similar. At the beginning of a round, all active nodes turn on their radios and wait a fixed amount of time to receive any packet from the BS. If received, the packet will contain a query or maintenance information. Next, all active sensors send their previous round’s sensed data to the BS. This is the time when all in-network data routing occurs according to the data routing tree preset for the corresponding round. After that, all nodes turn off their radios and the proter transition to their low power state. This is the time when sensors’ starts sensing. The sensors also transition to their low energy state once they finish sensing. All nodes remain in their low power state until the next round begins.

4) **Nodes’ Location:** The BS requires all nodes’ location information to build the data routing trees. Therefore, whenever a node is deployed in the WSN, an entry must be added at the BS describing the node’s authentication and location information. The easiest way of locating a node is to locate it relative to another node whose location is known. We will assume all nodes including the BS remain in one place throughout the WSN’s expected lifetime.

5) **Constructing a Data Routing Tree:** The BS constructs the data routing trees that minimize all active sensors’ total energy consumption at every round. The trees also ensure that sensed data from all active sensors are received by the BS at every round.

### III. THE REASONS FOR A SENSOR’S INACTIVITY

Let’s assume that the BS uses MITECRO - a data routing tree construction algorithm that minimizes the total energy consumption of all sensors at every round. MITECRO minimizes the total energy consumption of all sensors over all rounds by minimizing their energy consumption at every round. The algorithm is documented in [1]. A proof of its validity is also given there.

Recall that the BS uses a sensor model and a radio model to estimate sensors’ energy consumption. When all sensors are queried at every round, a sensor’s energy consumption is highly proportional to its usage by other sensors to relay their data packets. If the BS expects that a sensor not to have sufficient energy to be active at a particular round, it does not use that sensor to build the data routing trees for that and the following rounds. The BS also stops using a sensor whenever it fails to construct a tree that routes its packet to the BS. This can happen only when the sensor or its nearby sensors do not have sufficient energy to route its packet. If we assume that the sensors’ energy source is non-replenishable, then such forceful inactivation is justified.

The following definitions should be interpreted in terms of the BS’s expectation. For instance, ‘a sensor is inactive after some round’ should be interpreted as ‘the BS expects a sensor to be inactive after some round’.

**Definition 1:** A sensor inactivation due to insufficient energy to sense in a round is called a Pure inactivation or Pi (pronounced as ‘pai’).

**Definition 2:** A sensor inactivation that is not a Pi is a Forced inactivation or Fi (pronounced as ‘fai’).

**Definition 3:** A Fi that occurs at round 1 is an Initial Fi or IFi (pronounced as ‘i fai’). The only possible reason for an IFi is faulty sensor distribution that places no sensor within the maximum transmission range of some sensor.

**Definition 4:** A Fi that occurs after round 1 is an Eventual Fi or EFi (pronounced as ‘e fai’). The probability of an EFi occurrence is influenced by both the initial sensor distribution and the data routing protocol used.

### A. Finding the Reason of a Sensor’s Inactivity

The BS stops creating a new data routing tree after the round it expects all sensors to be inactive. At this point, all sensors’ expected residual energy can be analyzed to find their reasons of inactivity. If an inactive sensor has energy less than the energy required to sense in a round then the sensor must be a Pi victim. On the other hand, if a sensor has energy greater than or equal to the energy required to sense in a round then it must be a Fi victim. To distinguish whether a sensor is actually an IFi or EFi victim, we must keep track of the sensors that are inactivated at the first round. These are the IFi victim sensors. All other Fi victim sensors are actually EFi victims.

This way we can count the number of sensors died for a particular reason, say EFi. However, this is not enough for a good analysis because for example a 0.5 EFi (i.e. an EFi with 50% residual energy) must be considered worse than any EFi less than 0.5. Therefore, in addition, inactive sensor counts should be plotted and analyzed against different residual energy. We will use this approach in section V to analyze experimental results.

\(^3\)A node is active if it is not inactive. A sensor is inactive if it does not have enough energy to sense or if it cannot send its data packet to the BS. A node is inactive if it has no energy. While the BS constructs the data routing trees, if it assumes a node to be inactive after some round, it does not use that node in constructing the data routing trees for the following rounds. In other words, once a node is inactive it is inactive for all, unless it is less used than expected and a new tree construction process is initiated after the WSN’s expected lifetime. We will not discuss such process in this paper.
IV. MINIMIZING THE EXTENT OF FORCED SENSOR INACTIVATION

From the discussion of the previous section, it is clear that Pi is more energy efficient than Fi. Hence, the goal of any energy efficient WSN data routing protocol should be increasing Pi and decreasing Fi. The straightforward way to do this is to increase the probability of an active sensor having another active sensor or the BS within its transmission range at every round. Also, at least one active sensor must have the BS within its transmission range at every round.

Since an energy efficient data routing protocol must prefer using sensors closer to the BS to relay distant sensors’ packets, sensors’ energy consumption rate will be higher - the closer they are to the BS. As a result, some sensors will be inactive earlier than the others. This may isolate and therefore inactivate some other sensor with possibly high residual energy. If this happens then we say an EFI has occurred. The extent of this EFI is proportional to the victim sensor’s residual energy.

To minimize EFI, sensors can be deployed with variable energy based on their expected usage. Unfortunately, the exact usage of a sensor is undeterminable until the energy and location of all sensors are known to the BS and the BS has constructed all data routing trees! The next section will solve this problem by deploying variable energy proters and modifying the MITECRO data routing protocol.

A. MITECRO Upgrade

1) Node Distribution: Let’s assume that all nodes and the BS will be deployed on the same plane\(^6\). Let the total number of sensors in the WSN be \(s\). All these \(s\) sensors will be deployed using a uniformly distributed random coordinates generator. Once sensors’ locations are chosen, proters’ locations will be selected.

To choose proters’ locations, first embed the deployment field in a rectangle with width and height multiples of \(\frac{r}{2\sqrt{5}}\) (see figure 1). Here \(r\) is the nodes’ common maximum transmission range. A node’s maximum transmission range is the distance until which it can send packets to another node with no miss. If nodes do not have the same maximum transmission range then \(r\) will be the minimum of their maximum transmission ranges.

Divide the rectangle that embeds the deployment field into squares with sides \(\frac{r}{2\sqrt{5}}\). Each such square will have diagonals with length \(r\). Choose routers’ potential locations in the middle of these squares. Any potential location that falls outside the deployment field must be discarded. Proters will be deployed in all potential locations that are not discarded.

2) Setting Nodes’ Energy: Assume all sensors will have the same initial energy \(e_s\). Proters’ initial energy will be then computed in two steps. In the first step, we make an assumption that if a proter is expected to be used by \(n\) sensors to relay their packets then the proter must have \(n.e_s\) energy. In the second step, we will correct this assumption and reassign proters’ energy accordingly.

\(a)\) Setting Proters’ Energy: Step 1: Let the number of proters deployed be \(p\). Assign the energy of the \(i\)-th proter, \(ep_i = 0\). For each \(j\)-th proter of all \(k\) proters that are within \(r\) distance of the BS, assign \(ep_i = \frac{e_s}{r}\). Consider all these \(k\) proters both energy assigned and recently energy assigned. Until all proters are energy assigned, for each \(m\)-th proter of all \(n\) proters that are recently energy assigned, do the following: for each \(c\)-th proter of all \(d\) proters that are within \(r\) distance of the \(m\)-th proter, assign \(ep_i = ep_i + \frac{e_s}{d}\). Consider only the first time energy assigned proters as recently energy assigned.

\(b)\) Setting Proters’ Energy: Step 2: Notice that the distance between nearby proters is always between \(\frac{2r}{2\sqrt{5}}\) and \(r\), whereas the distance between a sensor and its nearest proter is between \(\frac{2r}{2\sqrt{5}}\) and \(\frac{r}{2}\). Since a proter cannot increase sensors’ energy consumption, if it receives any packet from a sensor, it must transmit it to another proter. Moreover, if a sensor communicate in the most energy efficient way, it will never send a packet to more than \(\frac{r}{2}\) distance. Hence, the assumption that ‘if a proter is expected to be used by \(n\) sensors to relay their packets then the proter must have \(n.e_s\) energy’ is wrong. Therefore, correct the assigned energy of each proter as \(ep_i = \frac{e_s}{d}\) where \(d\) is the average distance between a sensor and its nearest proter.

3) Modifying the MITECRO Data Routing Tree Construction Algorithm: The data routing trees are constructed at the BS and disseminated to all nodes before the first round of data gathering. The modifications that need to be performed in the MITECRO data routing tree construction algorithm [1] are stated below. Though we will primarily state the modifications, a prior knowledge of MITECRO may not be necessary.

As the original MITECRO works, at every round all active sensors are set to route their data packet to the BS such that their total energy consumption is minimized. To make sure

\(^6\)This constraint can be violated as long as a radio model that works with only horizontal distance gives accurate estimate of nodes’ energy consumption due to radio transmission.
sensors do not count proter’s energy consumption, the data routing tree construction process will be divided into two parts. In the first part, the BS will create the portion of the routing tree that connects all active proters to the BS in most energy efficient way. In the second step, all active sensors’ data will be routed to the BS such that their (i.e. sensors’) total energy consumption is minimized.

The ultimate data routing trees will minimize all active sensors’ total energy consumption at every round. They will also make the most efficient use of proter’s energy to route sensors’ data. Together the data routing protocol and the proter’s deployment and energy distribution strategy will work to keep each sensor connected to the BS as long as it has sufficient energy to sense and transmit a packet to its nearest proter, which must be within half of its maximum transmission range.

V. EXPERIMENTS

We simulated both the original MITECRO and MITECRO upgrade to compare their performance. Since the original MITECRO does not use proter, to keep the comparison fair, we deployed proter in both approach. In MITECRO upgrade, the location and energy of each proter is set according to the method described in section IV-A.2. In the original MITECRO, on the other hand, proter were deployed uniformly at random and they had equal energy, just like the sensors. To keep the comparison fair, in both approach we deployed the same number of proter with same total energy.

This section will discuss about our experimental results from two scenarios - one with 20 sensors and another with 200 sensors. The detailed experiment setups are documented in table I. The graphs in figures 2 and 3 plot active sensor counts against the rounds in the two scenarios, respectively. As these figures show, in scenario 1 both the original MITECRO and MITECRO upgrade give the same network lifetime. However, unlike the original MITECRO, MITECRO upgrade keeps most of the sensors active until the last rounds. In scenario 2, on the other hand, MITECRO upgrade almost doubles the network lifetime compared to the original MITECRO. This is not surprising as because MITECRO does not take any measure to keep all sensors connected to the BS; thus, when many sensors are deployed (as in scenario 2) and few nodes are disconnected, several other loosely connected nodes are also disconnected almost immediately.

Table II lists sensor counts against their reason of inactivity. As expected, in scenario 2 where MITECRO upgrade had 171 Pi and only 29 EFi, the original MITECRO had 50 Pi and 150 EFi. Literally this means that, unlike MITECRO upgrade, in the original MITECRO most sensors became inactive not because they did not have sufficient energy to sense, but because they didn’t have another node within their maximum transmission range which can relay their data packets.

A node is loosely connected if it has relatively fewer nodes within its maximum transmission range.

A. Possible Limitations of MITECRO Upgrade

MITECRO upgrade is basically a polynomial time-complex algorithm which minimizes the total energy consumption of all sensors at every round of data gathering and also keeps almost all sensors active until the very end of network lifetime. Despite these facts, the upgrade has few limitations. First, for the upgrade to work as expected, if a new node is added to the WSN, its energy and location information must be entered at the BS. The BS must also rebuild all data routing trees and inform all nodes about the changed trees. Second, a true 3-dimensional node deployment and presence of signal blocking objects within the deployment field were not considered. We expect that the upgrade would work as expected as long as the used sensor model and the radio model can accurately estimate

IFI victim sensors are not counted because their reason of inactivity is independent of the data routing protocol.
nodes’ energy consumption due to sensing and data routing, respectively.

VI. CONCLUSION

In this paper, we first presented a generic multipurpose wireless sensor network architecture and described its elements and activities. Then we modified an existing globally energy efficient WSN data routing protocol. The modified data routing protocol minimizes the total energy consumption of all sensors while keeps them connected until the very last rounds. We also demonstrated a method of finding and analyzing reasons of sensors’ inactivity. In near future, we would like to experiment the proposed data routing protocol with other types of queries, such as manual or self-generated queries.

REFERENCES


