EFFICIENT DISTRIBUTED CLUSTERING

4.1 Introduction

Clustering protocols are important because they have major role in improving network life to make networks more sustainable, scalable and efficient [64]. From survey it is found that, what are different working parameters and features of various clustering protocols? In previous chapter we proposed protocol named as Mutual Exclusive Distributive Clustering (MEDC). Section 4.2 discuss about background of MEDC and HEED protocol. HEED protocol work on Chprob and Snbr on the second side MEDC protocol work on mutual exclusion distributed algorithm. Working parameter of MEDC was residual energy and range of communication. Experimental results showed MEDC perform better than HEED in most of times, even in challenging condition it perform equal but not less than HEED. One part of background is about MEDC, algorithm can be referred from previous chapter.

Section 4.3.1 will discuss some network model assumptions related to our work 4.3.2 will discuss Radio model of communication for protocol. Section 4.3.3 will present gist of MEHEED protocol. This work is carried to add benefits of both two protocols one is MEDC and other is HEED. MEHEED protocol will take three parameters for its working. Out of them one parameter will be Chprob as like HEED does. Next two parameters will be residual energy and range of communication as like MEDC does. MEHEED will firstly try to select cluster heads on Chprob basis.

Section 4.4 will be about algorithm which is diagrammatically represented as flow chart in section 4.5. Section 4.6 is about experimental evaluation. We have simulated MEHEED 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. Last section 4.7 is about result discussion and section 4.8 will discuss conclusion of this work. Conclusion part also discuss advantage and drawbacks of this proposed work.
4.2 Background

To develop MEHEED we have tried to merge two protocols one is HEED and second is MEDC [80]. Both protocols are distributive in nature. Here we want to present review of these two protocols. HEED is distributive in nature and power base protocol [10]. HEED protocol periodically selects cluster heads according to a hybrid of the node residual energy and intra-cluster communication cost. HEED has an advantage that it does not require special node capabilities, such as location-awareness, or special node distribution etc. 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.

The algorithm is divided into three phases: first one is initialization phase, second one is repletion phase followed by third phase that is finalizing phase. In first phase 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. Cluster heads are finalized in last phase. And sensor nodes become cluster member under least cost reachable cluster head. HEED protocol also ensure that it will terminates in a constant number of iterations, independent of network diameter. Here is algorithm of HEED protocol

I. Initialize
   1. Snbr ← \{v: v lies within my cluster range\}
   2. Compute and broadcast cost to $\epsilon$ Snbr
   3. CHprob ← max (Cprob $\times$ Eresidual/ Emax , pmin)
   4. is_final_CH ← FALSE

II. Repeat
   1. If ((SCH ← \{v: v is a cluster head\}) != $\emptyset$
   2. my_cluster head ← least cost(SCH)
3. If (my_cluster_head = NodeID)
   4. If (CHprob = 1)
      5. Cluster_head_msg(NodeID, final CH, cost)
      6. is_final_CH ← TRUE
      7. Else
         8. Cluster_head_msg(NodeID, tentative_CH, cost)
         9. ElseIf (CHprob = 1)
            10. Cluster_head_msg(NodeID, final CH, cost)
            11. is_final_CH ← TRUE
            12. ElseIf Random (0, 1) <= CHprob
               13. Cluster_head_msg(NodeID, tentative_CH, cost)
               14. CHprevious ← CHprob
               15. CHprob ← min(CHprob × 2, 1)
               Until CHprevious = 1

III. Finalize

1. If (is_final_CH = FALSE)
   2. If ((SCH ← {v: v is a cluster head}) ≠ ∅)
      3. my_cluster_head ← least_cost(SCH)
      4. join cluster(cluster_head_ID, NodeID)
      5. Else Cluster_head_msg(NodeID, final CH, cost)
      6. Else Cluster_head_msg(NodeID, final CH, cost)

The second protocol we had considered is MEDC which is our first proposed work. Mutual Exclusive Distributive Clustering (MEDC) will form clusters of sensors on the basis of mutual exclusion algorithm. In MEDC Clustering protocol the Cluster heads will be chosen in a mutual exclusive way over a range of communication. Under a range of communication (says R_c), only a sensor which is having maximum of residue energy will
be the cluster head. The proposed protocol will run in iterations, each iteration follow four steps and the cluster head will be reechoes in succeeding iteration. 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 ($R_c$). Algorithm can be referred from section 3.4.

4.3 MEHEED

4.3.1 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.
- Sensor nodes are considered as location–unaware that means they are not equipped with GPS or other similar equipment.
- Periodically the recently sensed data and information by all nodes are gathered and sent to the data sink after aggregation.
- Each sensor node is assigned a unique identifier (ID). Each sensor will be assigned initial probability $C_{prob}$ which is used to calculate $Ch_{prob}$.
- The links are assumed to be symmetric. Communication can be bidirectional.

4.3.2 Radio model Equations

Sensors energy is dissipated on transmission and receiving activity along with sensing [61]. If any sensor want to transmit $k$ bits to a node located at distance $d$. Then energy dissipation is calculated by equation 4.1

- Transmission Energy Dissipation
\[ E_{tx}(k) = k* E_{elec} + E_{tx\_amp}(k, d) \quad (4.1) \]
\[ E_{tx\_amp}(k, d) = k*d^2*E_{fs} \quad (4.2) \]
\[ E_{tx\_amp}(k, d) = k*d^4*E_{mp} \quad (4.3) \]

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 4.2 and 4.3 are shown. Free space energy and multipath energy consumption is represented as \( E_{fs} \) and \( E_{mp} \) respectively. If any sensor node is receiving \( k \) bit from any sensor node then energy dissipated for \( k \) bits is represented as \( E_{rx} \)

- Receiving Energy Dissipation
  \[ E_{rx}(k) = k* E_{elec} \quad (4.4) \]

### 4.3.3 Gist of MEHEED

MEHHED protocol is extension work of our first contributory work MEDC protocol. MEHEED protocol is combination of MEDC and HEED protocols. MEDC protocol was working on the parameters of residue energy \( E_{residual} \) and range of communication on the other side HEED protocol considers three factors one of them is \( Ch_{prob} \), second is \( S_{nbr} \) and third is range of communication. The proposed MEHEED protocol will take first parameter same i.e. \( Ch_{prob} \) and second parameter will be \( E_{residual} \) instead of \( S_{nbr} \), the third factor is same for all three protocols here i.e. Range of communication.

The idea to change the second parameter is; instead of considering previous calculated \( S_{nbr} \) which was dependent on remaining energy of starting level, why not to consider \( E_{residual} \) that have been recalculated after each iteration. Benefit of this idea will be that recent updated value i.e. \( E_{residual} \) will also reflect energy detrainment of previous cluster heads. So decisions will be more accurate. MEHEED protocol adopts benefit of both protocols. When it take first decision sensor’s \( Ch_{prob} \) will be checked, which is calculated in first phase. If \( Ch_{prob} \) come out equal to one that it will select that particular sensor as cluster head, rest all computations will be simply skipped. If it is not equal to one in that case part of MEDC algorithm works out and selects cluster head which is having maximum residual energy. MEHEED protocol will works in two phases.

**First phase** will be of initialization and calculations phase is as like HEED. In first phase first of all sensors under the range of communication are queued. On basis of this queue,
Communication cost and $E_{\text{residual}}$ will be calculated. After that, $C_{\text{prob}}$ will be calculated on basis of residual energy and predefined $C_{\text{prob}}$ as like of HEED protocol.

**Second phase** will decide the cluster head and the cluster members under clusters. Second phase will decide which sensor will be cluster head. This decision firstly depends on $C_{\text{prob}}$ after that this decision will depend on the $E_{\text{residual}}$ unlike HEED, in which second factor was $S_{\text{nbr}}$. If and sensor node is having $C_{\text{prob}}$ is equal to one that it will be directly declared as cluster head and sensors under its queue will be cluster members for this. In this case further computations will be skipped both computation time and energy will be saved. In case if there is no sensor node under a range of communication have met first selection criteria then selection will be done according to MEDC protocol i.e. on basis of remaining energy. MEDC protocol will select cluster head to that sensor that is having maximum residual energy in simple way; no tentative cluster heads will be selected as in HEED again complex computations will be skipped. The results have been shown that this idea has given more affective results to save network life. Reader may wish to know how $C_{\text{prob}}$ can be equal to 1?

As we said $C_{\text{prob}} = \max (C_{\text{prob}} \times E_{\text{residual}}/E_{\text{max}}, p_{\text{min}})$.

Assume sensors may have their energy in range [0.0, 0.2]. 0.0 joule is the minimum energy a sensor can have and 0.2 joule is the maximum. Here $C_{\text{prob}}$ is initialized as 0.5. $P_{\text{min}}$ is assumed minimum probability equal to 0.1.

$E_{\text{residual}}$ is remaining energy of sensor say nodes i. Assume node i is having 0.18. $E_{\text{max}}$ will be calculated by average from network energy means calculating total energy of all sensor nodes and dividing by total no of sensor nodes. The fact is when sum will be calculated at that time some sensors might have energy at zero but they are part of network so their number will be calculated in network but energy in sum will be zero. Say we are having 50 nodes and. Let us say we had calculated sum that come out 4.5 joule. Average energy will be assigned to $E_{\text{max}}$ that will be (4.5/50=0.09).

Substituting the values in equation $C_{\text{prob}} = \max (0.5 \times 0.18/0.09, 0.1) = 1$
<table>
<thead>
<tr>
<th>HEED</th>
<th>MEDC</th>
<th>MEHEED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Ch&lt;sub&gt;prob&lt;/sub&gt;</td>
<td>1. Eresidual</td>
<td>1. Ch&lt;sub&gt;prob&lt;/sub&gt;</td>
</tr>
<tr>
<td>2. Snbr (cost)</td>
<td>2. Range of Communication</td>
<td>2. Eresidual</td>
</tr>
</tbody>
</table>

### 4.4 Algorithm MEHEED

Two phase protocol

i- Sensor node

Rf- Range of frequency

v- Nodes under Rf

Qi-Queue of sensor i

Cprob- Predefined probability of each sensor

CHprob- Probability of being cluster head

**Phase I**

1. For each sensor i
2. Qi &lt;-- v: v under of Rf
3. Compute the communication cost of i after investing on Qi. And find the Eresidual
4. CHprob = max (Cprob × Eresidual/Emax, pmin)

**Phase II**

1. For each sensor i
2. If (Chprobi == 1)
3. cluster_head_declaration message
4. else if
5. Advertise Epresent_i each j within Qi
6. Queue all incoming advertisements from sensors j into adv_Qi
7. Compare Epresent_i with Epresent_j in adv_Qi
8. If (Epresent_i ≥ Epresent_j)
9. cluster_head_declaration message
10. else
11. find j in adv_Qi having Epresent_j ≥ Epresent_i
12. compare this Epresent_j with other sensors in adv_Qi and find the highest energy sensor let its r
13. do Chprobr = 1
14. do Chprobi = Chprobi * 2
4.5 Flow Chart

Start of Clustering after interval

\[ i = 1 \]

\[ i \leq 100 \]

Yes

Insert all the sensors under Rf into \( O_i \)

Compute the communication cost and \( E_{\text{residual}} \)

\[ \text{CHprob} = \max \left( \text{Cprob} \times \frac{E_{\text{residual}}}{E_{\text{max}} \cdot \text{dmin}} \right); i++ \]

\[ i = 1 \]

\[ i \leq 100 \]

Yes

If \( \text{CHprob} i = 1 \)

Yes

Make i as cluster head
Send cluster head declaration, \( i++ \)

No

Run MEDC algorithm and find r as cluster head

\[ \text{Chprobr} = 1 \]
\[ \text{Chrobi} = \text{Chrobi} \times 2; i++ \]

Stop of interval
4.6 Experimental Evaluation

For simulation of MEHEED on MATLAB we had taken following parameters. Shown in Table 4.2

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Abbreviation</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random field x axis</td>
<td>Xm</td>
<td>100 meter</td>
</tr>
<tr>
<td>Random field y axis</td>
<td>Ym</td>
<td>100 meter</td>
</tr>
<tr>
<td>Initial energy of sensor</td>
<td>eo</td>
<td>0.05 Joule</td>
</tr>
<tr>
<td>Total number of sensor</td>
<td>n</td>
<td>100,200</td>
</tr>
<tr>
<td>Transmission energy per bit</td>
<td>Etx</td>
<td>50*1.E-9 Joule</td>
</tr>
<tr>
<td>Receiving energy per bit</td>
<td>Erx</td>
<td>50*1.E-9 Joule</td>
</tr>
<tr>
<td>Free space energy per bit</td>
<td>Es</td>
<td>10*1.E-12 Joule</td>
</tr>
<tr>
<td>Data aggregation energy per bit</td>
<td>EDA</td>
<td>5*1.E-12 Joule</td>
</tr>
<tr>
<td>Advertisement energy per bit</td>
<td>Eadv</td>
<td>50*1.E-12 Joule</td>
</tr>
<tr>
<td>Range of Communication</td>
<td>Rc</td>
<td>20,40,60,80 meters</td>
</tr>
<tr>
<td>Cluster Probability</td>
<td>Cprob</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Figure 4.1 - Performance of MEHEED
Figure 4.2 - Comparative Results HEED, MEDC, MEHEED number of sensor=100 Rc =20

Figure 4.3 - Comparative Results HEED, MEDC, MEHEED number of sensor=100 Rc =40
Figure 4.4 - Comparative Results HEED, MEDC, MEHEED number of sensor=200 Rc =20

Figure 4.5 - Comparative Results HEED, MEDC, MEHEED number of sensor=200 Rc =40
Figure 4.6 - Comparative Results HEED, MEDC, MEHEED number of sensor=100 Rc =60

Figure 4.7 - Comparative Results HEED, MEDC, MEHEED number of sensor=100 Rc =80
Figure 4.8-Comparative Results HEED, MEDC, MEHEED number of sensor=200 Rc=60

Figure 4.9-Comparative Results HEED, MEDC, MEHEED number of sensor=200 Rc=80
Figure 4.10-Comparison of No of dead sensors V/S Rounds when n=100 and Rc=20.

Figure 4.11-Comparison of No of dead sensors V/S Rounds when n=100 and Rc =20 along with detailed table.
Figure 4.12-Comparison of No of dead sensors V/S Rounds when n=100 and Rc=40.

Figure 4.13- Comparison of No of dead sensors V/S Rounds when n=100 and Rc =40 along with detailed table
Figure 4.14-Comparison of No of dead sensors V/S Rounds when n=200 and Rc=20.

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

Figure 4.15-Comparison of No of dead sensors V/S Rounds when n=200 and Rc =20 along with detailed table.
Figure 4.16-Comparison of No of dead sensors V/S Rounds when n=200 and Rc =40.

Figure 4.17- Comparison of No of dead sensors V/S Rounds when n=200 and Rc =40 along with detailed table
Figure 4.18-Comparison of No of dead sensors V/S Rounds when n=100 and Rc=60

Figure 4.19-Comparison of No of dead sensors V/S Rounds when n=100 and Rc =60 along with detailed table
Figure 4.20-Comparison of No of dead sensors V/S Rounds when n=100 and Rc=80

Figure 4.21-Comparison of No of dead sensors V/S Rounds when n=100 and Rc =80 along with detailed table

99
Figure 4.22-Comparison of No of dead sensors V/S Rounds when n=200 and Rc=60

Figure 4.23-Comparison of No of dead sensors V/S Rounds when n=200 and Rc =60 along with detailed table
Figure 4.24—Comparison of No of dead sensors V/S Rounds when n=200 and Rc=80

Figure 4.25—Comparison of No of dead sensors V/S Rounds when n=200 and Rc=80 along with detailed table
4.7 Results and Discussions.

MATLAB results for MEHEED are given in section 4.5. Figure 4.1 is experimental result how MEHEED perform. Network life time is measured as number of alive nodes per round. In results graph Y axis represents alive nodes and X axis represents rounds. Result 4.2 is shown for comparative performance of MEHEED, MEDC and HEED on parameter n=100 and range of communication is taken as 20. Red arc represent performance of HEED, Green arc represent MEDC and Blue Arc is representing MEHEED. Performance is again measured in terms network life time. Results 4.3-4.9 are again performance analysis of MEHEED, MEDC and HEED by varying values of n and \( R_c \) (Range of communication) n is varied on two values 100 and 200. \( R_c \) is also varied by four values which are 20, 40, 60 and 80. From different analysis we concluded that MEHEED perform better infect from MEDC also but as parameter are changed graph of MEHEED come closer to MEDC further close to performance of HEED. After n=200 and \( R_c =40 \) all three graphs overlap each other. Reason for this decrement in graph of MEHEED; is increased cost of communication. When we increase n from 100 to 200 or \( R_c \) from 20 to 40 or to 60 and 80 then their will be more no of sensors under range of communication to each other. More sensors under range more advertisements mean more cost of communication which result decrease in performance. Results 4.10- 4.18 are results of these three protocols for dead nodes calculation after rounds. Dead nodes count values after 100 then 200 up to 1000 is taken for all three protocols; and graphs are plotted.

4.8 Conclusion

MEHEED protocol performs clustering and solves network objectives. It enhance network lifetime of network, reduces energy consumption on communication so that network can work up to long time. This protocol has carried advantage of both protocols one is HEED and other is MEDC. MEHEED protocol posses attribute of fast selection along with simplicity. We had worked and proved its efficiency over previous two protocols. MEHEED perform better in most cases even comparable in worst cases. The reason for be betterment is two fold, one is if selection is based on Chprob then
computations are skipped a number of steps are overcooked, on the second hand if selection cannot be taken on probability basis then message passing as like MEDC protocol will select cluster heads in simple way. Message passing will be done and under a range of communication sensor with maximum residual energy will be chased as cluster head. Selected cluster head will transmit cluster head declaration message to sensors under its range of communication. Sensor that comes under range of communication will act as cluster members. Cluster head will aggregate sensory information from cluster members and transmit that data to base station.