Synopsis of Energy Efficient Routing Protocols in Wireless Sensor Networks (WSN)

Synopsis submitted in Fulfillment of the requirements for the degree of

MASTERS OF TECHNOLOGY (HONS.)

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By
Pawan Singh Faujadar
03 CS 3006

Under the guidance of
Prof. Niloy Ganguly

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR
KHARAGPUR-721302, INDIA

Abstract:

One of the limitations of wireless sensor nodes is their inherent limited energy resource. Besides maximizing the lifetime of the sensor node, it is preferable to distribute the energy dissipated throughout the wireless sensor network in order to minimize maintenance and maximize overall system performance. Any communication protocol that involves synchronization of peer nodes incurs some overhead for setting up the communication. So here we study various energy-efficient routing algorithms, and compare among themselves. We take into account the setup costs and analyze the energy-efficiency and the useful lifetime of the system. In order to better understand the characteristics of each algorithm and how well they really perform, we also compare them with an optimum clustering algorithm.

The benefit of introducing these ideal algorithms is to show the upper bound on performance at the cost of an astronomical prohibitive synchronization costs. We compare the algorithms in terms of system lifetime, power dissipation distribution, cost of synchronization, and simplicity of the algorithm.

1. INTRODUCTION:

Over the last half a century, computers have exponentially increased in processing power and at the same time decreased in both size and price. These rapid advancements led to a very fast market in which computers would participate in more and more of our society's daily activities. In recent years, one such revolution has been taking place, where computers are becoming so small and so cheap, that single-purpose computers with embedded sensors are almost practical from both economical and theoretical points of view. Wireless sensor networks are beginning to become a reality, and therefore some of the long overlooked limitations have become an important area of research.

In our project, we attempt to find out and overcome limitations of the wireless sensor networks such as: limited energy resources, varying energy consumption based on location, high cost of transmission, and limited processing capabilities. All of these characteristics of wireless sensor networks are complete opposites of their wired network counterparts, in which energy consumption is not an issue, transmission cost is relatively cheap, and the network nodes have plenty of processing capabilities. Routing approaches that have worked so well in traditional networks for over twenty years will not suffice for this new generation of networks.

Besides maximizing the lifetime of the sensor nodes, it is preferable to distribute the energy dissipated throughout the wireless sensor network in order to minimize maintenance and maximize overall system performance. Any communication protocol that involves synchronization between peer nodes incurs some overhead of setting up the communication. So here, we attempt determine whether the benefits of more complex routing algorithms overshadow the extra control messages each node needs to communicate. Each node could make the most informed decision regarding its communication options if they had complete knowledge of the entire network topology and power levels of all the nodes in the network. This indeed proves to yield the best performance if the synchronization messages are not taken into account. However, since all the nodes would always need to have global knowledge, the cost of the synchronization messages would ultimately be very expensive. For both the diffusion and clustering algorithms, we will analyze both realistic and optimum schemes in order to gain more insight in the properties of both approaches.

The usual topology of wireless sensor networks involves having many network nodes dispersed throughout a specific physical area. There is usually no specific architecture or hierarchy in place and therefore, the wireless sensor networks are considered to be ad hoc networks. An ad hoc wireless sensor network may operate in a standalone fashion, or it may be connected to other networks, such as the larger Internet through a base station. Base stations are usually more complex than mere network nodes and usually have an unlimited power supply. Regarding the limited power supply of wireless sensor nodes, spatial reuse of wireless bandwidth, and the nature of radio communication cost which is a function of the distance transmitted squared, it is ideal to send information in several smaller hops rather than one transmission over a long communication distance.

In our simulation, we use a data collection problem in which the system is driven by rounds of communication, and a sensor node has a packet to send to the distant base station if it detects a target within its sensing radius. The algorithms are mainly based on location, power levels, and load on the node, and their goal is to achieve better target sensing with minimizing the power consumption and maintenance throughout the network so that the majority of the nodes consume their power supply at relatively the same rate regardless of physical location. This leads to better maintainability of the system, such as replacing the batteries all at once rather than one by one, and maximizing the overall system performance by allowing the network to function at 100% capacity throughout most of its lifetime instead of having a steadily decreasing node population.

2. THE ROUTING PROTOCOLS:

In the next few sub-sections, we will discuss the protocols tested in detail. Briefly, the protocols are:

1. Direct communication, in which each node communicates directly with the base station.
2. Diffusion-based algorithm utilizing only location data.
3. E3D: Diffusion based algorithm utilizing location, power levels, and node load.
4. Random clustering, similar to LEACH, in which randomly chosen group heads receive messages from all their members and forward them to the base station.
5. An optimum clustering algorithm, in which clustering mechanisms are applied after some iterations in order to obtain optimum cluster formation based on physical location and power levels.

2.1 Direct Communication

Each node is assumed to be within communication range of the base station and that they are all aware who the base station is. In the event that the nodes do not know who the base station is, the base station could broadcast a message announcing itself as the base station, after which all nodes in range will send to the specified base station. So each node sends its data directly to the base station. Eventually, each node will deplete its limited power supply and die. When all nodes are dead and the system is said to be dead.

The main advantages of this algorithm lie in its simplicity. There is no synchronization to be done between peer nodes, and perhaps a simple broadcast message from the base station would suffice in establishing the base station identity. The disadvantages of this algorithm are that radio communication is a function of distance squared, and therefore nodes should opt to transmit a message over several small hops rather than one big one; nodes far away from the base station will die before nodes that are in close proximity of the base station. Another drawback is that communication collision could be a very big problem for even moderate size networks.

2.2 Diffusion based algorithm using location information

Each node is assumed to be within communication range of the base station and that they are all aware who the base station is. Once the base station identity is established, the second sequence of messages could be between each node and several of their closest neighbors. Each node is to construct a local table of signal strengths recorded from each of their neighbors, which should be a direct correlation to the distance those nodes are from each other. The other value needed is the distance from each neighbor to the base station, which can be figured out all within the same synchronization messages. This setup phase needs only be completed once at the startup of the system; therefore, it can be considered as constant cost and should not affect the algorithm's performance beyond the setup phase.

In every transmission, each node sends its data that is destined for the base station, to the best neighbor. Each node acts as a relay, merely forwarding every message received to its respective neighbor. The best neighbor is calculated using the distance from the sender and the distance from the neighbor to the base station. This ensures that the data is always flowing in the direction of the base station and that no loops are introduced in the system; this can be accomplished by considered not only the distance from the source to the candidate neighbor, but also from the candidate neighbor to the base station. Notice that the complete path is not needed in order to calculate the best optimal neighbor to transmit to. Since each node makes the best decision for itself at a local level, it is inferred that the system should be fairly optimized as a whole.

The main advantage of this system is its fairly light complexity, which allows the synchronization of the neighboring nodes to be done relatively inexpensive, and only once at the system startup. The system also distributes the lifetime of the network a little bit more efficiently.

The disadvantage of this system is that it still does not completely evenly distribute the energy dissipated since nodes close to the base station will die far sooner before nodes far away from the base station. Notice that this phenomenon is inversely proportional to the direct communication algorithm. It should be clear that this happens because the nodes close to the base station end up routing many messages per iteration for the nodes farther away.

2.3 E3D: Diffusion based algorithm using location, power, and load as metrics

In addition to everything that the basic diffusion algorithm performs, each node makes a list of suitable neighbors and ranks them in order of preference, similar to the previous approach. Every time that a node changes neighbors, the sender will require an acknowledgement for its first message which will ensure that the receiving node is still alive. If a time out occurs, the sending node will choose another neighbor to transmit to and the whole process repeats. Once communication is initiated, there will be no more acknowledgements for any messages. Besides data messages, there is an introduce exception messages which serve as explicit synchronization messages. Only receivers can issue exception messages, and are primarily used to tell the sending node to stop sending and let the sender choose a different neighbor. An exception message is generated in only three instances: the receiving node's queue is too large, the receiver's power is less than the sender's power, and the receiver has passed a certain threshold which means that it has very little power left.

At any time throughout the system's lifetime, a receiver can tell a sender not to transmit anymore because the receiver's queues are full. This should normally not happen, but in the event it does, an exception message would alleviate the problem. In the current schema, once the sending node receives an exception message and removes his respective neighbor off his neighbor list, the sending node will never consider that same neighbor again. We did this in order to minimize the amount of control messages that would be needed to be exchanged between peer nodes. However, future considerations could be to place a receiving neighbor on probation in the event of an exception message, and only permanently remove it as a valid neighbor after a certain number of exception messages.

The second reason an exception message might be issued, which is the more likely one, is when the receiver's power is less than the sender's power, in which if the receiver's power is less than the specified threshold, it would then analyze the receiving packets for the sender's power levels. If the threshold was made too small, then by the time the receiver managed to react and tell the sender to stop sending, too much of its power supply had been depleted and its life expectancy thereafter would be very limited while the sending node's life expectance would be much longer due to its less energy consumption. Through empirical results, we concluded that the optimum threshold is 50% of the receiver's power levels when it in order to equally distribute the power dissipation throughout the network.

In order to avoid having to acknowledge every message or even have heartbeat messages, we introduce an additional threshold that will tell the receiving node when its battery supply is almost gone. This threshold should be relatively small, in the 5~10% of total power, and is used for telling the senders that their neighbors are almost dead and that new more suitable neighbors should be elected.

The synchronization cost of e3D is two messages for each pair of neighboring nodes. The rest of the decisions will be based on local look-ups in its memory for the next best suitable neighbor to which it should transmit to. Eventually, when all suitable neighbors are exhausted, the nodes opt to transmit directly to the base station. By looking at the empirical results obtained, it is only towards the end of the system's lifetime that the nodes decide to send directly to the base station.

The main advantage of this algorithm is the near perfect system lifetime where most nodes in the network live relatively the same duration. The system distributes the lifetime and load on the network better than the previous two approaches. The disadvantage when compared to of this algorithm is its higher complexity, which requires some synchronization messages throughout the lifetime of the system. These synchronization message are very few, and therefore worth the price in the event that the application calls for such strict performance.

2.4 Random Clustering Based Algorithm

This algorithm is similar to LEACH, except there is no data aggregation at the cluster heads. Random cluster heads are chosen and clusters of nodes are established which will communicate with the cluster heads.

The main advantage of this algorithm is the distribution of power dissipation achieved by randomly choosing the group heads. This yields a random distribution of node deaths. The disadvantage of this algorithm is its relatively high complexity, which requires many synchronization messages compared to e3D at regular intervals throughout the lifetime of the system. Note that cluster heads should not be chosen at every iteration since the cost of synchronization would be very large in comparison to the number of messages that would be actually transmitted. In our simulation, we used rounds of 20 iterations between choosing new cluster heads. The high cost of this schema is not justifiable for the performance gains over much simpler schemes such as direct communication. As a whole, the system does not live very long and has similar characteristics to direct communication. Notice that the only difference in its perceived performance from direct communication is that it randomly kills nodes throughout the network rather than having all the nodes die on one extreme of the network.

The nodes that are farther away would tend to die earlier because the cluster heads that are farther away have much more work to accomplish than cluster heads that are close to the base station. The random clustering algorithm had a wide range of performance results, which indicated that its performance was directly related to the random cluster election; the worst case scenario had worse performance by a factor of ten in terms of overall system lifetime.

2.5 Ideal Clustering Based Algorithm

We implemented this algorithm for comparison purposes to better evaluate the diffusion approach, especially that the random clustering algorithm had a wide range of performance results since everything depended on the random cluster election. The cost of implementing this classical clustering algorithm in a real world distributed system such as wireless sensor networks is energy prohibitively high; however, it does offer us insight into the upper bounds on the performance of clustering based algorithms.

We implemented k-Means clustering (k represents the number of clusters) to form the clusters. The cluster heads are chosen to be the clustroid nodes; the clustroid is the node in the cluster that minimizes the sum of the cost metric to the other points of the corresponding cluster. In electing the clustroid, the cost metric is calculated by taking the distance squared between each corresponding node and the candidate clustroid and divided by the candidate clustroid's respective power percentage levels. The metric was calculate after some fixed iterations, and therefore yielded an optimal clustering formation throughout the simulation. We experimented with the number of clusters in order to find the optimum configuration, and discovered that usually between 3 to 10 clusters is optimal for the 20 network topologies we utilized.

3. Simulator

All these routing protocols were tested on a self written General Purpose Simulator. In this simulator we can regulate and monitor all the parameters of sensor nodes like sensing and transmission range, battery power, buffer limits, target generation, area to be monitored etc.