Simulation of LEACH Protocol and LEACH-C Protocol Using Contiki-NG

CSE257 Airi Kokuryo *University of California Santa Cruz* Winter 2024

***Abstract*—LEACH(Low-Energy Adaptive Clustering) protocol [1] was first proposed in the year 2000, as the first clustering-based protocol for wireless sensor networks (WSNs). Even after 24 years of existence, this protocol has increased the number of its variants. The paper will be focusing on the two protocols; the original LEACH protocol and the LEACH-C(LEACH Centralized) protocol [2] as its successor variant. Contiki-NG is a network simulator which allows simple implementation of the communication schemes, therefore will be used as the network simulation medium in this project. This paper also discusses on the different energy calculation from the original papers of the implemented protocols and its code structures. Finally, the paper concludes with possible research domains of the LEACH variations in the future.**

***Index Terms*—clustering, WSNs, LEACH, LEACH-C, energy efficiency**

I. INTRODUCTION

A Wireless Sensor Networks (WSNs) [3] is an infrastructure-less wireless network consisting of thousands of resource-constrained sensor nodes. The main goal of WSNs is to solve problems associated with resource management, scalability, and reliability. Therefore topology management is crucial in deciding the performance of the WSNs.

There are three main topology managements in WSNs. One is the direct communication with the Edge Gateway(or Base Station), another is the Minimum Energy Multi-hop Routing(MTE), and third is the clustering.

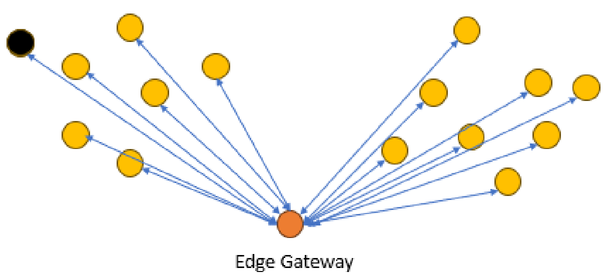

Fig. 1. Direct communication with the Edge Gateway

Fig. 1 shows the direct communication with the Edge Gateway, where sensor nodes have no intermediate nodes and only communicate directly. A large amount of transmission power is used, therefore the further the nodes are from the Edge Gateway, the more likely they are to run out of energy.

Fig. 2 shows the minimum energy multi-hop routing(MTE), where the nodes route through intermediate nodes to the Edge

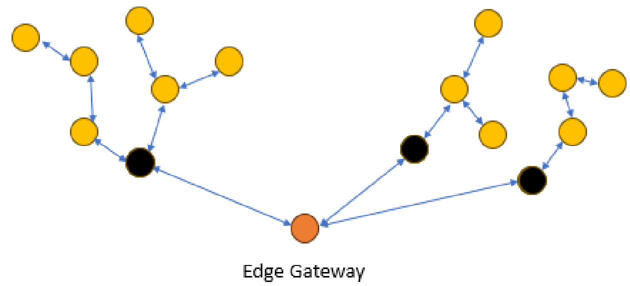

Fig. 2. Minimum energy multi-hop routing(MTE)

Gateway. The nodes that receive and send the most, which are the nodes closest to the Edge Gateway, are more likely to die out of energy compared to the tree leaves of routing.

The direct communication with the edge gateway is often compared to with the minimum energy multi-hop routing because they have the opposite trade-offs of whether the first node dies earlier or the last node lasts longer. Even though clustering, which will be introduced next, is a different topology management style that is more complicated, it could have the same trade-offs depending on its implementation. LEACH protocol that is more self sustained and is less reliant on the Edge Gateway could be seen as a derivative of the MTE, as LEACH-C protocol that is more dependent on the Edge Gateway and needs to send a uni-cast to the Edge Gateway often may be seen as a derivative of the direct communication topology management.

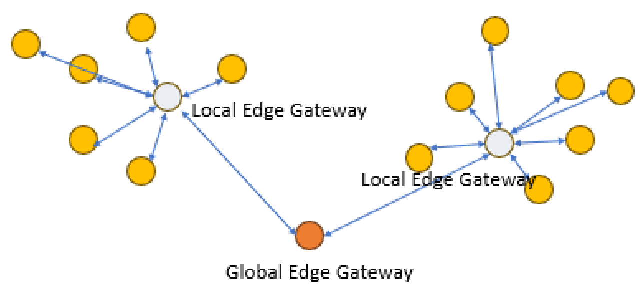

Fig. 3. Clustering

Fig. 3 shows the clustering topology management, where the nodes are organized into a group called clusters. The cluster nodes will have one or more CHs(Cluster Heads), and it will communicate with the global edge gateway.

In this design, the CHs must be in a high energy mode, as it uses the most energy within the cluster. Therefore CHs must be interchanged with other nodes at a certain frequency. LEACH solves this problem by changing the CH and the clusters every round consisting of the setup phase and the steady-state phase [1]. The main difference between LEACH and LEACH-C protocols are their CH selection methods. LEACH uses the distributed CH selection method, where each nodes will have the privilege to CH decision for themselves. Therefore it will not assure any fixed numbers of CHs and clusters. Comparing to, LEACH-C uses the centralized CH selection methods where the global edge gateway has the privilege to decide the CH among all nodes. This allows to always have a fixed number of CHs and clusters, leading to a fairer cluster allocation.

By simulating these two protocols using Contiki-NG from Cooja, the paper will analyze the improvements in energy efficiency of the model. Some modification to the LEACH-C protocol had to be made, since Contiki does not offer energy information of each node. Therefore in this paper, it will be using the energest API from Contiki, which provides the amount of time consumed in transmission, listening, and sleeping. This will also be used for calculating the difference of energy consumption between LEACH and LEACH-C.

II. LEACH: Low-Energy Adaptive Clustering Hierarchy

LEACH [1] is a clustering-based protocol, which use randomized CH selection to distribute a fair energy load throughout the wireless sensor networks. In LEACH, the CH selection occurs first, followed by the cluster formation where the CHs perform as the local edge gateways. In LEACH, the CHs will have the capability to fuse the received data to reduce the amount of data being sent to the global edge gateway. The data will be sent using the TDMA approach, which saves energy by sleeping during the slots in which they are not sending during the process.

A. LEACH: Setup Phase

In LEACH, each sensor node will elect themselves as a CH with a certain probability based on the number of rounds and whether it has become a CH within a certain number of rounds. The decision of the CH is made by the node, choosing a random number between 0 and 1. If the number is less than the calculated threshold, it will become a CH. The threshold is calculated as shown below:

$$\text{T(n)} = \begin{cases} \frac{P}{1-P\cdot(r \mod \frac{1}{P})} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases}$$

P is the desired percentage of the CH, and will be using 0.05 for its value in this paper. G is the set of nodes that have not become a CH in the past 1/P rounds, and r is the current round number. Here, we assume all nodes have the same amount of energy for each node at the beginning of the round 0.

The clusters that elected themselves as a CH will send out a broadcast advertisement message to the surrounding nodes. During this, the non CH nodes keep their receivers on to hear the advertisements. Upon receiving the messages, the non CH nodes will decide which clusters to join, depending on the signal strength of the received broadcasts. They will send out a uni-cast to those CHs to advertise themselves to become a part of the cluster. This uni-cast will be done using the CSMA MAC protocol.

The CHs will receive the uni-cast messages, and create a TDMA schedule telling each node when it can transmit. This schedule is broadcast back to the nodes in the cluster.

B. LEACH: Steady-State Phase

The non CH nodes will start sending the data using the received TDMA schedule, and by uni-casting the data to the CH. After the data transmission has finished, the CH will fuse its received data into one signal, and send this to the global edge gateway. The steady-state phase will require more time for data transmission compared to the setup phase.

III. LEACH Implementation With Contiki-NG

For a simple implementation of the LEACH protocol using Contiki, this project uses the NullNet function. NullNet is a network layer provided by Contiki-NG, where the packets for the network-layer are created directly by the application. Since the LEACH protocols assume homogeneous nodes that are one hop away from each other, it will be using a single code file to implement the LEACH protocol. As shown on Fig. 2, there will be a hundred sensor nodes and one edge gateway, which is the same as the simulation done in the original paper for LEACH. The possibility to become a CH, P will be set to 0.05 as well. The transmission range is set to 150m, so all nodes will have a one hop connection to all the other sensor nodes, including the global edge gateway. Further details about the implementations will be explained below.

A. Initialization

When implementing LEACH in Contiki, the timing of when each node send the data and receive was crucial. Therefore, the setup phase and the steady-state phase is broken into several thread processes. The processes are time critical, so its timing for when the processes will start and end will be handled by the etimers. There are different etimers for each processes for advertisement, listening for data, going through one round, and sending data. The same timers will be used for the global edge gateway to synchronize the communication period. Here is the list of processes that were used for implementing LEACH:

1. Main process
2. CH selection process
3. CH advertisement process
4. Non CH advertisement process
5. TDMA schedule creation process
6. TDMA data transmission process

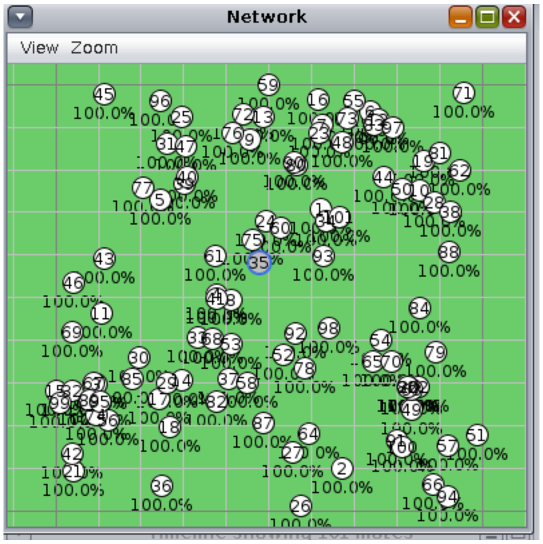

Fig. 4. LEACH nodes

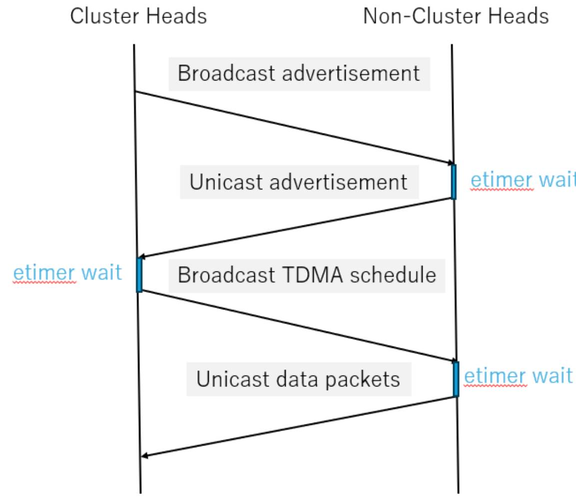

Fig. 6. Broadcast and Uni-cast by CHs and non-CHs

7. CH data fusion process
8. Free variable process

```
05:00.884   ID:13   [INFO: LEACH_Node] [0] Received packet: address 101
05:00.885   ID:59   [INFO: LEACH_Node] [0] Received packet: address 101
05:00.899   ID:3    [INFO: LEACH_Node] [0] Received packet: address 101
05:00.906   ID:21   [INFO: LEACH_Node] [0] Received packet: address 101
05:00.945   ID:23   [INFO: LEACH_Node] [0] Received packet: address 101
05:00.965   ID:100  [INFO: LEACH_Node] [0] Received packet: address 101
05:00.976   ID:57   [INFO: LEACH_Node] [0] Received packet: address 101
05:00.978   ID:32   [INFO: LEACH_Node] [0] Received packet: address 101
05:00.984   ID:91   [INFO: LEACH_Node] [0] Received packet: address 101
```

Fig. 5. Receiving advertisement from global edge gateway

```
1:45:00.491  ID:52  [INFO: LEACH_Node] [0] Cluster head process
1:45:00.491  ID:52  [INFO: LEACH_Node] random value: 0.050000 0.728847
1:45:00.491  ID:52  [INFO: LEACH_Node] [0] is_ch: 0
1:45:00.491  ID:52  [INFO: LEACH_Node] [0] Waiting for messages...
1:45:00.501  ID:99  [INFO: LEACH_Node] [0] Cluster head process
1:45:00.501  ID:99  [INFO: LEACH_Node] random value: 0.050000 0.020432
1:45:00.501  ID:99  [INFO: LEACH_Node] [0] is_ch: 1
1:45:00.501  ID:99  [INFO: LEACH_Node] [0] Sending out broadcast
```

Fig. 7. CH selection process

The main process will include all other processes listed above, which will perform differently depending on whether it is a CH or not after the CH selection process. Inside the main process and before any other processes are called, the global edge gateway will send out a broadcast message to all the nodes within the range to advertise them of the destination address. Fig. 3 shows how the nodes are receiving the packet from the 101th node, which is the global edge gateway. This will minimize the communication between the global edge gateway and the CHs, as the global edge gateway will only be sending one broadcast at the beginning of the main process, and will not repeat during any later rounds.

B. Setup Phase

During the setup phase, the CH selection process, the CH advertisement process, Non CH advertisement process, and the TDMA schedule creation processes are proceeded. During the CH selection process, every node will determine whether they have become the CH or not in the past 1/P rounds by keeping track of the latest round number that they have become a CH. If they have not, it will call the function where it generates a random number from 0 to 1, and if it is below threshold T(n), it will become a CH using the boolean is_ch. Fig. 5 shows how the is_ch = 0 are the non-CHs that will wait for the broadcast messages and is_ch = 1 are the CHs that sends out the broadcast during this process.

After deciding the CH, only the CHs will proceed the CH advertisement process where it sends out broadcast to their one hop neighbor. The non-CHs will check the RSSI of the received packets, and keep the link address of the CHs with the strongest RSSI. Here, only the non-CHs will receive the broadcast and only the CHs receive the uni-cast due to the structure it has in LEACH. Fig. 4 shows how CHs and Non-CHs will do either the broadcast or the uni-cast to send the messages. Receiving the broadcast from CH, the non-CHs sends back an advertisement to the strongest neighbor CH, as it can be seen on Fig. 6.

The uni-cast will be received by the CH, where it will use the TDMA Packet structure which holds the list of sensor node addresses inside the cluster and its maximum cluster size. CH will add the source address of the received uni-cast packet, and add 1 to the number of cluster size. When finishing receiving all uni-casts from the non-CH sensor nodes, CH will move onto the TDMA schedule creation process where it will send out the TDMA Packet structure using the broadcast. Non-CHs will only receive broadcast from their strongest neighbor CH during this process.

C. Steady-State Phase

During the steady-state phase, the actual data transmission which are not advertisements will occur. Therefore the TDMA uni-cast.png

Fig. 8. Non-CH advertisement process

```
2:00:00.690   ID:26   [INFO: LEACH_Node] [0] Sending TDMA schedule
2:00:00.690   ID:26   [INFO: LEACH_Node] [0] Sent out tdma schedule
2:10:00.002   ID:76   [INFO: LEACH_Node] [0] Sending with TDMA process
2:10:00.013   ID:69   [INFO: LEACH_Node] [0] Sending with TDMA process
2:10:00.026   ID:39   [INFO: LEACH_Node] [0] Sending with TDMA process
2:10:00.027   ID:34   [INFO: LEACH_Node] [0] Sending with TDMA process
2:10:00.035   ID:24   [INFO: LEACH_Node] [0] Sending with TDMA process
```

Fig. 9. TDMA processes

data transmission process and CH data fusion process is proceeded here. During the TDMA data transmission as shown on Fig. 7, each non-CH sensor nodes that has receives their own TDMA schedule from the strongest neighbor CH uses the TDMA schedule to send out data to the CHs. In order to send and receive data that are comparably larger than the advertisement messages which was only a single byte, it will generate an array of 10 random integers to send to the CH as the data. The reason why it is only 10 random integers, is because any data that are too large will not be sent by the nodes due to the system of Contiki. Receiving the data from all the cluster nodes, CH will fuse the data by calculating the mean of the received data, and sends this to the global edge gateway as shown on Fig. 8.

After all the processes has been proceeded is over, free variable process is executed to ensure that the last data was being sent. During this process, both the CHs and the non-CHs will reset the parameters they have used for the current round. Then it will go onto the next round by increasing the count and resetting the etimers.

D. LEACH Energy Calculation

After all the rounds have completed, each sensor node will calculate each Radio listen time, transmit time, and radio-

```
2:20:00.690   ID:26    [INFO: LEACH_Node] [0] Sending fused data
2:20:00.717   ID:101   [INFO: App       ] Received data from cluster head: 26
```

Fig. 10. CH data fusion process

```
5:56:40.073   ID:8    [INFO: LEACH_Node] [2] Round_ch: 2
5:56:40.073   ID:8    [INFO: LEACH_Node] [2] Freeing cluster neighbor information
5:56:40.073   ID:8
5:56:40.073   ID:8    Energest:
5:56:40.073   ID:8     CPU          18300000000s LPM      0s DEEP LPM 0s  Total time 18300000000s
5:56:40.073   ID:8     Radio LISTEN 13799906800s TRANSMIT 72280s OFF      4500020920s
5:56:40.079   ID:67   [INFO: LEACH_Node] [2] Round_ch: -20
5:56:40.079   ID:67   [INFO: LEACH_Node] [2] Freeing cluster neighbor information
5:56:40.079   ID:67
5:56:40.079   ID:67   Energest:
5:56:40.079   ID:67    CPU          18300000000s LPM      0s DEEP LPM 0s  Total time 18300000000s
5:56:40.079   ID:67    Radio LISTEN 12000022592s TRANSMIT 9112s OFF      6299968296s
```

Fig. 11. Energest calculation

off time. Here, it assumes that the less time listening to the radio and transmitting, the better energy consumption is. Fig. 9 shows the energest sum seconds after round 2. Round_ch is the round where it was last a CH, so this shows that the node ID:8 was a CH during this round 2. Comparing this to the node with ID: 87, this node has only been a non-CH node, and its transmit time is less 1/8 and its sleep time is longer than the node that has been a CH once. This indicates that being a CH consume more energy than the non-CH.

IV. LEACH-C: LEACH Centralized

LEACH-C [2] is one of the LEACH successor variant where it scatters the CH throughout the sensor networks to create a well established clusters. This solves the problem of LEACH not having a guaranteed placement or number of clusters in one round. As this may cause issues such as where one round may have absolutely no cluster heads for the formation of clusters, LEACH-C performs well in having a guaranteed number of clusters every round. It uses the centralized CH selection method, therefore has a setup phase that is completely executed at the global edge gateway. Edge gateway will use the information of how much energy each sensor node has to determine its possibility of becoming a CH. The steady-state phase where the data transmission occurs, is the same as the original LEACH protocol, therefore will be using the same code for implementation.

A. Setup Phase

The setup phase of LEACH-C starts with each node sending information about its current location using a GPS receiver and its energy level to the global edge gateway. Edge gateway will then calculate the average of the node energy, thus the nodes with lower energy than the average will not be participating in the CH selection. The edge gateway will then decide the clusters with annealing algorithm [4] to find the k optimal clusters [5]. Upon deciding on the CH and its corresponding clusters, the global edge gateway broadcasts a message with the cluster head ID, otherwise determine itself as a non-CH and calculates its TDMA slot for data transmission.

B. Steady-state Phase

The steady-state phase of LEACH-C is the same as LEACH. Here, the data is sent by the non-CHs using the TDMA slots determined by themselves during the setup phase. The data received by the CHs will be fused during this process, then is sent to the global edge gateway.

V. LEACH-C Implementation With Contiki-NG

As LEACH-C requires energy calculation in deciding its clusters, this was not possible using Contiki at this point as the data on energy left in nodes could not be found. Therefore, in this project, the amount of energest time consumed will be used for determining the energy left in nodes and energy consumption. As a complete implementation of the LEACH-C protocol was not possible within time, I would like to introduce the ways LEACH-C could be implemented in Contiki and

Here is the list of processes that were used for implementing LEACH-C:

1. Main process
2. Node energy and GPS advertisement process
3. CH and cluster selection/broadcast process
4. TDMA data transmission process
5. CH data fusion process
6. Free variable process

All processes will be run inside the main process, using the same etimer as LEACH. The number of processes are less than LEACH as most of the calculation along with the CH and cluster selection is done within one global edge gateway. The number of sensor nodes, global edge gateway, and the transmission range will be the same as LEACH. One difference is that it will have a higher frequency of communication between the global edge gateway and the nodes, therefore the nodes will be sending more data to the edge gateway compared to LEACH. On the other hand, the nodes will be sending less data to eachother compared to LEACH, so finding out how LEACH-C performs compared to LEACH also indicates the pros and cons of performances between centralized and distributed CH selection methods.

Comparing Fig. 4 to Fig. 10, we can see that the Cluster Head assignment and the cluster formation take the same process. Therefore it saves the overall communication frequency of each node, however has a downside where all nodes need to send and receive a heavier data packet from the Edge Gateway.

A. Setup Phase

Setup phase of LEACH-C implementation in Contiki starts with the same global edge gateway advertisement used in implementing LEACH. This will advertise the surrounding nodes within the communication range (which here we assume is all nodes in this WSNs) its link address to send back a uni-cast including about their information. This advertisement is only done once, because the edge gateway is static, and should not be informing it every round.

After the edge gateway sends out the advertisement, nodes will move onto the Node energy and GPS advertisement process, where they will use a structure including energy and GPS information to send a uni-cast to the edge gateway. Upon

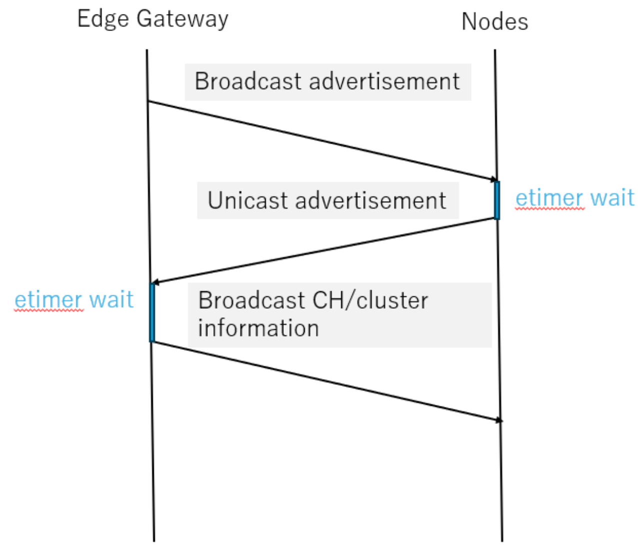

Fig. 12. Broadcast and uni-cast by Edge Gateway and Nodes

receiving the uni-cast from all nodes, the edge gateway will perform the CH and cluster selection/broadcast process where it calculates the mean average of the node energy.

With Contiki not supporting node energy, we will use the energest module. Therefore each node will send out the total energy used, which is the sum of time it listened and transmit(we will call this the time energy for short). This approach is different from the way it was proposed in the paper for LEACH-C. From the energy consumption by the simulation of LEACH using the same energest module, we found out that there are explicit difference within the transmission time and listen time when it is a CH. Therefore we can assume that the concept of using the large amount of energy node and using the nodes with small amount of transmission and listen time could be seen as nearly equivalent, when basing the energy calculation off the energest module.

After calculating the mean average of the received time energy, the edge gateway will then compare between each node to determine if they could become a cluster head or not. If the node has a smaller time energy than the average, it will be allowed to participate in the cluster head selection. As Contiki does not support the GPS information either, it may not be able to include the largest trade-off LEACH-C has, which is that each node needs to consume extra energy for the GPS. However, there are several factors we could use to calculate the distance without using the GPS, such as the RSSI value and the speed for receiving an acknowledgement towards an advertisement.

B. Steady-state Phase

The functions of the steady-state phase will be the same as the LEACH protocol, however in LEACH-C, the data being sent from the global edge gateway includes all the information about the clusters within the WSN. Therefore it will take more process to find which cluster it belongs to and which TDMA slot they have, comparing to the process in LEACH where they only receive the TDMA slots from the already known CH. The CHs will also need to go through every address to find out if they are a CH or not. Therefore this process will take a longer time than the LEACH protocol, but may have a shorter time consumption overall with having less advertisement phase.

C. LEACH-C Energy Calculation

The energy calculation will be the same as LEACH energy calculation implementation. It will use the energest module to find out the time it took to listen, transfer data, and sleep.

Discussion

In LEACH-C, more energy is being used by the global edge gateway by them sending and calculating more compared to LEACH. However, the communication between the non-CHs and the CHs is minimized by not having to discover the CH and cluster sensor nodes by themselves. This is a trade-off in centralized CH selection method used in LEACH-C to the distributed CH selection method used in LEACH. LEACH-C may seem better, as the edge gateway is usually more powerful than the other sensor nodes. However when each node needs to send out their GPS and energy information to the global edge gateway, it creates a high overhead because this must be sent out frequently when re-selecting clusters every round.

Conclusion

To improve the resource management, scalability and reliability of the WSNs, clustering was introduced with its dynamic implementations. LEACH was the first protocol proposed using the clustering topology, and this outperformed the prior topology management systems; the direct communication with the Edge Gateway and MTE.

The main goal of this project was to compare the performances of LEACH to one of its successor protocols. The energy efficiency was targeted the most, as energy efficiency could be said is the most important factor in WSNs where the sensor nodes must be self-sustainable and long-lived. There are many trade-offs between LEACH and LEACH-C that may affect their energy efficiency. LEACH ensures that the cluster heads are the nodes that require the lowest amount of transmission power, however LEACH-C ensures that all clusters will be evenly spread out across the nodes. LEACH, with its distributed CH selection, only the CH needs to send a data to the Edge Gateway once, which requires a larger amount of energy compared to the communication with the non-cluster sensor nodes. On the other hand, LEACH-C, with its centralized CH selection, it requires less communication overall, and all the calculations could be done in one single node which makes it much simple in terms of communication.

However when implementing a protocol using a network simulator, it is important to consider the features they have and do not have. There were issues with implementing the LEACH-C protocol as they did not support the node energy, along with the GPS information which was needed to implement its system for creating the perfect clusters. There are alternatives which may compensate these problem, such as the usage of RSSI and communication speed, and energest module for figuring out the node energy.

Future Work

I would like to keep working on the LEACH-C implementation, as there might be a way to figure out how to find the vector coordinate and use them for calculating the distance between each nodes. However this will not include the energy consumption they may have continuously by having the GPS, so it might require some adjustments in the energy usage.

In the LEACH implementation, there were problems where it is not guaranteed to have a fixed number of CHs in each round. This may lead to a situation where there are no CHs during a round, or there are too many CHs at once. This problem was fixed in the LEACH-C protocol, however by using the centralized CH selection method.

As a future work, there may be a LEACH variant using the distributed CH selection with the implementation of ideas from the Distance Vector Routing to figure out the best clusters.

References

[1] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient routing protocols for wireless microsensor networks," in Proc. 33rd Hawaii Int. Conf. System Sciences (HICSS), Maui, HI, Jan. 2000.

[2] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," IEEE Trans. Wireless Commun., vol. 1, no. 4, pp. 660–670, Oct. 2002.

[3] G. Pottie, "Wireless sensor networks," in Proc. Information Theory Workshop, San Diego, CA, June 1998, pp. 139–140.

[4] T. Murata and H. Ishibuchi, "Performance evaluation of genetic algorithms for flowshop scheduling problems," Proc. 1st IEEE Conf. Evolutionary Computation, vol. 2, pp. 812–817, June 1994.

[5] P. Agarwal and C. Procopiuc, "Exact and approximation algorithms for clustering," in Proc. 9th Annu. ACM-SIAM Symp. Discrete Algorithms, Baltimore, MD, Jan. 1999, pp. 658–667.