CHAPTER 3

ENERGY AWARE FUZZY CLUSTERING ALGORITHM (EAFCA)

3.1 OVERVIEW

Wireless Sensor Networks (WSNs) exert a pull on the modern research community towards many design challenges, especially, constraints on their lifetimes. Solutions proposed to save energy in WSNs possess their own merits and limitations. The trends evolved from the perspective of improving performance and scalability of conventional clustering approaches. They emerge by adopting cognitive techniques to handle uncertainty and instability present in the application environment. A wireless sensor network consists of homogenous/heterogeneous sensor nodes which work together to accomplish a common set of objectives. These nodes are typically equipped with limited, non-rechargeable power resources which are seldom under human monitoring. The lifetime of a WSN is defined with respect to the application-specific requirements of sensor networks. Invariably, energy preservation in the sensor nodes becomes the dominant and monolithic constraint on achieving lifetime efficiency under all these scenarios.

There has been a substantial quantum of research contributions made on cluster formation process from the perspective of achieving energy efficiency. Many of these approaches form clusters with broad or narrow
boundaries and they exhibit inefficiency owing to the lack of their consideration over the location of the sink. This leads to hot-spot issues, especially in a multi-hop environment. Cluster heads near to the sink faces higher energy depletion compared to the cluster heads which are located far from the sink. Some emerging research works propose unequal clustering in which clusters of different sizes are formed. The clusters near to the sink are smaller in their size compared to the clusters that are far from the sink to reduce intra-cluster relay in these clusters and hence they can be balanced to accommodate the heavier inter-cluster relays.

There have been multiple clustering approaches proposed which are focusing and emphasizing versatile requirements of WSNs. Among them, one of the most traditional clustering approaches proposed by Heinzelman et al. (2000), namely, LEACH rotates the role of cluster heads in a cluster periodically for each round from the energy perspective. Here, a round refers to an interval that exists between two consecutive cluster formation processes. Many variants of the LEACH approach have been observed in the literature focusing various performance constraints. Each of these variants composes of its own merits and demerits towards the QoS requirements of the WSN on a comprehensive perception. The pure probabilistic model used by LEACH to select the cluster head leaves entropy and uncertainty of the solutions abandoned and hence reveals many sensible challenges from the real-time viewpoints. Also, LEACH demands that all sensor nodes should be connected to the sink exactly at 1-hop distance via cluster heads which is infeasible in many sensor applications. Amidst these limitations, LEACH is considered to be a benchmarked protocol to determine the efficiency of contemporary research works owing to its conventional exposure and adoptability to sensor applications.
This work proposes Energy Aware Fuzzy Clustering Algorithm for WSNs which addresses the issues raised above. It combines the advantages of many research works and contributes a hybrid model which benefits from the multifaceted analysis on WSN performance factors. This work contributes a cluster head election process with respect to three constraints: the residual energy, node centrality and 2-hop node coverage. The cluster heads transmit the data to the sink based on the probabilistic based multi-hop relay model.

The lifetime of a sensor network is defined on various degrees depending upon the nature of the application, traffic pattern and the availability of resources. A sensor node conserves energy during transmission, reception and idle states. When a sensor node drains off its total energy, it is considered to be a ‘dead’ node.

Handy et al. (2002) employ three metrics to measure the lifetime of a sensor network: FND, HNA and LND. The first definition among them estimates the lifetime of a WSN till the first node dies in the network. The second definition extends the lifetime of a WSN till half of the population drains off the energy. The final metric scales up the lifetime of a WSN till the last node in the network dies. The definition chosen for a WSN should be justified against the application-specific requirements and pragmatic constraints. Since this research work examines the efficiency of the proposed clustering approach under the functional state of the WSN, FND and HNA metrics are chosen. The proposed work is evaluated against selected representatives of conventional and modern clustering approaches and demonstrates improvement in the lifetime of WSNs with respect to various scenarios.
3.2 PROPOSED SYSTEM

The proposed system has been designed from the perspective of achieving lifetime enhancement through energy reduction in WSNs. The following set of assumptions is made about the characteristics of energy model and network model for the proposed algorithm.

3.2.1 Energy Model

The energy consumption model for this work is adopted from the work done by Kang and Nguyen (2012). Equation (3.1) represents the amount of energy required to transmit ‘L’ bits of data to a distance of ‘d’. The energy components of Equation (3.1) are calculated at transmitter circuitry and RF amplifier.

\[
E_{tx}(L,d) = \begin{cases} 
LE_{elec} + LE_{elec} d^2 & d < d_0 \\
LE_{elec} + LE_{elec} d^4 & d \geq d_0 
\end{cases}
\]  

(3.1)

where ‘d’ is the transmission distance, \( \varepsilon_{fs} \) and \( \varepsilon_{mpf} \) are the amplifier energy factors for free space and multi-path fading channel models, respectively. ‘d_0’ represents the threshold distance that differentiates these two fading models.

Equation (3.2) calculates the amount of energy dissipated in receiving ‘L’ bits of data in the receiver side.

\[
E_{rxr}(L) = LE_{elec}
\]  

(3.2)
3.2.2 Network Model

The WSN environment is modelled with the following set of assumptions:

- Sensor nodes are randomly deployed and unattended after deployment.
- All sensor nodes and the sink are stationary.
- All sensor nodes are equipped with same amount of energy at the initial deployment.
- Every sensor node is assigned with a unique identifier.
- The distance between any two nodes can be computed from the received signal strength and the links are symmetric.

Consider the distributed unconnected graph, $G$ as

$$G= \{V(G), E(G)\}$$  \hspace{1cm} (3.3)$$

where, $V(G)$ is set of vertices and $E(G)$ as set of edges; they are used to connect the different nodes in the network.

$$V(G)= \{s_{n1}, s_{n2}, s_{n3}, \ldots, s_{nn}\}$$  \hspace{1cm} (3.4)$$

where ‘$n$’ denotes the number of nodes in the distributed network.

$$E(G)=\{e_1, e_2, e_3, \ldots, e_n\}$$  \hspace{1cm} (3.5)$$

and ‘$e$’ represents the edge between a pair of nodes in the network.
Ψ (G) denotes the relationship between connected nodes and the edge. For example,

\[ \Psi(e_1) = (n_1, n_2) \]  
(3.6)

The nodes \( n_1 \) and \( n_2 \) are connected through the link ‘\( e_1 \)’ and that is denoted as \( \Psi(e_1) \).

3.3 FUZZY BASED CLUSTERING ALGORITHM

3.3.1 Cluster Formation

During the sensor network deployment phase, the sink broadcasts a beacon signal to all the sensor nodes. The sensor nodes compute the distance to the sink by received signal strength. According to the node density, the sink determines a fraction of ‘\( f \)’ nodes as temporary cluster heads from the network. A threshold ‘\( T \)’ is calculated and communicated to all the sensor nodes for every round to determine the eligibility of a tentative cluster head according to LEACH. Each sensor node computes a random number in the interval (0,1) and compares the same against the threshold ‘\( T \)’. If the computed value is more than the threshold, it declares itself as a cluster head and broadcasts the same to other nodes. Otherwise, it considers itself to be an ordinary cluster member. A wireless sensor network with 2-hop intra-cluster coverage is shown in Figure 3.1.
The fuzzy logic is introduced to elect eligible cluster heads from the set of tentative cluster heads. It is proposed to form the clusters in the WSN in such a way that any node can reach the cluster head with a maximum of 2-hop distance. For every tentative cluster head, the following set of parameters is computed.

- **Remaining Residual Energy (represented as ‘Energy’):**

  This parameter is expected to be higher for an eligible cluster head in a competition phase since it is heavily engaged to intra-cluster and inter-cluster data traffic.
• **Node Degree at its 2-hop Coverage (represented as ‘2-hop ND’):**

This parameter stands for the total number of neighbors in the 2-hop distance from the tentative cluster head. This value is calculated as in Equation (3.7) which is expected to be higher for an appropriate cluster head.

\[
2\text{-hop Node degree} = \frac{|s_{2\text{-hop-nbr}(i)}|}{\# \text{ nodes}}
\]

where \(s_{2\text{-hop-nbr}(i)}\) represents the set consisting of all neighbors of node ‘i’ which can be reached at a maximum of 2-hop distance.

• **Centrality of the Cluster Head (represented as ‘Node Centrality’):**

For an effective cluster head, this parameter should yield low values to reduce energy consumption during the data aggregation and flooding processes. Centrality of a node is calculated using Equation (3.8).

\[
\text{Node Centrality} = \sqrt{\frac{\sum_{j \in s_{2\text{-hop-nbr}(i)}} d^2(i,j)}{|s_{2\text{-hop-nbr}(i)}|}}
\]

where the parameter ‘\(d(i,j)\)’ represents the distance between nodes ‘i’ and ‘j’ in which node \(j\) is a member of the set 2-hop-nbr. The variable ‘\(A\)’ represents the area of the network.

The process of fuzzification maps each crisp input value on the above three parameters for every tentative cluster head to a set of respective fuzzy membership functions. The output variable ‘chance’ represents the possibility of a tentative cluster head to become a cluster head. The fuzzy sets for the input and output variables are presented in Table 3.1. Input fuzzy
Membership functions are presented in Figures 3.2, 3.3 & 3.4. Output fuzzy membership function is provided in Figure 3.5.

### Table 3.1 Fuzzy Sets for Input & Output Variables

<table>
<thead>
<tr>
<th>Energy</th>
<th>2-hop ND</th>
<th>Node Centrality</th>
<th>Chance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Low</td>
<td>Far</td>
<td>Very Weak</td>
</tr>
<tr>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>Weak</td>
</tr>
<tr>
<td>Low</td>
<td>Low</td>
<td>Close</td>
<td>Little Weak</td>
</tr>
<tr>
<td>Low</td>
<td>Medium</td>
<td>Far</td>
<td>Weak</td>
</tr>
<tr>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>Little Weak</td>
</tr>
<tr>
<td>Low</td>
<td>Medium</td>
<td>Close</td>
<td>Little Weak</td>
</tr>
<tr>
<td>Low</td>
<td>High</td>
<td>Far</td>
<td>Little Weak</td>
</tr>
<tr>
<td>Low</td>
<td>High</td>
<td>Medium</td>
<td>Little Weak</td>
</tr>
<tr>
<td>Low</td>
<td>High</td>
<td>Close</td>
<td>Medium</td>
</tr>
<tr>
<td>Medium</td>
<td>Low</td>
<td>Far</td>
<td>Little Weak</td>
</tr>
<tr>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
<td>Little Medium</td>
</tr>
<tr>
<td>Medium</td>
<td>Low</td>
<td>Close</td>
<td>Medium</td>
</tr>
<tr>
<td>Medium</td>
<td>Medium</td>
<td>Far</td>
<td>Little Medium</td>
</tr>
<tr>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Medium</td>
<td>Medium</td>
<td>Close</td>
<td>High Medium</td>
</tr>
<tr>
<td>Medium</td>
<td>High</td>
<td>Far</td>
<td>Medium</td>
</tr>
<tr>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
<td>High Medium</td>
</tr>
<tr>
<td>Medium</td>
<td>High</td>
<td>Close</td>
<td>Little Strong</td>
</tr>
<tr>
<td>High</td>
<td>Low</td>
<td>Far</td>
<td>Medium</td>
</tr>
<tr>
<td>High</td>
<td>Low</td>
<td>Medium</td>
<td>High Medium</td>
</tr>
<tr>
<td>High</td>
<td>Low</td>
<td>Close</td>
<td>Little Strong</td>
</tr>
<tr>
<td>High</td>
<td>Medium</td>
<td>Far</td>
<td>High Medium</td>
</tr>
<tr>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
<td>Little Strong</td>
</tr>
<tr>
<td>High</td>
<td>Medium</td>
<td>Close</td>
<td>Strong</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>Far</td>
<td>Little Strong</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>Strong</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>Close</td>
<td>Very Strong</td>
</tr>
</tbody>
</table>
Figure 3.2  Membership Function for Residual Energy

Figure 3.3  Membership Function for 2-hop Node Degree

Figure 3.4  Membership Function for Node Centrality

Figure 3.5  Membership Function for Chance
Fuzzy if-then rules are processed in the fuzzy inference engine using Mamdani method. For each rule, the respective output is generated. The centre of area method is used in the process of defuzzification to obtain crisp values, from the fuzzy logic rules. Every tentative cluster head broadcasts the value of ‘chance’ to all of its 1-hop neighbours. The same is forwarded to its 2-hop neighbours through the process of flooding and hence the size of each cluster is restricted with 2-hop coverage.

Suppose, a tentative cluster head has received advertisement with a higher value of ‘chance’ from any other competitor within a fixed period of time, then it declares itself to be an ordinary cluster member; otherwise, it declares itself to be a cluster head elected and the same is communicated to all its 1-hop and 2-hop neighbours to form clusters. Now, every ordinary sensor node in the WSN prepares itself to become a member of a cluster. It looks for the arrival of advertisements from the cluster heads. Under these circumstances, an ordinary sensor node has a chance to receive the advertisement from more than one cluster head. In this case, an ordinary sensor node joins the nearest cluster head. In the case of a tie in selecting its cluster head, it joins the cluster head whose advertisement comes first. Thus, clusters are formed in the WSN environment without any degree of overlapping.

3.3.2 Data Aggregation

After the clusters are formed, the required data is generated and disseminated by the members of a cluster. In the experimental environment, all the clusters are formed in such a way that any cluster member can forward its data to the cluster head at a maximum of 2-hop distance. The cluster head is responsible for collecting and aggregating the data received from the sensor nodes.
3.3.3 Inter-cluster Relay

After the data aggregation process is accomplished, the cluster heads have to report the aggregated data to the sink. Unlike LEACH and many conventional algorithms, the proposed algorithm inherits the presence of multi-hop relay between the cluster head and the sink from the unequal clustering algorithm presented by Bagci & Yazicy (2010). This multi-hop relay favors the selection of any one path based on certain probability among multiple choices and the selected path is not necessarily to be an optimal path as followed in many approaches. The repetitive employment of a few popular paths which have been identified as efficient paths causes energy depletion of nodes on these paths and leaves them die soon. This multi-hop relay balances energy dissipation involved in these operations and hence the energy gain of this relay becomes scalable across the network size. Either the cluster head directly delivers the data to the sink or the data reaches the sink through a multi-hop relay in this work.

3.3.4 EAFCA algorithm

The list of steps involved in the proposed Energy Aware Fuzzy based Clustering Algorithm (EAFCA) are summarized as follows:

Algorithm: Energy Aware Fuzzy Clustering Algorithm (EAFCA)

Input: A randomly deployed WSN with N sensor nodes
Output: Clusters with Cluster Heads (CHs)

1. Begin
2. S ← Set of Sensor Nodes, |S| = N
3. Status (S[i]) ← M, i=1,2,…N
4. T ← Number of Tentative CHs
5. CH ← Set of Temporary Cluster Heads | CH[i], i=1,2,…T selected from S
6. S ← S - CH
7. Chance[j]← Probability of CH[j] to become a Cluster Head, j=1,2,…..,T
8. For every Tentative Cluster Head CH[j], j=1,2…T
9. Calculate Chance[j] using fuzzy if-then mapping rules
10. Broadcast Advertisement (Chance[j]) to all its 1-hop and 2-hop neighbours
11. While (timer)
12. If (Advertisement from any CH[x] & (Chance[j] < Chance[x]))
13. Add CH[j] to S
14. CH← CH- {CH[j]}
15. End if
16. Else
17. Status (CH[j]) ← H
18. Broadcast Advertisement (Status (CH[j])) to all its 1-hop and 2-hop neighbours
19. End else
20. End While
21. End For
22. For every sensor node S[i]
23. If Advertisement(Status(CH[k])) received form exactly one Cluster Head CH[k]
24. Add S[i] to CH[k]
25. End if
// To avoid overlapping of clusters
26. Else If (CHAdvertisement() message received from N number of Cluster Heads)