A Study of the LEACH Protocol for Wireless Sensor Networks

Weeks 3-4: Implementing LEACH

Jonathan Henderson
Mississippi Valley State University
Itta Bena, Mississippi 38941

1. Introduction

Low-Energy Adaptive Clustering Hierarchy (LEACH) [1] is a clustering based protocol to collect data from wireless network. In the network, hundreds and thousands of wireless sensors are dispersed that collects and transmit data. Also in these networks cluster heads are elected out of the sensors to transmit the data collected to base station. Also, with the each of the sensor nodes being inexpensive and simple, their power level is low cannot be replaced and because of this, each sensor must take its turn as being a cluster head to make the protocol energy efficient.

After researching on LEACH, it was time to create a program to simulate the actual protocol. Just to get a feel of how LEACH operates the network created was very simple. Although the amount of sensor nodes and percentage of cluster heads were in small numbers, the total number of rounds varied. The program created started with just generating nodes for the network to simulating data transmitting and receiving, but it does not have any specific transmission range.

2. Generation of Nodes

Getting started was probably the easiest part of the program, simple generating some random numbers to be x and y coordinates. To get truly random numbers, simply using time as the seed value was obvious. Since each sensor node was to have an x and a y coordinate the idea of using a structure for this was used. The structure members consisted of the node identification number as well as the x and y coordinates.

```
struct node
{
	int nodeID;
	float x, y;
};
```

To get an ID number and set of coordinates for each of the one hundred sensor nodes, the use of a For loop was used. Also each node and its information were stored in an array of the structure.

```
for(int i = 0; i < numOfNodes; i++)
{
	r_x = ((float)rand()/RAND_MAX)*xMAX;
	r_y = ((float)rand()/RAND_MAX)*yMAX;

	nodeInfo[i].nodeID = i + 1;
	nodeInfo[i].x = r_x;
	nodeInfo[i].y = r_y;
}
```

Also, the coordinates are coded not to exceed the network area which is 100m x 100m. The results of this coding are displayed with the next two figures. The first is the output of the program showing the node ID as well as the random generated x and y coordinates.

Node ID	X Coordinate	Y Coordinate
0	26.3833	91.2595
1	52.6688	13.7638
2	38.9264	57.9821
3	66.6951	16.1412
4	52.7238	42.9792
5	32.7158	44.4136
6	60.7959	3.96741
7	36.8297	76.69
8	68.0227	62.9383
9	76.4489	49.2569
10	17.954	57.973
11	17.478	68.8742
12	88.9309	52.7299
13	31.7576	74.8711
14	18.9184	75.3899
15	49.1379	97.9156
16	68.4561	8.96634
17	8.59096	12.3508
18	81.0236	4.87686
19	42.7381	28.9712
20	50.2762	41.6181
21	80.1141	25.0557
22	53.4806	0.439467
23	99.5972	58.6413
24	60.5274	79.8853
25	21.3752	63.8173
26	77.9229	21.0578
27	68.8009	73.9433
28	37.0952	65.5477
29	85.5007	69.6982
30	86.4437	17.7862

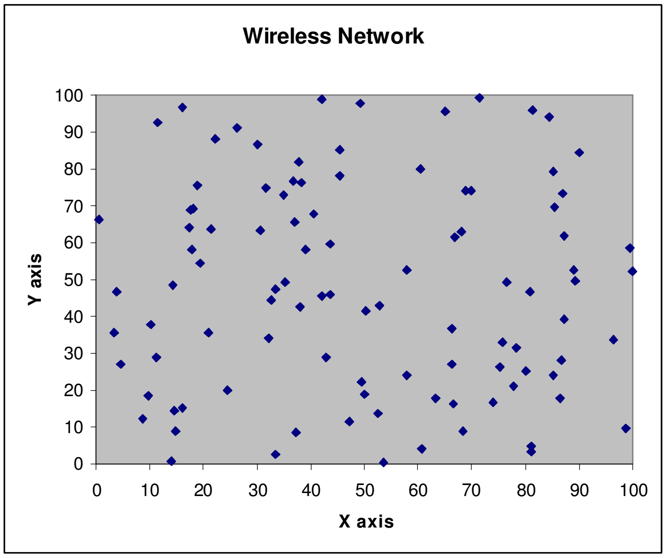

Figure 1. A graph view of the nodes randomly generated inside of the network with a range of 100m x 100m, using Microsoft Excel.

3. Selection of Cluster Heads

Creating clusters out of the nodes in the network with LEACH requires for a cluster head to be chosen. The cluster collects all of the data from its cluster and then compresses the data. The compress data is then sent to the base station and this will also be a high-energy transmission because of the distance of the base station. Each round a sensor node elects itself as a cluster head by selecting a random number to compare to the threshold value. The threshold *T(n)* is set as: *T(n)* = {P / 1 – P * (rmod1/P)} if n Є G, if not its 0. P is the desired percentage of cluster heads, r is the current round, and G is the set nodes that have not been cluster heads in the last 1/P rounds. Within the 1/P rounds, each node will have a chance to be the cluster head because of the threshold and after 1/P rounds each node has a chance to be a cluster head again.

Getting to the cluster head coding, another member was added to the structure. This member was a little character value to simulate that node has been used. Although the checker variable was included, it was an error in the program where a node would be selected as a cluster head again before the end of 1/P rounds. This error corrected by making the selecting code as a nested loop.

```
while(clusterCount < numCHs)
        {
            nodeCount = 0;
            while((clusterCount < numCHs) && nodeCount < (numOfNodes))
            {

                CHs[nodeCount].rNum = ((float)rand()/RAND_MAX);
```

After the error was corrected the following were results rendered from the coding. The first is 5% of cluster heads in a network of 100 nodes.

Round 1: 26 44 51 71 80
Round 2: 37 40 46 59 78

Round 3: 49 67 69 45 54
Round 4: 28 32 43 62 79
Round 5: 15 38 70 95 5
Round 6: 39 48 94 23 41
Round 7: 53 61 74 75 85
Round 8: 66 90 17 76 84
Round 9: 20 65 4 11 16
Round 10: 25 34 0 2 83
Round 11: 9 27 42 73 30
Round 12: 68 86 96 97 98
Round 13: 35 47 58 81 60
Round 14: 1 8 14 31 57
Round 15: 7 24 50 63 92
Round 16: 93 19 52 72 82
Round 17: 3 10 12 13 18
Round 18: 22 33 88 6 29
Round 19: 21 36 55 56 64
Round 20: 87 99 77 91 89

The next one is 10% of cluster heads in a network of 100 nodes.

Round 1: 7 14 26 41 44 49 50 53 71 73
Round 2: 6 11 15 17 22 34 40 51 68 72
Round 3: 2 4 10 12 24 52 60 77 81 86
Round 4: 28 30 32 37 47 63 64 65 84 92
Round 5: 1 9 66 69 82 83 89 0 8 13
Round 6: 21 23 25 29 31 36 39 43 59 74
Round 7: 5 27 35 45 48 56 62 67 70 75
Round 8: 16 20 38 42 54 57 76 78 80 88
Round 9: 3 18 19 33 46 55 58 61 79 85
Round 10: 97 99 87 90 95 98 96 94 93 91

After the cluster heads are chosen for the round, the sensor nodes calculate the distance between themselves and the cluster heads. The cluster head that is the closest or calls for the least amount of energy, is selected. After each cluster is finished, the cluster creates a schedule to collect each nodes data.

4. Transmission and Reception

Simulating the transmission and reception of data was not easy much of the coding had to be rearranged to have the correct data. Also, another member was added to the structure and this member was the amount of energy the sensor has. The initial amount of energy for each node was one joule, for this program. The cost of transmitting a message is:
$E_{TX}(k, d)=E_{elec} * k + \varepsilon_{amp} * k * d^2$. The cost of receiving a message is: $E_{RX}(k)=E_{elec} * k$.

The results of the program, it shows the round, the total energy lost, and the average energy lost per round.

Round #	Total Energy Consumption	Avg. Energy Consumption per Round
1	0.308478	0.308478
2	1.42768	0.713838
3	2.26208	0.754026

4	3.18469	0.796172
5	4.05291	0.810582
6	4.99641	0.832735
7	5.91044	0.844348
8	6.93677	0.867096
9	7.85366	0.872628
10	8.90056	0.890056
11	9.79771	0.890701
12	10.6676	0.888969
13	11.5932	0.891783
14	12.5856	0.898969
15	13.3981	0.893209
16	14.3452	0.896573
17	15.3733	0.904309
18	16.3215	0.906748
19	17.2418	0.907465
20	18.2201	0.911004
21	19.1815	0.913403
22	20.1201	0.91455
23	21.1384	0.919062
24	22.0232	0.917634
25	22.9484	0.917936
26	23.8186	0.916098
27	24.7566	0.916913
28	25.6816	0.917199
29	26.7342	0.92187
30	27.7567	0.925222

5. Conclusion

This program was created to simulate a clustering protocol called LEACH. This protocol is an energy-efficient way to monitor data in a wireless network. The program created was simulation of a network the ranged from 100m x 100m, but asked the user for the number of nodes to be dispersed into the network, as well as the number of rounds, and the energy amount. First, it started with generating just random numbers for coordinates to make the nodes and then adding the codes to for selecting cluster heads and then transmitting and receiving data to show how LEACH works.

6. Future Work

In the future, for this project, we would study the impact of transmission range on the connectivity level of the sensors to cluster heads. The transmission range of each sensor will be varied to study the connectivity level and we are expecting the connectivity will increase with the increase of cluster heads. The exact dependence of connectivity on transmission range and the percentage of cluster heads will be analyzed. Also, later, the network will be divided into a grid and each block of the grid will have its own cluster head and a set of sensor nodes. The LEACH protocol will be run on the grid-based network and the impact of grid block length on energy consumption and node lifetime will be studied.

References

[1] W. Heinzelman, A. Chandrakasan and H. Balakarishnan, "Energy-Efficient Communication Protocols for Wireless Microsensor Networks," *Proceedings of the Hawaaian International Conference on Systems Science*, January 2000.