CHAPTER 3 MUTUAL EXCLUSIVE DISTRIBUTIVE CLUSTERING

3.1 Introduction

Clustering protocols are needed in WSN for making them more sustainable, scalable and efficient. Number of clustering protocols was discussed in [9]. After intense survey it is found that distributive clustering protocols are easy to develop and no extra setup cost is required so distributive protocols must get preference although their disadvantage is that they are not always reliable. We in this chapter will present our proposed protocol named as Mutual Exclusive Distributive Clustering (MEDC). Section 3.2 is about background for this work. Section 3.3.1 will discuss some network model assumptions related to our work. 3.3.2 Will discuss Radio model of communication for protocol. Section 3.2.3 will present gist of this protocol and also enumerate an example to show working of MEDC. Section 3.4 will be about algorithm which is diagrammatically represented as flow chart in section 3.5. Section 3.6 is about experimental evaluation. We have simulated MEDC protocol on MATLAB. Results are taken on varying two parameters among them one is number of sensor nodes and second is range of communication. For simulation we had taken two values for sensors one is 100 another is 200. For Range of communication we had simulated on four values 20, 40, 60 and 80. Section 3.7 about result discussion after that work is concluded in section 3.8.

3.2 Background

Clustering protocols select cluster heads to reduce communication cost so that network can live to maximum. A number of clustering protocols has been produced which may have different working parameters and features. We had taken residual energy or can say power based clustering protocols. HEED is one of them we will discuss HEED protocol in detail. We had considered this protocol as base protocol of our work and compared performance with it. The reason for considering HEED is our network assumptions. HEED protocol has worked on similar network assumptions [10]. HEED is distributive in nature and power base protocol. It selects cluster heads on basis of two factors one is
nodes remaining energy and second factor is intra-cluster communication cost. HEED protocol does need extra capable sensor nodes for location awareness even this protocol need not any special sensor node distribution methods. Cluster head selection in HEED protocol is primarily based on the remaining energy of each sensor node. Remaining energy is estimated current energy in the node. Secondary clustering parameter that authors has considered is intra-cluster communication cost. This protocol includes three phases. First phase is known as Initialization phase, in which broadcast cost is calculated according to nodes under range of communication and also cluster head probability for each sensor node is calculated. Second phase is repeat phase; this phase decides tentative cluster heads. If any node is having probability equal to one then this sensor node will be declared as final cluster heads otherwise least cost head is taken as tentative cluster head. Last phase is called finalizing phase; Cluster heads are finalized in this phase and sensor nodes become cluster member under least cost reachable cluster head. HEED protocol works with minimum selection probability and use minimum network operation interval due to which HEED protocol easily tuned to optimize resource usage according to the network density and application requirements. HEED protocol also ensure that it will terminates in a constant number of iterations, independent of network diameter [10].

3.3 MEDC

3.3.1 Network Model assumption

- Sensor nodes are of homogeneous type means they have the same capabilities and resources like battery power etc.
- Sensor nodes are stationary deployed in local region for monitoring and the data sink is located far from the sensing field.
- Network is formed of location unaware sensor nodes; nodes does not having any capability like GPS
- Sensory data is aggregated at different levels and sent to data sink generally called base station at regular period of time
- A different identifier will be used for each sensor node.
- Communication is done on symmetric links. Communication can be bidirectional.
3.3.2 Radio model Equations

Sensors energy is dissipated on transmission and receiving activity along with sensing. If any sensor want to transmit t bits to a node located at distance d. Then energy dissipation is calculated by equation 3.1

- Transmission Energy Dissipation
  - $E_{tx} (t) = t*E_{elec} + E_{tx\_amp} (t, d)$ (3.1)
  - $E_{tx\_amp} (t, d) = t*d^2*E_{fs}$ (3.2)
  - $E_{tx\_amp} (t, d) = t*d^4*E_{mp}$ (3.3)

Where $E_{tx}$ represent transmission energy will be calculated from electron energy ($E_{elec}$) and amplification energy ($E_{tx\_amp}$). Amplification energy consumption also varies depending on free space communication or multipath communication equations 3.2 and 3.3 are shown. Free space energy and multipath energy consumption is represented as $E_{fs}$ and $E_{mp}$ respectively. If any sensor node is receiving t bit from any sensor node then energy dissipated for t bits is represented as $E_{rx}$

- Receiving Energy Dissipation
  - $E_{rx} (t) = t*E_{elec}$ (3.4)

3.3.3 Gist of MEDC

Mutual Exclusive Distributive Clustering (MEDC) will form clusters of sensor on basis of mutual exclusion algorithm. MEDC Clustering protocol chooses cluster heads on basis of mutual exclusion algorithm from the number of sensors. Within a range of communication (says $R_c$), only that sensor node which is having maximum of residue energy will be chased as cluster head from that range. This protocol is having four steps in single iteration and each iteration will decide new cluster heads which are having maximum residue energy under a range of communication. Under $R_c$, the Cluster head sensor node will be decided on factor of remaining energy. Sensor node which one is having highest remaining energy among the sensors node under $R_c$ will be chosen as cluster head. The idea of Mutual exclusion works by message passing system we will say it by advertisements. Why we are considering the remaining energy factor, is because sensor’s remaining energy will be different for each sensor nodes infect it depend on how much each sensor node spends energy for sending and receiving advertisements up to last
iteration. Which in turn how much advertisement depends on how many sensors were under range of communication (Rc)

Here we want to highlight why we are considering only residual energy for choosing cluster heads not distance factor. Some reader may say distance can also be good factor for selection. But here are considering sensors are deployed in local region say in 500* 500 meters; base station is far away say at 25 Km. So relative distance for all sensor nodes will be same for all sensors. As shown in diagram for two sensor nodes.

Secondly here we consider communication is multipath so in between relay nodes will be there so relay nodes will forward the data without consideration from which sensor they are receiving data for forwarding. The third point is we cannot burden near located nodes for relay service in that case they deplete their energy and cause portioning of network problem.

Steps of MEDC are as follows:

**Step 1**
Sensors will send advertisement packets to all other nodes that come under their range of communication say $R_c$. Advertisement packets consists residual (remaining) energy of
sensors nodes. Each sensor node will precipitate in sending and receiving of advertisements. Received advertisements will be saved in queues of sensor nodes.

**Step 2**
All sensors nodes will check their queue. Each sensor node will send OK message after selecting sensors from queue. The selection criteria for OK message is; comparison of residual energies. Sensor node will select other sensors from queue which are having more residual energy then its own. The sensor node which has sent any OK message to other sensor from queue that cannot be cluster head for this iteration. If there exists any node that is not sending any OK message then it will wait.

**Step 3**
Step 3 will decide which sensors will act as cluster heads and which will be cluster members. As CDMA slot ends sensors will check their own status. The sensor node who has not sent any OK message who has not found any node from queue with more energy then owns it means this node will declare itself as cluster head. Cluster head declaration message will be sent to all nodes that come under its range of communication.

**Step 4**
Cluster head sensor nodes will collect and aggregate data from cluster members under $R_c$. Cluster heads will transmit the aggregated data to sink node. Cluster head rotation will be done in next iteration. Next iteration starts from step 1.

**Example**
Here we want to show how MEDC will run and select cluster heads. In following figure five sensors are randomly deployed and they are identified as IDi. When they are deployed they were having same energy say 0.2 see figure 3.2. Figure 3.2 shows that ID1, ID2, ID3, ID4, ID5 are the sensors within range $R_c$. MEDC protocol will start working. In step 1 sensor nodes start sending and receiving advertisements under range of communication, Sensor nodes deplete their energy on sending and receiving advertisements due to which it may possible that nodes have different residual energies refer figure 3.3. Sensor nodes queue incoming advertisements here. Sensor nodes ID1, ID2, ID3, ID4 and ID5 will advertise their remaining battery life 0.12, 0.14, 0.19, 0.13
and 0.11 respectively. Each sensor node will maintain queue for incoming advertisements. Sensor queues are shown in Table 3.1.

![Initial Deployment of sensor](image)

Table 3.2 has shown that each node has sent OK message to which nodes. For example sensor node ID1 has received advertisement from ID2, ID3, ID4 and ID5. But from queue it will send OK message to ID2, ID3 and ID4; but not to ID5. The reason is ID5 is having residual energy 0.11 which is less than residual energy of its own i.e. 0.12 so ID1 will not send OK message to ID5. Same step will be executed for each sensor.
In step 3 sensor nodes will check their status. If any sensor who has not sent any OK message this means that this node is having maximum energy under range of communication.

Here from table we came to know that node ID3 has not sent any OK message. Here sensor node ID3 will declare itself as cluster head and send this declaration message to ID1, ID2, ID4 and ID5 from range of communication see figure 3.4 mentioned as green colored.

During Step 4 cluster head ID3 will receive sensor data from cluster members; aggregate it and transmit it to base station. Next iteration will start after TDMA slot which is required to transfer data up to base station. Cluster heads will be change in successive iteration. Iteration will start from step1 to step 4.
Table 3.1 - Sensor Queue

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Queues</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID1</td>
<td>ID2 (0.14), ID3 (0.19), ID4 (0.13), ID5 (0.11)</td>
</tr>
<tr>
<td>ID2</td>
<td>ID1 (0.12), ID3 (0.19), ID4 (0.13), ID5 (0.11)</td>
</tr>
<tr>
<td>ID3</td>
<td>ID1 (0.12), ID2 (0.14), ID4 (0.13), ID5 (0.11)</td>
</tr>
<tr>
<td>ID4</td>
<td>ID1 (0.12), ID2 (0.14), ID3 (0.19), ID5 (0.11)</td>
</tr>
<tr>
<td>ID5</td>
<td>ID1 (0.12), ID2 (0.14), ID3 (0.19), ID4 (0.13)</td>
</tr>
</tbody>
</table>

Figure 3.4 - Node ID3 will become Cluster Head
Table 3.2 Sent OK messages

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Sent OK message to</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID1</td>
<td>ID2, ID3, ID4</td>
</tr>
<tr>
<td>ID2</td>
<td>ID3</td>
</tr>
<tr>
<td>ID3</td>
<td>Nil</td>
</tr>
<tr>
<td>ID4</td>
<td>ID2, ID3</td>
</tr>
<tr>
<td>ID5</td>
<td>ID1, ID2, ID3, ID4</td>
</tr>
</tbody>
</table>

It might possible that some issue can occur out of which we are listed some of them and Issue resolving in MEDC

1. If a sensor gets more than one cluster head’s cluster_head_declaration.
   Solution to this issue can be that the sensor will consider its cluster head with lower ID from many cluster head’s and it will sensed it sensory values to that cluster head only.

2. If a sensor does not get any cluster_head_declaration.
   Solution to this issue can be that the sensor waits till the next iteration.

3. If a sensor does not get any OK message.
   Solution to this issue can be that the sensor will act as cluster head for its own.

4. If a sensor has sent cluster_head_declaration and it get some other sensors cluster_head_declaration
   Solution to this will be on basis of timestamp ordering. It might possible that message has been transmitted in previous iteration and get received in next iteration may be due to congestion. Message with lower timestamp will be ignored. Events are logically synchronized each event carry timestamp along it.
3.4 Algorithm MEDC

IDi : ID of node i.

Rf : Radius of frequency/ Range of communication.

Qi: Queue of sensor i.

Epresent_i: Present energy of node i.

Procedure Cluster_formation (n)

1. For each Next iteration
2. For each IDi
3. Counter =0
4. For each IDj within Rf of IDi
5. Advertise Epresent_i
6. For each IDi
7. Put all incoming advertisements from sensors j into Qi
8. For each IDi
9. While Qi is not empty
10. If Epresent_i <= Epresent_j
11. Send ok message to IDj
12. Counter = 1
13. Else
14. Delete this advertisement from queue
15. For each IDi
16. If counter =0
17. Send cluster_head_declaration message to IDj within Rf

Procedure cluster formation will be executed on each sensor with start of TDMA slot.
One TDMA slot has setup phase in which clusters are formed rest part is steady phase
cluster heads collect information from cluster members. Aggregation is done at cluster
heads and then this aggregated information is sent to base station at the end of TDMA
slot. TDMA slots are usually taken by considering maximum time transmission can take.
3.5 Flow Chart

Start

\[ i = 0, \ n=100 \]

\[ i < n \]

\[ \text{counter } i=0 : i++ \]

\[ i = 0, \ n=100 \]

\[ i < n \]

Advertise Epresent_i

Queue advertisements from sensor j to Qi

\[ i++ \]

\[ i = 0, \ n=100 \]

\[ i < n \]

\[ i = 0, \ n=100 \]

\[ i < n \]

\[ \text{Delete Advertisement from queue} \]

\[ \text{Send OK message to} \]

\[ \text{ID}_j \]

\[ \text{Counter } _i=1 \]

\[ \text{Send Cluster Head Declaration within } R_c \]

End of Round
3.6 Experimental Evaluation

For simulation of MEDC on MATLAB we had taken following parameters. Shown In table 3.3.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Abbreviation</th>
<th>Values</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random field x axis</td>
<td>Xm</td>
<td>100</td>
<td>Meters</td>
</tr>
<tr>
<td>Random field y axis</td>
<td>Ym</td>
<td>100</td>
<td>Meters</td>
</tr>
<tr>
<td>Initial energy of sensor</td>
<td>Eo</td>
<td>0.05</td>
<td>Joule</td>
</tr>
<tr>
<td>Total number of sensor</td>
<td>N</td>
<td>100,200</td>
<td></td>
</tr>
<tr>
<td>Transmission energy</td>
<td>Etx</td>
<td>50*1.E-12</td>
<td>Joule per bit</td>
</tr>
<tr>
<td>Receiving energy per bit</td>
<td>Erx</td>
<td>50*1.E-12</td>
<td>Joule per bit</td>
</tr>
<tr>
<td>Free space energy per bit</td>
<td>Efs</td>
<td>10*1.E-12</td>
<td>Joule per bit</td>
</tr>
<tr>
<td>Data aggregation energy</td>
<td>EDA</td>
<td>5*1.E-12</td>
<td>Joule per bit</td>
</tr>
<tr>
<td>Advertisement energy</td>
<td>Eadv</td>
<td>50*1.E-12</td>
<td>Joule per bit</td>
</tr>
<tr>
<td>Range of Communication</td>
<td>RC</td>
<td>20,40,60,80</td>
<td>Meters</td>
</tr>
</tbody>
</table>

Figure 3.5 - Initial Deployment of 100 Sensors
Figure 3.6 - Sensor with Cluster Heads

Figure 3.7 - Initial Deployment of 200 Sensors
Figure 3.8 - Sensor with Cluster Heads

Figure 3.9 - Performance of MEDC when no of sensors are 100 and Rc is 20
Figure 3.10 - Performance of MEDC when no of sensors are 100 and Rc is 40.

Figure 3.11 - Performance of MEDC when no of sensors are 200 and Rc is 20.
Figure 3.12- Performance of MEDC when no of sensors are 200 and Rc=40

Figure 3.13- Comparative Results of MEDC and HEED number of sensors are 100 Rc is 20
Figure 3.14 - Comparative Results of MEDC and HEED number of sensors are 200 Rc is 20.

Figure 3.15 - Comparative Results of MEDC and HEED number of sensors are 100 Rc is 40.
Figure 3.16- Comparative Results of MEDC and HEED number of sensors are 200 Rc is 40

Figure 3.17- Comparative Results of MEDC and HEED number of sensors are 100 Rc is 60
Figure 3.18 - Comparative Results of MEDC and HEED number of sensors are 200 Rc is 60

Figure 3.19 - Comparative Results of MEDC and HEED number of sensors are 100 Rc is 80
Figure 3.20- Comparative Results of MEDC and HEED number of sensors are 200 Rc is 80

Figure 3.21- Comparison of No of dead sensors V/S Rounds when n=100 and Rc=20
Figure 3.22-Comparison of No of dead sensors V/S Rounds when n=100 and Rc =20 along with detailed table.

Figure 3.23-Comparison of No of dead sensors V/S Rounds when n=100 and Rc=40
Figure 3.24-Comparison of No of dead sensors V/S Rounds when n=100 and Rc =40 along with detailed table

Figure 3.25-Comparison of No of dead sensors V/S Rounds when n=200 and Rc=20
Figure 3.26 - Comparison of No of dead sensors V/S Rounds when n=200 and Rc=20 along with detailed table

Figure 3.27 - Comparison of No of dead sensors V/S Rounds when n=200 and Rc=40
Figure 3.28 - Comparison of No of dead sensors V/S Rounds when n=200 and Rc =40 along with detailed table

Figure 3.29 - Comparison of No of dead sensors V/S Rounds when n=100 and Rc=60
Figure 3.30-Comparison of No of dead sensors V/S Rounds when n=100 and Rc =60 along with detailed table

Figure 3.31-Comparison of No of dead sensors V/S Rounds when n=100 and Rc=80
Figure 3.32 - Comparison of No of dead sensors V/S Rounds when \( n = 100 \) and \( R_c = 80 \) along with detailed table

<table>
<thead>
<tr>
<th>Number of Rounds</th>
<th>HEED</th>
<th>MEDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>37</td>
<td>33</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>60</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>70</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>80</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>90</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

Figure 3.33 - Comparison of No of dead sensors V/S Rounds when \( n = 200 \) and \( R_c = 60 \)
Figure 3.34-Comparison of No of dead sensors V/S Rounds when n=200 and Rc =60 along with detailed table

Figure 3.35-Comparison of No of dead sensors V/S Rounds when n=200 and Rc=80
3.7 Results Discussions

MATLAB results of MEDC are given from figures 3.5 to figure 3.20. Figure 3.5 is shown result for initial deployment of one hundred sensors over a field. Random function is used for deployment. Figure 3.6 is result of MATLAB shown here for scenario when some sensors under their range become cluster heads. Blue nodes are cluster heads and red nodes are sensors which will act as cluster members under clusters heads from their range of communication. Figure 3.7 and figure 3.8 are results same as like 3.5 and 3.6 respectively but changed n (number of sensor nodes are 200). Figure 3.9 is shown for performance of MEDC on parameter n=100 and range of communication is taken as 20. Performance is measured in terms network life time. Network life time is measured as number of alive nodes per round. In result graph Y axis represents alive nodes and X axis represents rounds. Figures 3.10, 3.11, 3.12 are performance analysis of MEDC by varying values of n and Rc (Range of communication) n is varied on two values 100 and 200. Rc is also varied by two values one is 20 and another is 40. Figures 3.13-3.20 are
Comparative results of MEDC with HEED. Comparative results are also taken on n value 100 and 200. Rc value is varied from 20, 40, 60 and lastly 80. It is concluded from results that MEDC works better in contrast of HEED. On parameters n=100 and Rc=20 it gives 33.3 improvement in network life time. As parameter are changed, graph of MEDC come close in performance to HEED.

Reason for this decrement in graph is increase cost of communication. When we increase n from 100 to 200 or Rc from 20 to 40, 60 or 80 then their will be more no of sensors under range of communication to each other. More sensors under range more advertisements which mean more cost of communication which result decrease in performance. Figure from 3.21 to 3.36 show results of dead nodes corresponding to rounds. X axis represent rounds and y axis represent number of dead sensor nodes. Up to Graph 3.28 we calculate dead nodes are taken after every hundred rounds and this process is repeated on varying both parameters one is number of sensors deployed and second parameter is range of communication. From figure 3.29 onwards we have calculated dead nodes after every ten rounds as from performance analysis of network life time we came to know that all nodes become dead before 100 rounds for all three protocols that’s why for near check we calculated after ten rounds and plotted graph up to hundred rounds on x axis

3.8 Conclusion

In wireless sensor networks battery life saving is important aspect. Clustering protocols provide a solution to it. This work has proposed a new clustering protocol named MEDC. MEDC is distributive protocol works on message passing system. It select cluster on mutual exclusion algorithm basis. MEDC protocol is simple in nature and select cluster heads to those sensors which have maximum residual energy within range of communication.

On similar network assumption MEDC perform better than HEED because it consider directly residual energy up to last iteration for next iteration’s cluster heads selection in contrast of HEED where residual energy is considered for probability calculation. It is also found in worst cases when the sensors with long range of communication are deployed poorly large in number over small region; which may
increase in between message communication among the sensors even in these cases MEDC perform equal to HEED.