EE5132 Wireless and Sensor Networks
EE5024 IoT Sensor Networks

Assignment

Simulation Study of Variations to LEACH Protocol (Variation A)

Group 05

Zhu Qi (A0224460N)

Ivander Jonathan Marella Waskito (A0200825R)

Emil Yong Kai Wen (A0169907M)

Tang Yuyi (A0232683A)

Chen Zuona (A0224670J)

Faisal Ahmad (A0034178E)

1 Introduction

This report studies the Low Energy Adaptive Clustering Hierarchy (LEACH) protocol's performance against other variations that improve the cluster selection process. LEACH is an effective low energy protocol that improves the lifetime of wireless sensor networks by forming of clusters with cluster heads (CH) aggregating and compressing data it receives from nodes within their clusters [1]. LEACH selects CH occurs in a distributed manner, based on the predetermined percentage of CH for the network and the number of times the node has been a CH before [2]. LEACH outperforms direct transmission and static clustering methods by requiring only a few CHs to transmit far distances and rotating these nodes to distribute the energy-intensive communication process [2]. Further variations are available in the CH selection process that makes them more energy-efficient than LEACH.

1.1 Literature Survey

Research papers that modified LEACH's cluster selection process were analysed. There were several variations proposed which can be categorised as distributed, centralised and hybrid clustering [3].

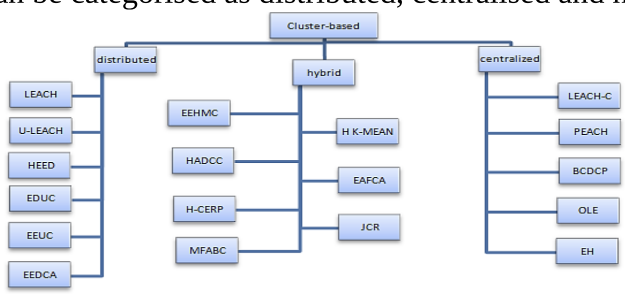


Figure 1 - Cluster-based routing protocols [3]

Distributed Clustering

Distributed clustering protocols include Unequal Clustering Scheme based LEACH (U-LEACH), Hybrid Energy-Efficient Distributed (HEED), Energy-Driven Unequal Clustering (EDUC), Energy Efficient Unequal Clustering (EEUC), Energy Efficient and Density Control Clustering Algorithm (EEDCA), Stable Election Protocol (SEP). These variations include various considerations in their CH selection such as energy vs distance ratio to have better energy performance than LEACH. Some of them lead to unequal cluster sizes e.g U-LEACH, EEUC and HEED.

Centralised Clustering

The centralised clustering approach uses the base station's global knowledge of nodes' energy levels and location to form optimal clusters, which are often of unequal sizes. The protocols include LEACH Centralised (LEACH-C), Power Efficient and Adaptive Clustering Hierarchy (PEACH), Base Station Controlled Dynamic Clustering Protocol (BCDCP), Optimized Lifetime Enhancement (OLE) [3]. LEACH-C is a centralised CH selection approach that considers the available energy levels and location of all the nodes in the network before CHs are selected. The base station (BS) will collect and process this information during the set-up phase and determine the cluster groupings and CH, i.e. centralised clustering. BCDCP forms balanced clusters (i.e. equal number of nodes in clusters), places CHs uniformly throughout the network, and uses CH-to-CH routing to transfer the data to the base station.

1.2 Proposed Approach

In this paper, we will first simulate LEACH in Part I to establish a baseline performance before going on to simulate and investigate LEACH-C and BCDCP in detail to compare their performance against LEACH.

2 PART I – Extension on LEACH

In Part 1, we are focusing on extending the LEACH algorithm given to compute the performance of LEACH when the network diameter is varied from 10 to 200 meters.

2.1 Simulation Results (Faisal: analysis, Programming: Emil & Zuona)

Figure 2 shows a trend of increasing number of the data packets received by the base station as network diameter decreases. For example, at 1500 rounds, a network with a diameter of 10 meters (solid red line) receives close to 15,000 data packets, almost twice as much as the 8,000 data packets received in a network diameter of 200 meters (dotted light blue line). This corresponds with the theoretical understanding that in a larger network with the same number of nodes, the node-to-CH and CH-to-BS distances are larger, thus leading to higher energy expended for each data packet sent i.e. energy dissipation is proportional to the distance squared. This effect is seen in Figure 3 where the number of data packets reaching the BS is lesser for any given amount of energy when the network is larger. This means that more energy is needed to send the same amount of data in larger networks attributed to the larger node-to-CH and CH-to-BS distances. This overall increase energy expenditure reduces the network lifespan as well and this is confirmed by Figure 4 which shows that the first onset of node death occurs much earlier for larger networks. For example, the first death occurs at less than 500 rounds for d=200m, twice faster as compared to the smallest network (d=10m) which has first death at 1000 rounds. We can see in Figure 5 that for larger networks, the higher energy usage and the resultant earlier first death leads to lesser overall data packets received at BS.

Figure 6 and Figure 7 further reiterate the overall higher energy consumption in larger networks. In Figure 6, the largest network (d=200m) shown by the dotted light blue line has higher total energy consumed at any given round as compared to the smaller networks. In Figure 7, we can see that the energy dissipation trend grows proportionally with network diameter as expected. This is supported by Figure 8 in [2]. Figure 8 shows the distribution of the number of rounds at which the networks of various diameters attain the specified percentages of node deaths. For example, the first death occurs earlier for larger networks (bars from the right) as compared to smaller networks (bars from the left) and at 50% of nodes dead, the trend continues but at larger rounds. We can also see that the variance between the lowest and higher number of rounds at each percentage of node death increases as the percentage increases. This means that the larger networks are more vulnerable to network death and LEACH seems to be work better at increasing the longevity of relatively smaller networks.

The implementation in Figure 7 is based on the accumulated cumulative energy shown on Figure 6. Nodes from different network sizes die at different rounds since energy dissipation is exponentially proportional to distance. Therefore, the plot on Figure 7 considers only a certain number of rounds before any nodes in all network size runs out of energy. This is achieved by registering the last round before first death occurs in each network size. From these registered rounds, the lowest round is chosen to get the accumulated dissipated energy for all network sizes. Normalization is performed by dividing each accumulated dissipation energy with the maximum dissipation's energy among all network sizes.

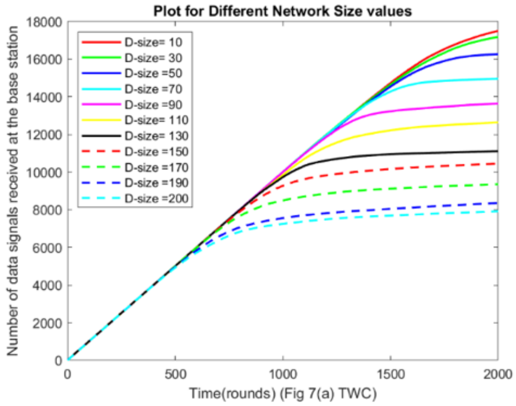


Figure 2

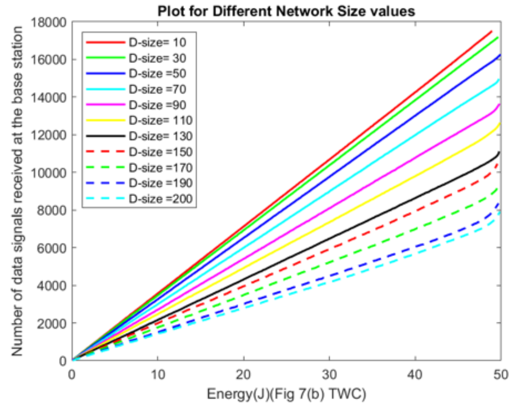


Figure 3

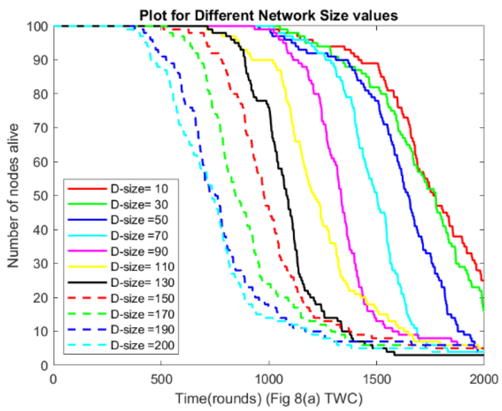


Figure 4

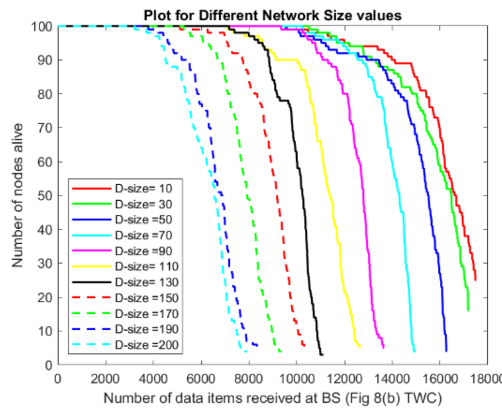


Figure 5

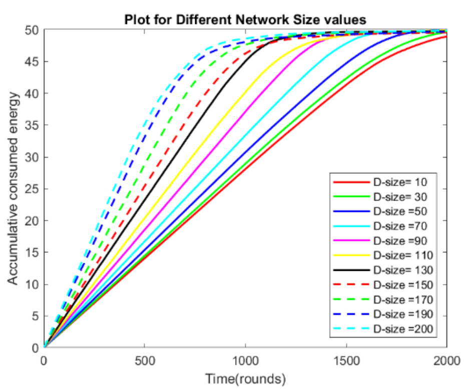


Figure 6

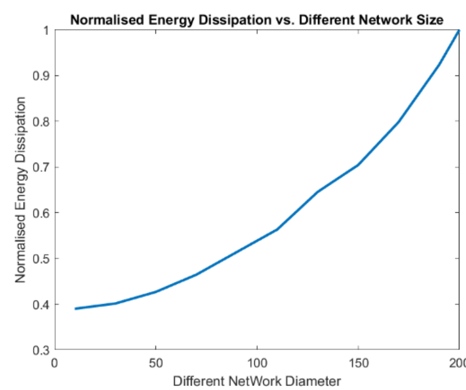


Figure 7

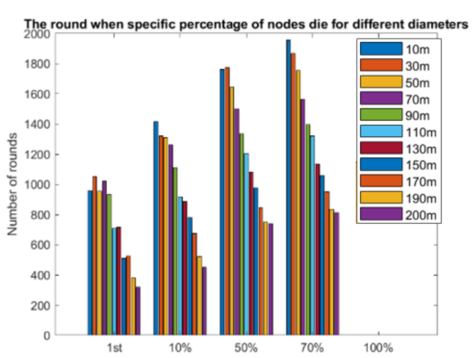


Figure 8

```
minIdx = 2001;
for i=1:length(container_for_plot)
    minIdx = min(container_for_plot(i).lastIdx, minIdx);
end
minIdx
```

Figure 9: Finding common round before all nodes die in any network size

3 PART II – LEACH-C

Before beginning on LEACH-C and BCDCP variations, we shall analyse on the nature of the equation which determines the optimal cluster number to decide on whether the base stations should be placed within or outside the network when running the further simulations.

3.1 Optimum Number of Clusters (Analysis & Programming: Emil & Zuona)

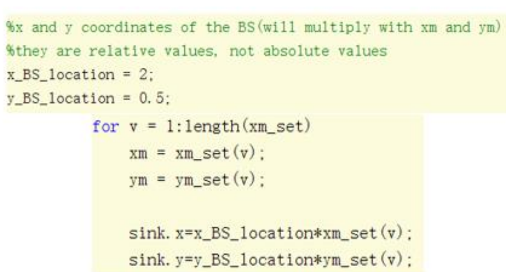
The theoretic optimum number of clusters is given as the equation below in [4], which is calculated by setting the derivative of total energy with respect to cluster number.

$$k_{opt} = \frac{\sqrt{N}}{\sqrt{2\pi}} \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}} \frac{M}{d_{toBS}^2} \quad \text{----- Equation 0}$$

, where N is the number of nodes, M is the network diameter, d_{toBS}^2 is the square distance between node and base station, $\varepsilon_{fs}$is the power amplifier of free space model, and ε_{mp} is the power amplifier of multipath model.

In the experiments, the common setting is N=200 nodes, ε_{mp}=0.0013, ε_{fs}=10. We can get maximum value and minimum value of d_{toBS} by iterating over nodes and calculating their distances to BS. After plugging in the maximum and minimum values of d_{toBS} into k_{opt}, we can get the minimum and maximum value of the optimum cluster number. If the maximum value of the optimum cluster number is larger than the number of total nodes, we set it as the number of total nodes, because it is impossible for the number of CHs to exceed total nodes. We also calculated an average distance, which corresponds to the average optimal k.

We tried different network diameters and different base station locations, to see whether the actual number of CHs was within the calculated range. The relative codes are as follows. We varied the relative values (not absolute values) of x and y axis for BS, to change BS's location.



```
%x and y coordinates of the BS(will multiply with xm and ym)
%they are relative values, not absolute values
x_BS_location = 2;
y_BS_location = 0.5;
        for v = 1:length(xm_set)
            xm = xm_set(v);
            ym = ym_set(v);

            sink.x=x_BS_location*xm_set(v);
            sink.y=y_BS_location*ym_set(v);
```

Figure 10

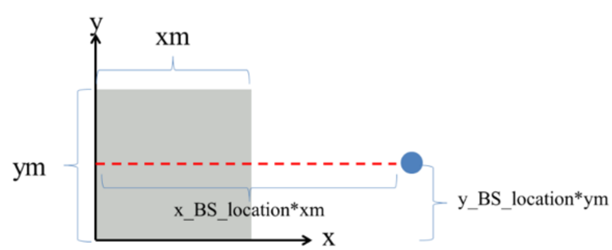


Figure 11

Simulating BS outside the network
x_BS_location = 1.35; y_BS_location = 0.5;

When the network diameters are between 70 and 200, the numbers of clusters for all rounds are roughly all within the theoretical optimum k range. We select some experimental results to show below. The yellow line corresponds to the actual number of CHs, which corresponds to the number of cluster as well. The blue line is the upper bound of the range, while the red line is the lower bound of the range. The green line is the average optimum k value calculated from average distance.

From these figures, we can see that:

- The numbers of CHs are mostly within the calculated optimal k range.
- For large network diameters (such as 200), many nodes die in later rounds because of long distance's energy consumption, which results in very few or zero CHs.
- The upper bound of range decreases, when diameter increases.

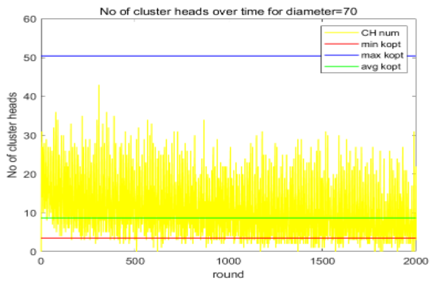


Figure 12

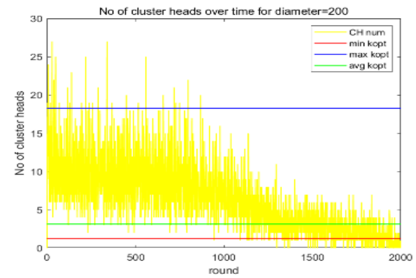


Figure 13

However, when network diameter is very small i.e. high node density, the theoretical optimum k range is larger than the actual number of clusters. Refer to the appendix for more experimental analysis using different positions of base station. In the appendix, experimental results and mathematic analysis both show that when BS within network and BS very far from network, actual CH numbers are not within the theoretical range. Hence, to align with Equation 1, the implementation and analysis of Leach C and BCDCP will consider the base station to be outside of the network at 1.5*xm , 1.5*ym with 4000 rounds to ensure k optimal clusters.

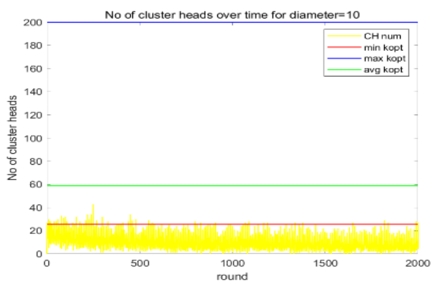


Figure 14

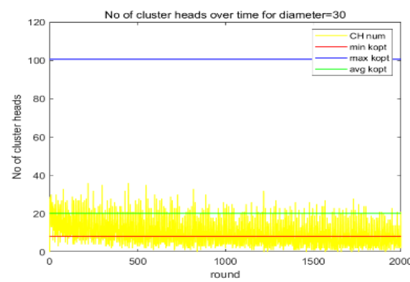


Figure 15

3.2 LEACH-C (Analysis & Programming: Emil & Zuona)

From Section 2, it is noticeable that LEACH is a prominent hierarchical cluster-based routing protocol for improving the longevity of the nodes in the wireless sensor networks. It groups sensor nodes into clusters to reduce energy dissipation. However, the selection of CHs in classical LEACH is distributed randomly. Since clusters are adaptive at every round, there is no guarantee that this protocol can ensure good clustering and optimal energy dissipation. This sparks an initiative to improve the clustering criteria in LEACH called, Low Energy Adaptive Clustering Hierarchy – Centralized (LEACH – C).

LEACH-C centralizes the cluster forming process at the base station using the global knowledge of the network to produce well-distributed clusters. In the set-up phase, the base station (BS) will need to know the location of all the nodes and their energy level at every round to determine the cluster grouping and CHs, i.e centralized clustering. Letting G be the set of nodes that have not been clusters head yet, all nodes in set G generate a random number from 0 to 1 at the beginning of 1/P rounds.

$$T(r) = \frac{P}{1-P*(r \bmod P^{-1})} \quad \textit{Where P is the cluser head probabilty} \ \forall nodes \in G$$ -- Equation 1

In order to decide the role of each node in each round, the random number corresponding to each node at the beginning of 1/P rounds will check against the threshold value calculated in Equation 1. If the random value is lesser than the threshold value, the node can be considered as a CH. In addition, the BS will take into account the total remaining energy from all available nodes in the network and calculates the average energy. Nodes that are below the threshold value and above the average energy are then deemed as CHs for Leach C. This method of clustering ensures that all nodes are given a chance to be a CH and the energy load is distributed evenly among all nodes since CHs have larger expenses of energy compared to normal nodes.

Figure 16 shows the programming code for CH formation. It will first check if the node is in set G. This corresponds to nodes that are alive and have not been considered as a CH in the previous round. The code highlighted in yellow in Figure 16 checks the respective node at each round if its random number is below the threshold and the remaining energy is above the average energy of all alive nodes.

```
if(S(i).E>0)
    temp_rand=rand;
    if ( (S(i).G)<=0)
        if (temp_rand<= (p/(1-p*mod(r,round(1/p)))))
            if (S(i).E>Eavg) % should be a CH
                countCHs=countCHs+1;
                packets_TO_BS=packets_TO_BS+1;
                PACKETS_TO_BS(r+1)=packets_TO_BS; %independent variable
                S(i).type='C';
                S(i).G=round(1/p)-1;
                C(cluster).xd=S(i).xd;
                C(cluster).yd=S(i).yd;
                distance=sqrt( (S(i).xd-(S(n+1).xd) )^2 + (S(i).yd-(S(n+1).yd) )^2 );
                C(cluster).distance=distance;
                C(cluster).id=i;
                X(cluster)=S(i).xd;
                Y(cluster)=S(i).yd;
                cluster=cluster+1;
```

Figure 16: Cluster head formation (code)

The CH takes up the role of an intermediate node for nodes that are far away from the base station. It performs local preprocessing from all member nodes in the cluster it governs before forwarding the data to the base station. Therefore, the minimum number of packets the base station is going to receive for that round will be the same as the number of CHs. In Leach C, the algorithm uses a simple radio hardware energy dissipation model where the energy dissipation is heavily dependent on the distance and the power amplifier. Leach C considers two different power amplifier models: free space and multipath model. The selection of which power amplifier model to use is based on the threshold value, d_0. If the distance is less than d_0, a free space amplifier is used, otherwise, the multipath model is utilized. The equation of transmitting k bits of message can be expressed in the equation below.

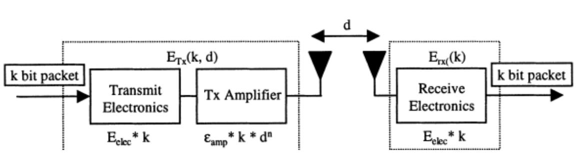


Figure 17: Simple radio model

$$E_{Tx}(k,d) = E_{Txelec}(k) + E_{Txamp}(k,d) \qquad \text{- Equation 2}$$

$$E_{Tx}(k,d) = kE_{elec} + k\varepsilon_{fs}d^2 \qquad if\ d < d_0$$

$$E_{Tx}(k,d) = kE_{elec} + k\varepsilon_{mp}d^4 \qquad if\ d \geq d_0$$

$$d_0 = sqrt(\frac{\varepsilon_{fs}}{\varepsilon_{mp}})$$

Besides dissipating energy when transmitting from CH to the base station, each CH also dissipates energy when it receives information from cluster members nodes. The received dissipation energy is as follows:

$$E_{Tx}(k,d)=kE_{elec.} \quad \text{----- Equation 3}$$

As shown in above equations, the energy dissipation of each node is exponentially dependent on the distance from where it transmits to the destination. Hence, to increase the longevity of the node, each node aims to send its data packets to the base station using the minimum distance it could find. If the distance to the base station is nearer than any CHs, the node will send its packet directly to the base station rather than using a CH as an intermediate node since it is more energy efficient. On the other hand, if the nodes are very far away from the base station, it will be more efficient to send its data packet to the CH to relay to the base station. CHs during transmission phase will collect data from nodes within their respective clusters and apply data fusion before forwarding directly to the base station. This pre-process the data from the cluster and reduces the complexity and computational load for the base station. However, this also means that larger network size will incur higher energy expense for cluster head since distance is larger. This is shown in Figure 19 that cluster head dies earlier in larger network size as compared to smaller network size.

In this algorithm, the distance metric used between two nodes is the sum of square distance.

```
for i=1:1:n
    if ( S(i).type=='N' && S(i).E>0 ) % if it is a node OR alive
        if(cluster-1>=1)
            min_dis=sqrt( (S(i).xd-S(n+1).xd)^2 + (S(i).yd-S(n+1).yd)^2 );
            min_dis_cluster=1;
            for c=1:1:cluster-1
                temp=min(min_dis,sqrt( (S(i).xd-C(c).xd)^2 + (S(i).yd-C(c).yd)^2 ) );
                if ( temp<min_dis )
                    min_dis=temp;
                    min_dis_cluster=c;
                end
```

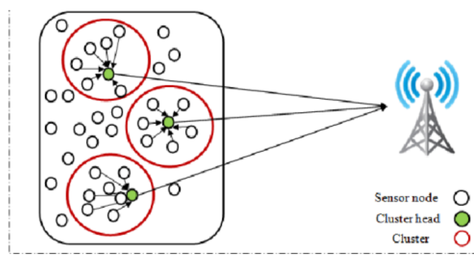


Figure 18: Cluster formation (code)

Since the ultimate objective is to send the data packet from each node to the base station with minimum distance, the programming code shown in Figure 18 & 20 first checks its distance to the base station. After which, it will compare its distance with all other cluster nodes. If the node is closer to the base station, the data packet is directly forward to the base station and does not rely on any CHs as a relay. This method prevents redundant energy dissipation on the chosen CHs for that round. This is one of the key differences in terms of programming between the implemented Leach C code by the authors versus the skeleton code provided.

In summary, Leach C algorithm factors in the remaining energy of each node for CH selection. This evenly distributes the energy load among all nodes since CH dissipates energy on two occasions; transmitting energy to the base station and receiving energy from cluster members. Thus, guaranteeing the longevity of the nodes.

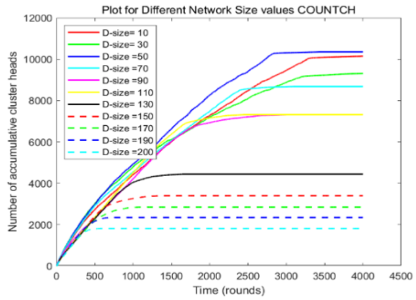


Figure 19 : Cluster head counts for all network sizes

```
else % Closer to the BS than all other cluster head. Send to BS straight
    min_dis;
    if (min_dis>do)
        S(i).E=S(i).E- ( ETX*(pkt_size) + Emp*pkt_size*( min_dis * min_dis * min_dis * min_dis));
    end
    if (min_dis<=do)
        S(i).E=S(i).E- ( ETX*(pkt_size) + Efs*pkt_size*( min_dis * min_dis));
    end
    packets_TO_BS=packets_TO_BS+1;    % send pkt to BS
```

Figure 20(a): Sending data packet directly to the Base Station since node is closer (code)

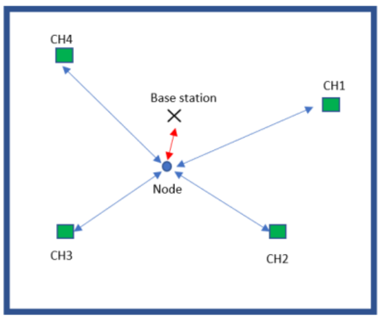


Figure 20(b): Node nearer to BS

```
% Because CH is the intermediate node to the BS
    % Data received from other nodes also will have
    % energy dissapation.
    % Data Aggregation etc
S(C(min_dis_cluster).id).E = S(C(min_dis_cluster).id).E- ( (ERX + EDA)*pkt_size );
packets_TO_CH=packets_TO_CH+1;
```

Figure 21: Energy dissipation when cluster head receives data packet (code)

3.3 Simulation Results (Analysis & Programming: Emil & Zuona)

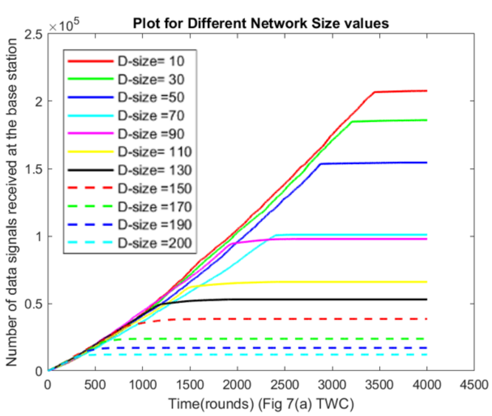


Figure 22(a): Data received by BS vs Rounds

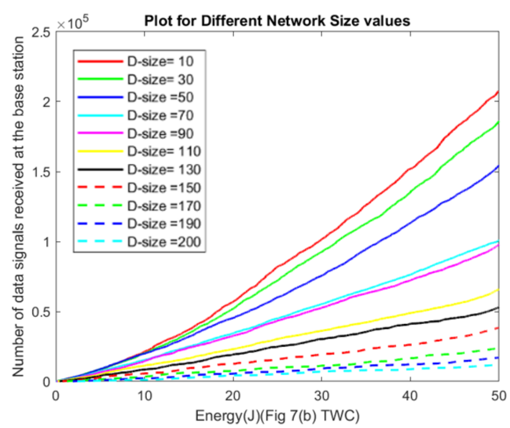


Figure 22(b): Data received by BS vs Energy dissipated

Figure 22 to 24 corresponds to Figure 7 and 8 in [4], showing the trend of the number of the data packets received by the base station being significantly higher when network diameter is smaller. For example, at 4000 rounds in Fig 22(a), a network with a diameter of 10 meters (solid red line) receives 2.1×10^5 data packets, almost four times as much as 0.5×10^5 data packets received in a network diameter of 130 meters.

This phenomenon is due to the nodes being closer to the base station when the network diameter is small, where there is a higher probability of nodes sending directly to BS. Therefore, base station will receive more data packets from individual nodes. Also, nodes generally lives longer in smaller network since distance is shorter. However, in larger network sizes, nodes generally send their data packet to a CH to relay information to the base station. Since the CH performs aggregation and compacts the information before sending it to the base station, the amount of data packets the base station receives will be lesser but more concise.

This also means that the base station for a larger network will require lesser computation for the same amount of nodes that are randomly placed, since the CHs have already performed some preprocessing. In addition, another reason as to why the number of data packets received by the base station is lower at round 4000 might be due to the number of CHs available to forward data packets to the base station. With a larger network, the rate at which the CH dies are significantly higher as well.

Figure 26 reinforced this idea. The plot shows the dissipated energy versus the network size. With the larger network size, the amount of dissipated energy required to send data packets to the base station is significantly larger as compared to a smaller network. These simulation results are aligned with Equation 2. The larger the network size, the further the nodes are required to send the data packet which in return results in higher energy dissipation.

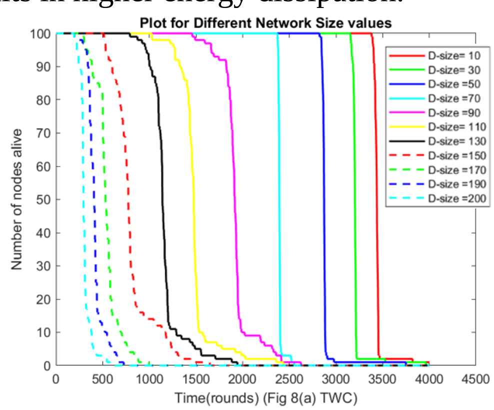


Figure 23: Nodes alive vs Rounds

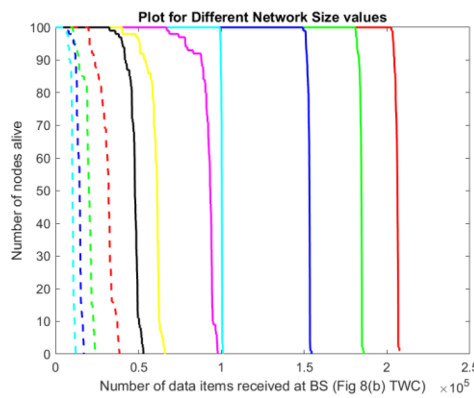


Figure 24: Nodes alive vs data received by BS

The advantage of Leach C algorithm is the efficiency of energy dissipation to ensure longevity of each node. Figure 23 shows the plot between number of nodes alive versus each round and Figure 24 shows the plot between the number of nodes alive versus the number of data item received at the BS. Since energy dissipation is directly proportional to distance, the plots shows that number of nodes alive decreases at larger network size. At 170 to 200 meters network diameter, the number of alive nodes drop drastically from 100 to close to 0 in a few rounds. However, from the plots we can also see the effectiveness of leach C; the first occurrence of the dead is very late as compared to leach. Taking 10m as an example, leach C first death occurs at 3500 rounds, but leach occurs at 2100 round. In addition, the drastic drop in alive nodes in leach C as compared to leach shows its effectiveness. Nodes dying almost together implies that the workload in leach C is evenly distributed thus ensuring fairness of workload which results in longer lifetime.

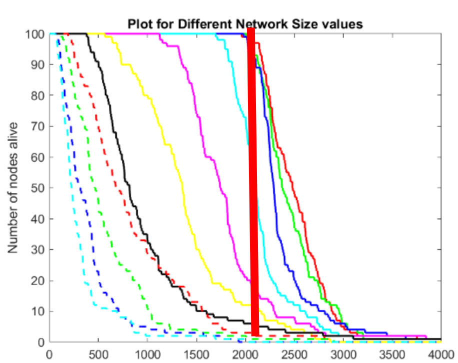


Figure 25: Leach.m with Leach C parameters

Figure 27 shows the occurrence where the network encounters its first death, 10% ,50%,70% and 100% of the nodes. It also shows a trend where the first dead node and all dead nodes happens in a span of few rounds -height of the bar charts are almost the same. Furthermore, it also shows that nodes die faster in larger network size.

Figure 26 shows the normalised energy dissipation against different network size. Similarly, it shows an upward trend where energy dissipation for nodes exponentially increases as network grow larger since energy dissipation is power factor with distance. At smaller diameter size, 20 to 50 meters the energy dissipation is gradual unlike at larger network diameter > 140 meters. This phenomenon can be due to the energy model used where the threshold, d0 determine which energy dissipation model; free space or multipath mode. Distance in larger network diameter usually use multipath energy dissipation model which explains the quadratic exponential in Figure 26. In summary, the greater the network size, the larger the energy dissipate by the nodes which lead to rapid death.

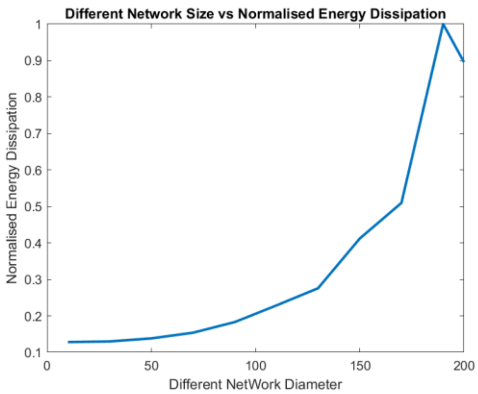


Figure 26: Energy dissipation vs network size

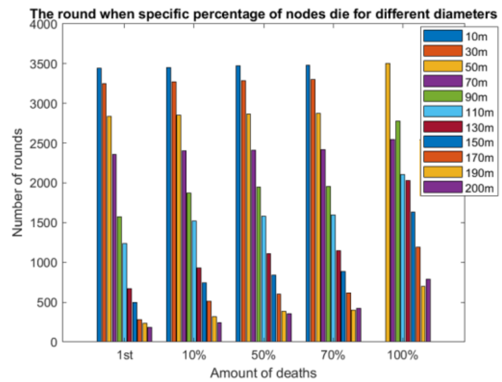


Figure 27: Different % of death for different network sizes

4 PART II - BCDCP

4.1 BCDCP – Programming and analysis (Programming: Yuyi, Ivander & Zhu Qi)

To improve LEACH and LEACH-C, a new centralised clustering-based routing protocol for Wireless Sensor Network (WSN) named Base Station Controlled Dynamic Clustering Protocol (BCDCP) is introduced in [1], which uses a high-energy BS to set up CHs within the network that is like LEACH-C. However, unlike LEACH and LEACH-C, instead of using a probabilistic algorithm to select its CH, BCDCP uses a clustering algorithm based on the coordinate location of the sensor nodes.

BCDCP emphasises the formation of balanced cluster sizes to further ensure each CH serves an equal number of member nodes to avoid overloaded CHs. In addition, with the aid of Minimum Spanning Tree (MST) [5] approach, the lowest-energy cluster-head-to-cluster-head (CH-to-CH) multi-hop routing path can be formed. Random choice of the Final CH to collect information from peer CHs then transmit to the BS will also help with minimising energy dissipation and distribute the burden of routing evenly among all clusters.

BCDCP operates in two major phases, which are the Setup Phase and the Data Communication Phase. Detailed operations in BCDCP including network and radio model setup will also be illustrated in the following sections.

A. The Setup Phase

The Setup Phase is crucial to the whole BCDCP algorithm. There are three key aspects in this phase which include Randomized CH Selection, Cluster Splitting, and CH-to-CH Routing Path Formation.

1. Randomized CH Selection: In the setup phase, the BS will first collect information on the current energy status from all nodes within the WSN to filter nodes that have above average energy to be cluster head candidates and group them into a set, e.g. *temp_S*. To ensure only nodes with sufficient energy can prolong the lifetime by performing tasks that require low energy costs.

Figure 28 indicates the programming code for selecting nodes above average energy. Firstly, the average energy *E_avg* is calculated based on the node energy status after the last round transmission. Secondly, a new set *temp_S[]* is created to store all nodes that contain energy above *E_avg* and potentially to be selected as CHs.

2. Cluster Splitting: To select proper CHs from *temp_S* and group the other nodes into smaller sub-clusters. Since equal cluster size is a priority in BCDCP, an iterative cluster splitting algorithm until the predetermined *Nch* number of clusters is reached. The optimal Nch should be log_2n, where n is the number of nodes within the WSN. Detailed procedures and code can be summarized in the following five steps:

1. Searching the largest cluster in the networks, which should be *temp_S* at the first round.
2. Find two nodes in this cluster with maximum separation distance, marking them as ch_1 and ch_2.
3. Group each of the remaining nodes in the current cluster with either ch_1 or ch_2 based on shorter Node-to-CH distance.
4. Balance the two newly created sub-clusters so that they have the same number of nodes.
5. Repeat the above steps of cluster splitting until *Nch* clusters are attained.

```
% Select nodes that are above average energy
E_sum = 0;
for i=1:1:n
    if S(i).E > 0
        E_sum = E_sum + S(i).E;
    end
end
E_avg = E_sum / n;
STATISTICS.Total_energy(r)=E_sum;
STATISTICS.Avg_energy(r)=E_avg;

temp_S = [];
for i=1:1:n
    if S(i).E > 0 && S(i).E - E_avg > - E_avg * 0.000000001
        temp_S = [temp_S S(i)];
    end
end
```

Figure 28: Select nodes above average energy(bcdcp.m line186)

```
% find largest distance
for i=1:1:length(c)-1
    for k=i+1:1:length(c)
        %dist_temp = sqrt((c(i
        dist_temp = DM(i,k);
        if dist_temp > dist
            dist = dist_temp;
            ch1 = c(i);
            ch2 = c(k);
        end
    end
end
```

Figure 29 - Find two candidate nodes maximum separation dstance(bcdcp.m line630)

```
function DM = createDistanceMatrix(S)
    DM = zeros(length(S), length(S))
    for i=1:1:length(S)-1
        for k=i:1:length(S)
            dist_temp = sqrt((S(i).xd - S(k).xd)^2 + (S(i).yd - S(k).yd
            DM(i,k) = dist_temp;
            DM(k,i) = dist_temp;
        end
    end
end
```

Figure 30 - Find two candidate nodes with largest distance(bcdcp.m line612)

The clustering algorithm will terminate once the number of clusters reaches Nch; before that, the algorithm will split the current largest cluster that exists and re-elect cluster heads. In our implementation as shown in Figure 31 below, clusters are stored in the array *clusters*, and cluster heads are stored in the array *cluster_heads*. On each iteration, the variable *size* and *j* will store the number of nodes and the index of the cluster in *clusters* with the largest number of nodes. For a special case where the largest cluster only contains a single node or no nodes, the loop will break as no more cluster splitting is possible. Else, the largest cluster will be split into two clusters, *c1* and *c2*, with *ch1* and *ch2* respectively. The previously largest cluster is replaced by *c1* and *c2* will be appended to the array of clusters. The full code of twoClustering function is shown in Figure 32 and Figure 33.

3. CH-to-CH Routing Path Formation: BCDCP uses a CH-to-CH multi-hop routing scheme to transfer the sensed data to the BS. Once clusters and CHs have been identified, BCDCP will randomly select a CH as the Final CH that collects data packets from peer CHs and sends them to the BS. The lowest-energy routing path can be confirmed with the MST approach *'[T, pred]=minspantree(G)'* as indicated in

```
% Balanced Iterative Clustering
clusters = {temp_S};
cluster_heads = {randsample(temp_S,1)};
nch = p * n;
m_size = n / nch;
while length(clusters) < nch
    % select largest cluster and split it to two
    j = 1;
    size = 0;
    for k=1:1:length(clusters)
        if length(clusters{k}) > size
            j = k;
            size = length(clusters{k});
        end
    end

    if size < 2
        m_size = size;
        break
    end

    [c1, ch1, c2, ch2] = twoClustering(clusters{j}, DM);
    clusters{j} = c1;
    cluster_heads{j} = ch1;
    clusters = [clusters {c2}];
    cluster_heads = [cluster_heads {ch2}];
end
```

Figure 31 – Clustering Algorithm (bcdcp.m line 261)

```
function [c1,ch1,c2,ch2] = twoClustering(c, DM)

    dist = -1;
    ch1 = -1;
    ch2 = -1;
    % find largest distance
    for i=1:1:length(c)-1
        for k=i+1:1:length(c)

            dist_temp = DM(i,k);
            if dist_temp > dist
                dist = dist_temp;
                ch1 = c(i);
                ch2 = c(k);
            end
        end
    end

    c1 = [];
    c2 = [];

    % group nodes to the two clusters
    for i=1:1:length(c)
        d_ch1 = DM(c(i).id, ch1.id);
        d_ch2 = DM(c(i).id, ch2.id);

        if d_ch1 < d_ch2
            c1 = [c(i) c1];

        else
            c2 = [c(i) c2];

        end
    end
```

Figure 32 – Clustering Algorithm (bcdcp.m line 624)

```
    % fix imbalances
    diff = length(c1) - length(c2);
    if diff ~= 0

        % if |c1| > |c2| find nodes in c1 closest to c2 then move them
        if diff > 0
            tomove = floor(diff / 2);
            dists = [];
            for i=1:1:length(c1)

                dists = [dists, DM(c1(i).id, ch2.id)];
            end
            dists;
            [X,idx] = sort(dists);

            c2 = [c1(idx(1:tomove)) c2];
            c1 = c1(idx(tomove+1:length(idx)));

        % if |c1| < |c2| find nodes in c2 closest to c1 then move them
        elseif diff < 0
            tomove = - floor(diff / 2);
            dists = [];
            for i=1:1:length(c2)

                dists = [dists, DM(c2(i).id, ch1.id)];
            end
            dists;
            [X,idx] = sort(dists);

            c1 = [c2(idx(1:tomove)) c1];
            c2 = c2(idx(tomove+1:length(idx)));
        end
    end

end
```

Figure 33 – Clustering Algorithm (bcdcp.m line 659)

Figure 34 which ensures the minimum energy consumption for each CH while performing the data forwarding. A visualization of the created MST with Nch = 16 is shown in Figure 35. This method can significantly prolong the lifetime of nodes since it is assumed that the BS is far away from the clusters and sending messages to BS is an energy-intensive task. Upon the routing path is found in each epoch, a CH will be randomly picked as the *main_ch* (Figure 36) to be put as the final CH that aggregates data packets and send to the BS following the routing path *ch_path* .

```
chs = length(cluster_heads);
s = [];
t = [];
w = zeros(chs*(chs + 1)/2);
edge_count = 1;
% Minimum Spanning Tree Routing
for i=1:1:length(cluster_heads)-1
    for k=i+1:1:length(cluster_heads)
        s = [s string(cluster_heads{i}.id)];
        t = [t string(cluster_heads{k}.id)];
        w(edge_count) = DM(cluster_heads{i}.id, cluster_heads{k}.id);
        edge_count = edge_count+1;
    end
end
edge_count = edge_count-1;
G = graph(s(1:edge_count), t(1:edge_count), w(1:edge_count));
[T, pred] = minspantree(G);
```

Figure 34 – Generation of Minimum Spanning Tree (bcdcp.m line 286)

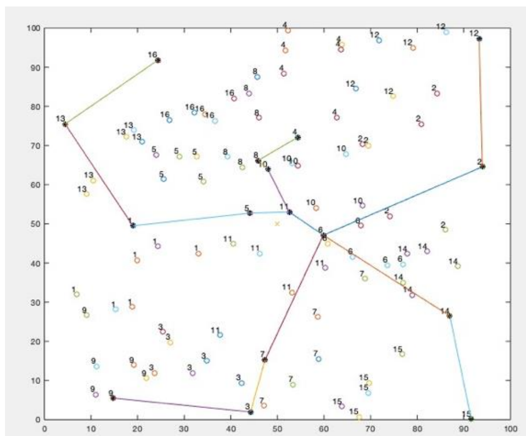


Figure 35 – Clusters and Spanning tree Visualization of BCDCP, with *Nch=16*

```
% random pick of main cluster head sending to Base Station
main_ch = randsample(cluster_heads,1);

  % ===== ch-to-ch routing =====
  ch_path = shortestpath(T, string(ch), string(main_ch));
```

Figure 36 – Routing Paths (bcdcp.m line 322 and 367)

4. Schedule Creation: The BCDCP protocol utilizes Time-Division Multiple Access (TDMA) scheduling scheme to minimize collisions between member nodes trying to transmit data to the CH. An Interim

Schedule Creation ID (SCID) is created for all nodes to avoid the collision. This is the last major issue related to the BCDCP setup phase. However, in our simulation, we will assume there will be no collision in data transmission by default.

Radio model: To compare different algorithms and keep consistency, we use the same sensor node radio model as discussed in LEACH-C Equation 2 and Equation 3. Energy cost parameter of the wireless sensor node is indicated in Figure 17. A threshold of distances (d0) between nodes is set to determine which model to choose the energy required by the transmit amplifier $E_{amp}(r)$.

4.2 Simulation Results - (Ivander, Yuyi & Zhu Qi: Analysis)

To evaluate the algorithm performance, we simulated BCDCP using Matlab. Figure 37 to Figure 42 indicated the performance of the implemented BCDCP protocol. As mentioned in Chapter 2, we are focusing on evaluating the total system energy dissipation changes by varying the network size. The simulation parameter is chosen with initial energy E0=0.5 Joules, number of nodes within the WSN n=100, number of simulation epoch r=4000 rounds, with the BS located at (1.5 xm, 1.5 ym).

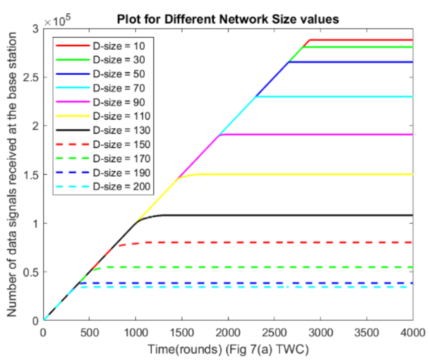


Figure 37: Data packets received over time

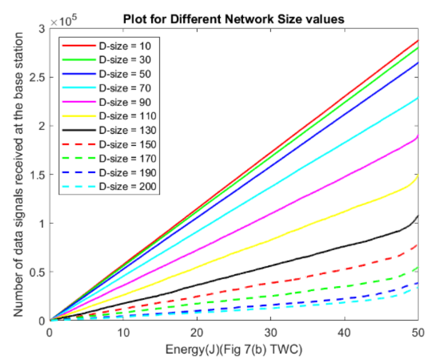


Figure 38: Energy expended to transmit data packets

From Figure 37, we found a similar growing trend observed in LEACH and LEACH-C, where the number of data packets received by the BS increases as the network diameter decreases. For example, at 2000 rounds, the smallest network (d=10m) shown by the solid red line is still thriving with the BS receiving close to 2×10^5 data packets, five times as much as the 0.4×10^5 data packets that the largest network (d=200m, shown by dotted light blue line) managed to send to the BS before dying completely at around 500 rounds. This again reiterates our theoretical understanding of the network performance attributed to the proportional relationship between energy dissipation and internodal distance squared. This effect is also seen in Figure 38 where the number of data packets reaching the BS is lesser for any given amount of energy when the network is larger.

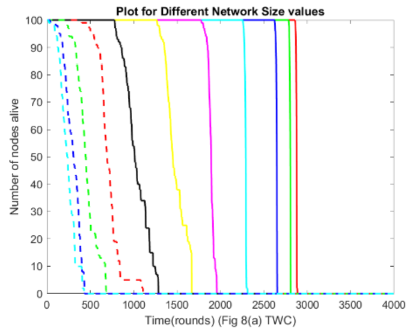


Figure 39: Number of nodes alive per unit time

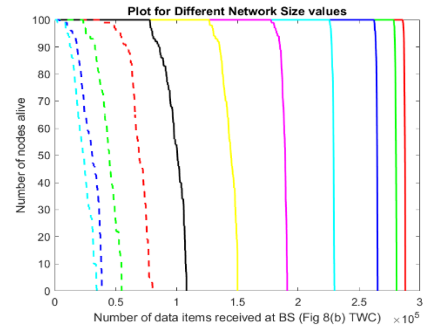


Figure 40: Number of nodes alive per amount of data sent to sink node (BS)

The overall increase in energy expenditure reduces the network lifespan as well, and this improvement gained through BCDCP is further exemplified by the system lifetime graph in Figure 39 and Figure 40 that the first onset of node death occurs much earlier for larger networks. It is obvious that there is a similar trend between these two plots due to the earlier mentioned reason, energy dissipation of each node is directly proportional to the distance between each node. Therefore, the number of nodes in a network with a smaller D-size remains alive for a longer time, which results in more data packets being received successfully at the BS. With network diameter D-size =200, the number of nodes alive drops drastically at the 10^{th} round with $3.5x10^4$ data packets being transmitted to the BS. On the contrary, this drop-phenomenon only started at around 2861^{st} round with $2.9x10^5$ data packets transmitted to the BS when D-size =10. With a clearer horizontal comparison in Figure 42 below, the number of rounds when the 1st, 10%, 50%, 70%, and 100% nodes die in the network appear much earlier with a larger D-size, which further supports the findings in Figure 39 and Figure 40.

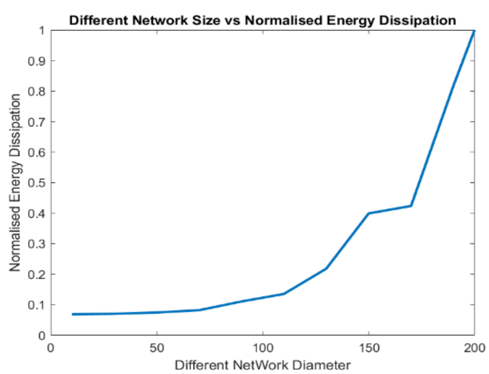


Figure 41: Energy dissipation vs network size

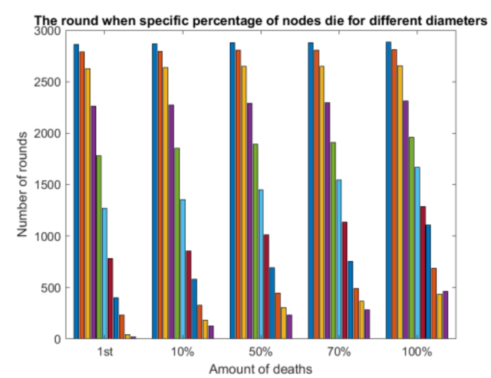


Figure 42: Round when specific percentage of nodes die for different size

Lastly, normalized energy dissipation versus network diameters is shown in Figure 41 above. There is an exponentially increasing trend of normalized energy dissipation as the network diameter increases. During D-size = 20~150, the growth trend is slow, however, after D-size > 170, there is a sharp increase followed. We assume that the difference between the growing speed might be due to the threshold distance d_0. In BCDCP, we use both the free-space propagation model and the two-ray ground propagation model to approximate the path loss due to wireless channel transmission. Given a threshold transmission distance of d_0, the free space model [Equation 2] is employed when $d \leq d_0$ which results in square growth. However, when $d > d_0$, the two-ray model [Equation 2] is being adapted which results in a quadratic aggressive growing trend in energy dissipation. Another point that needs to be mentioned here is, there is a growing trend slowing down during D-size = 150~170, we assume this might be due to radio model changing phase.

5 FINAL EVALUATION (Analysis: Everyone)

We will compare the performance of all 3 protocols studied in this paper – LEACH, LEACH-C and BCDCP. For fair comparison, LEACH will be simulated using the same radio model computation that considers free space or multipath effects. Similarly, the BS is located outside the network with a distance factor of 1.5 times of network diameter to be consistent with LEACH-C and BCDCP. The number of rounds will be 4000 and all other parameters will follow the ones used in Part 2.

LEACH	LEACH-C	BCDCP

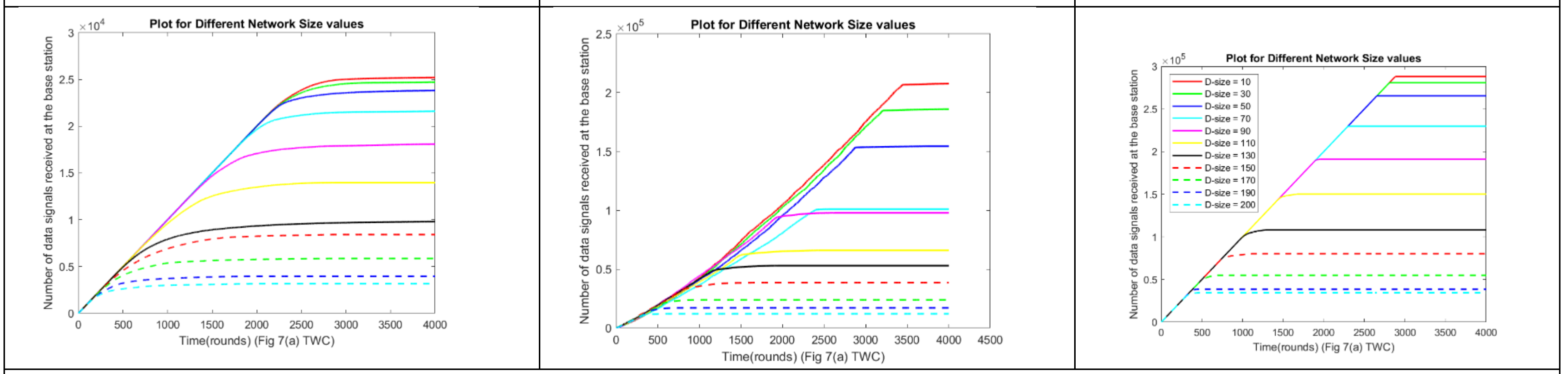


Figure 43 - Comparison of Data Delivery against No. of Rounds

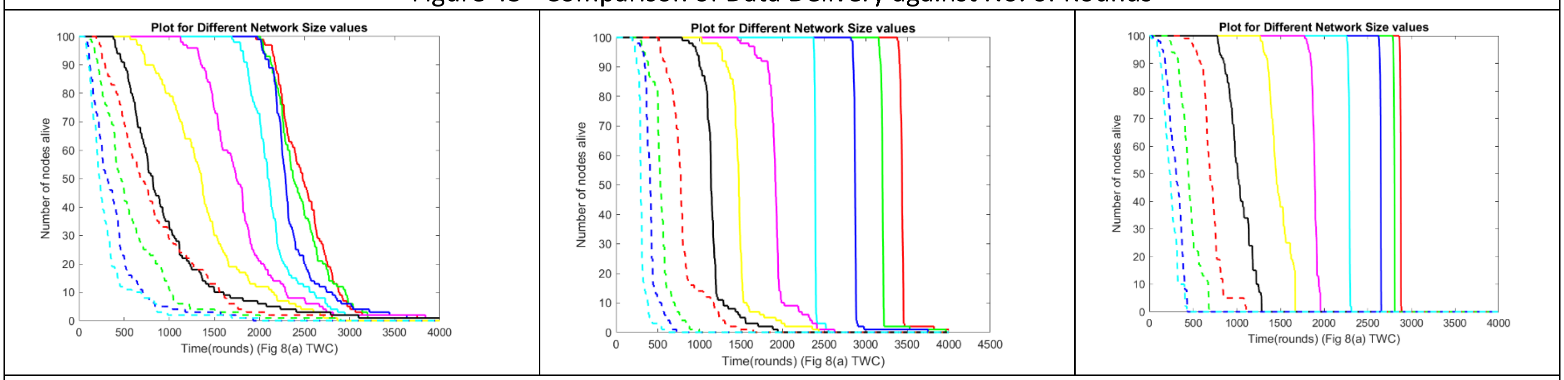


Figure 44 – Comparison of Nodes Alive against No. of Rounds

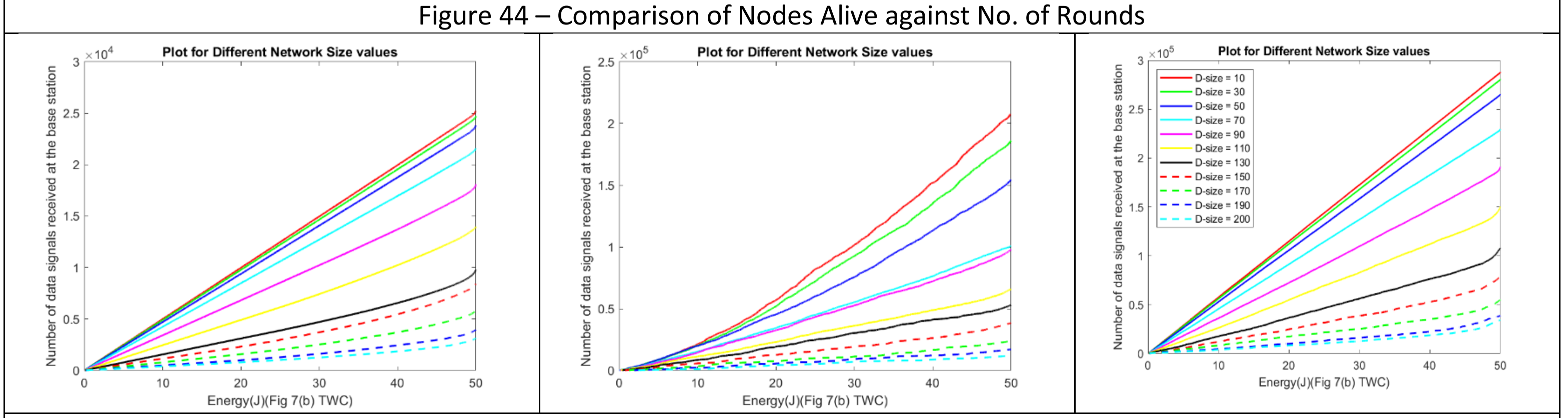


Figure 45 - Comparison of Data Delivery per unit of energy

We can see from the side-by-side comparisons that LEACH-C and BCDCP improves the network longevity in terms of the first onset of node death. From Figure 44, at network d=90m, the first death occurs at 1100 rounds for LEACH, 1500 rounds for LEACH-C (35% longer than LEACH) and

1750 rounds for BCDCP (60% longer than LEACH). Besides, both LEACH-C and BCDCP face faster network death upon first death illustrated by the steep slopes in Figure 44. This phenomenon can be explained since they all utilize the BS for cluster formation. The BS receives the information of the location and energy level of each node at the beginning of each round, so that it can assign the nodes with relatively higher energy to be the CH. In this way, they can distribute the burden of sending messages evenly among all nodes, so that all nodes die at similar times.

Figure 43 and Figure 45 show that the data received at BS improves with LEACH-C and BCDCP. The data delivered with LEACH is in the order of 10^4, while it is higher for both LEACH-C and BCDCP at the order of 10^5. BCDCP proves to be better with a higher value than Leach C. This corresponds to findings in [4] where LEACH-C delivers data faster and with more data per unit energy than LEACH and in [1] where BCDCP outperforms LEACH and LEACH-C. BCDCP enjoys better performance due to its cluster balancing algorithm that ensures equal cluster size and spatial distribution as compared to LEACH and LEACH-C which randomly distributes clusters. Additionally, the CH-to-CH routing helps in keeping energy cost low in large networks.

6 CONCLUSION

In this paper, we investigated the improvements in network performance with the use of LEACH variations, namely LEACH-C and BCDCP. We confirmed the theoretical understanding that LEACH-C and BCDCP are more energy-efficient than LEACH by having a better CH selection based on remaining energy and location of nodes. Both BCDCP and LEACH-C do well in distributing the burden of sending messages evenly among all nodes, so that all nodes die at similar times. BCDCP increases longevity of network by using its CH-to-CH spanning tree routing and cluster balancing that ensures equal cluster size and spatial distribution of clusters. Thus, BCDCP performs the best of the three in terms of network longevity and data delivery.

REFERENCE

[1] S. D. Muruganathan, D. C. F. Ma, R. I. Bhasin, and A. O. Fapojuwo, “A centralized energy-efficient routing protocol for wireless sensor networks,” *IEEE Commun. Mag.*, vol. 43, no. 3, pp. S8-13, Mar. 2005, doi: 10.1109/MCOM.2005.1404592.

[2] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, Jan. 2000, p. 10 pp. vol.2-. doi: 10.1109/HICSS.2000.926982.

[3] A. A. Hassan, W. Shah, M. F. Iskandar, and A. A.-J. Mohammed, “Clustering Methods for Cluster-based Routing Protocols in Wireless Sensor Networks: Comparative Study,” vol. 12, no. 21, p. 11, 2017.

[4] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *IEEE Trans. Wirel. Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002, doi: 10.1109/TWC.2002.804190.

[5] H. Shen, "Finding the k Most Vital Edges with Respect to Minimum Spanning Tree", Proc. IEEE Nat'l. Aerospace and Elect. Conf.., vol. 1, pp. 255-262, 1997-July.

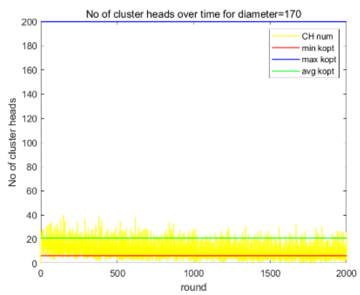
APPENDIX

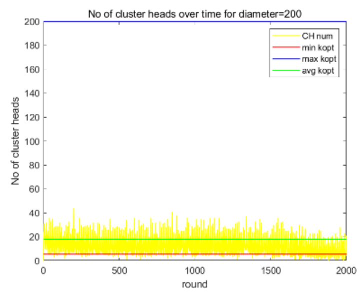
Section 3.1 – LEACH-C (Analysis: Emil & Zuona)

Experiment 2: BS inside the network
x_BS_location = 0.5; y_BS_location = 0.5;

When base station is located inside the network (in the center of the network for our experiment), the upper bound loses its meaning. The calculated upper bound is larger than the number of nodes and is set as 200 (number of nodes). This is because, the distance between a node and BS can be really small (node very close to BS), which results in extremely big optimum k value as k is inversely proportional to d_{toBS} squared.

$$\uparrow k_{opt} = \frac{\sqrt{N}}{\sqrt{2\pi}} \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}} \frac{M}{d_{toBS}^2 \downarrow}$$

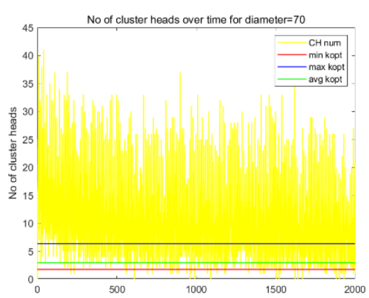


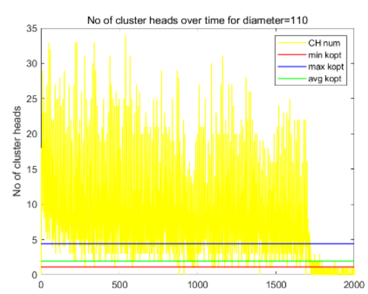


In order to avoid having extremely large optimal k value, we will be placing the BS outside the network for our further evaluations.

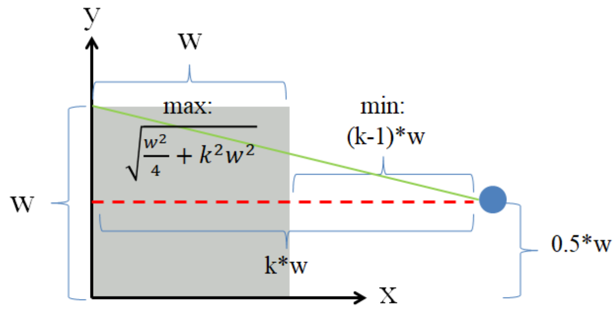
Experiment 3: BS outside of network, far from network. (Analysis : Emil & Zuona)
x_BS_location = 2;
y_BS_location = 0.5;

From figures, we can see that when base station is far from network, the theoretical optimum k range does not match the actual number of clusters. The theoretical value is smaller than the actual one.





Analysis of abnormality:



Assume the network is of width w and length w, and the location of BS is (k*w, 0.5*w). The possible farthest node to BS is at the northwest corner of network. The square of distance is:

$$\max d_{toBS}^2 = \frac{w^2}{4} + k^2w^2$$

The possible closest node to BS is at the middle of right boundary. The square of distance is:

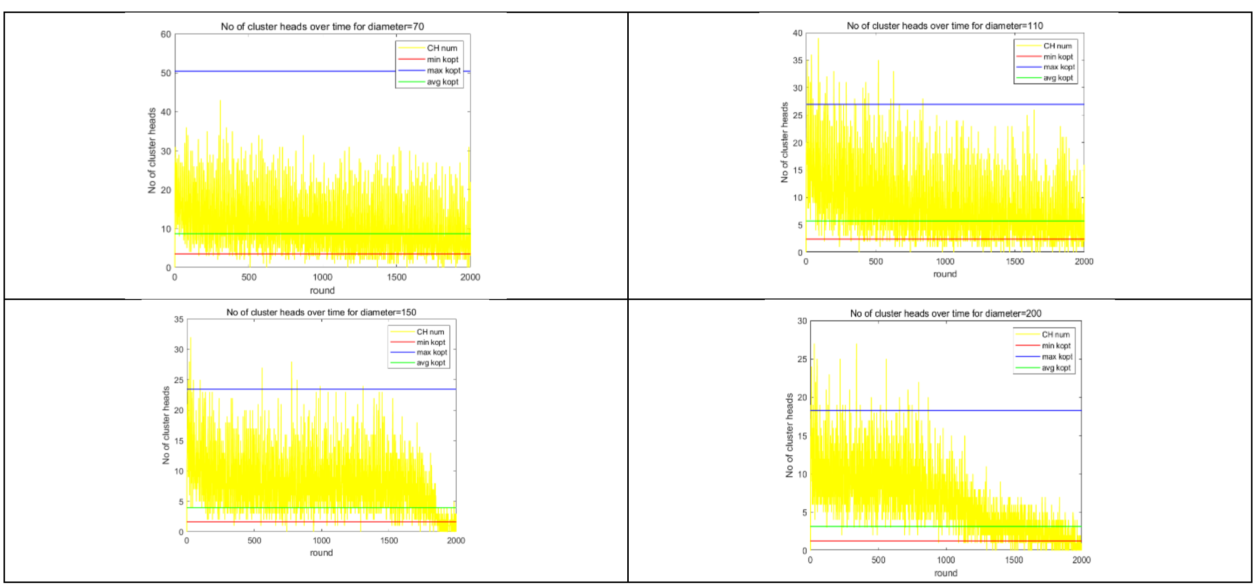
$$min\ d_{toBS}^2 = (k-1)^2w^2$$

Plug above values into the optimum k equation, we can get:

$$\max k_{opt} = \sqrt{\frac{N\varepsilon_{fs}}{2\pi\varepsilon_{mp}}}\frac{1}{(k-1)^2w}$$

$$\min k_{opt} = \sqrt{\frac{N\varepsilon_{fs}}{2\pi\varepsilon_{mp}}}\frac{1}{(\frac{1}{4}+k^2)w}$$

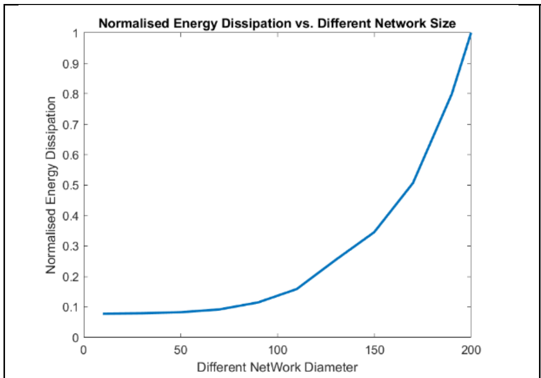
When k increases (BS is farther from network) or w increases (bigger size of network), the upper bound and lower bound of optimum k range both decrease. When k decreases (BS is closer to network) or w decreases (smaller size of network), the upper bound and lower bound of optimum k range both increase. For extreme parameters, such as network size which is too small (described in experiment 1) and BS too far (described in experiment 2), the theoretical range does not match with the actual number of CHs.

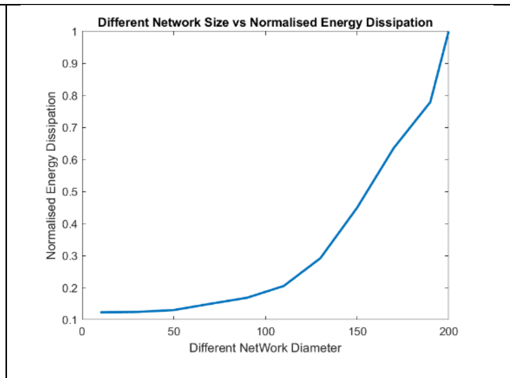


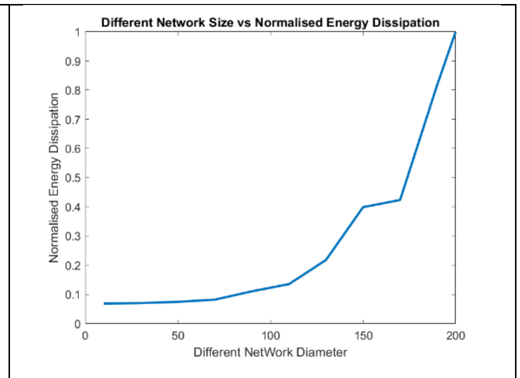
Section 6 - Final Evaluation (Analysis: Everyone)

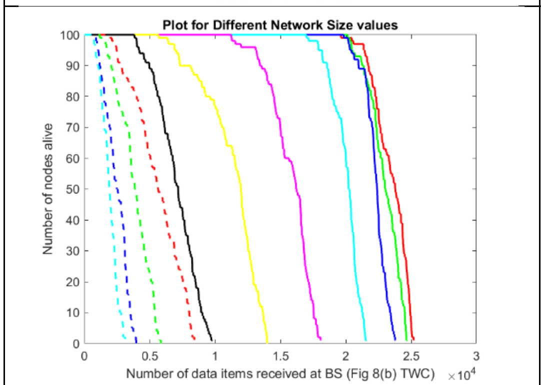
Parameter used:

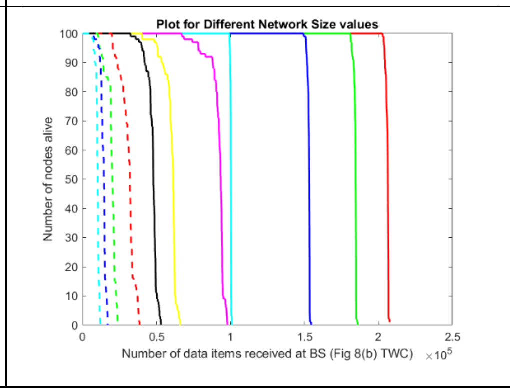
```
ETX=50*0.000000001; ERX=50*0.000000001; Efs=10*0.000000000001; Emp=0.0013*0.000000000001;
EDA=5*0.000000001; do=sqrt(Efs/Emp);
```

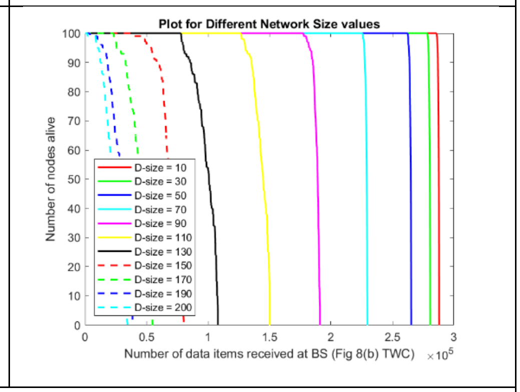




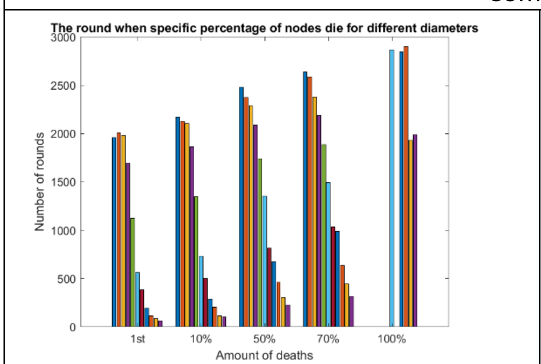


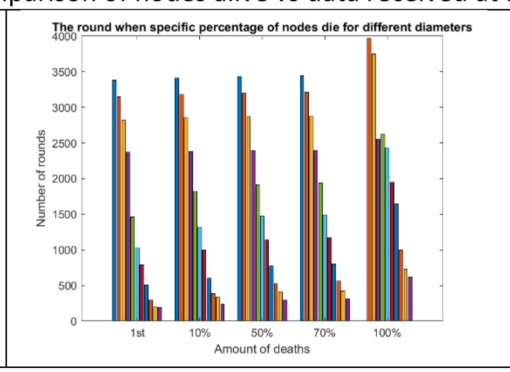


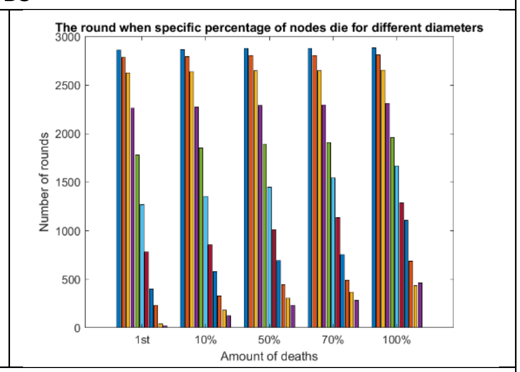




Comparison of nodes alive vs data received at BS







Comparison of no. of rounds against specific node death percentages

Statement of Contributions

A. Joint Work, e.g., in planning and conceptualization etc. (briefly list a few specific aspects):
1. Literature Review
2. Final Evaluation
3. Conclusion

B. Zhu Qi's Work (briefly list a few specific aspects):
1. Theoretical analysis of BCDCP
2. Assist in programming Part II BCDCP
3. BCDCP result analysis

C. Ivander Jonathan Marella Waskito's Work (briefly list a few specific aspects):
1. Programming for Part II BCDCP
2. Generate plots for BCDCP
3. Code explanation for Part II BCDCP

D. Emil Yong Kai Wen's Work (briefly list a few specific aspects):
1. Project Coordination
2. Programming of Part 1
3. Programming & analysis of optimum number of clusters
4. Programming & analysis of LEACH-C
5. Support BCDCP team by explaining in detail the skeleton leach code and leach C line by line in multiple meetings
6. Overall formatting and ensuring consistency of report

E. Tang Yuyi's Work (briefly list a few specific aspects):
1. Theoretical analysis of BCDCP
2. Assist in programming simulation Part II BCDCP
3. BCDCP result analysis

F. Chen Zuona's Work (briefly list a few specific aspects):
1. Programming of Part 1
2. Programming & analysis of optimum number of clusters
3. Programming & analysis of LEACH-C

G. Faisal Ahmad's Work (briefly list a few specific aspects):
1. Project Coordination
2. Section 1 of Report
3. Part 1 Extension on LEACH – Analysis

We agree that the statements above are truthful and accurate.

Signatures and Names:

(Zhu Qi)

(Ivander Jonathan Marella Waskito)

(Emil Yong Kai Wen)

Tang Yuyi

(Tang Yuyi)

Zuona Chen

(Chen Zuona)

(Faisal Ahmad)