Enhanced Life Time of Wireless Sensor Networks Using Dual Hop Clustering

Jubair Ahmed. L.¹, Manikandan.V.² & R. Sathishkumar³
¹Assistance Professor ME,Ph.D, Department of Electronics and Communication Engineering, Sri Eshwar College of Engineering, Coimbatore-641202
²PG SCHOLAR M.E., Department of Electronics and Communication Engineering, Sri Eshwar College of Engineering, Coimbatore-641202.
³CSS CORP, Chennai.

Abstract— Wireless sensor node is popular methods used to create mobile network. The approach for the mobile users to collect data over a long communications. Wireless Sensor Network designs consist of static nodes which are closely positioned completed a sensing area on clustering of nodes and mobile data are collecting. A clustering algorithms done clustering and data aggregation for sensor network. A WSN lifetime is dependent on the energy of individual sensor nodes therefore efficient utilization of each node’s energy is a vital issue. Dual cluster heads are implemented by using the load balancing algorithm for the saving the energy of nodes. Multiple cluster heads and sink also used in this approach to reduce energy Consumption of each nodes. Data aggregation from the all nodes given to the sink and node life time increased then whole network life time also increased.

Keywords: Clustering Head, Sink, Load Balancing Clustering, leach.

I. INTRODUCTION

Wireless sensor networks (WSNs) are large collections of small sensor devices that can be an effective tool for collecting data from various environments. Each sensor sends its data to Base Station (BS), and finally BS sends these data to end user. Clustering is considered as an effective approach to provide better data gathering and scalability for large sensor networks. Clustering is an efficient method for providing better data aggregation and scalability for Wireless Sensor Networks. WSN’s have less performance due to obstacle presence. After sensors form into autonomous organizations, those sensors near the data sink typically deplete their batteries much faster than others due to more relaying traffic. When sensors around the data sink deplete their energy, network connectivity and coverage may not be guaranteed. Due to these constraints, it is crucial to design an energy-efficient data collection scheme that consumes energy uniformly across the sensing field to achieve long network lifetime [1]. Furthermore, as sensing data in some applications are time-sensitive, data collection may be required to be performed within a specified time frame. Therefore, an efficient, large-scale data collection scheme should aim at good scalability, long network lifetime and low data latency. Based on these observations, propose a three-layer mobile data collection framework, named Load Balanced Clustering and Dual Data Uploading (LBC-DDU). The main motivation is to utilize distributed clustering for scalability, to employ mobility for energy saving and uniform energy consumption, and to exploit Multi-User Multiple-Input and Multiple-Output (MU-MIMO) technique for concurrent data uploading to shorten latency. The main contributions of this work can be summarized as follows. First, we propose a distributed algorithm to organize sensors into clusters, where each cluster has multiple cluster heads. In contrast to clustering techniques proposed in previous works our algorithm balances the load of intra-cluster aggregation and enables dual data uploading between multiple cluster heads and the mobile collector. Second, multiple cluster heads within a cluster can collaborate with each other to perform energy efficient inter-cluster transmissions.

Fig 1. Neighboring nodes in Routing Table of node in WISNs for manufacturing machines.
clusters are only used to route small-sized identification (ID) information of cluster heads to the mobile collector for optimizing the data collection tour. Third, we deploy a mobile collector with two antennas (called SenCar) to allow concurrent uploading from two cluster heads by using MUMIMO communication. The SenCar collects data from the cluster heads by visiting each cluster. It chooses the stop locations inside each cluster and determines the sequence to visit them, such that data collection can be done in minimum time. Our work mainly distinguishes from other mobile collection schemes in the utilization of MUMIMO technique, which enables dual data uploading to shorten data transmission latency. We coordinate the mobility of SenCar to fully enjoy the benefits of dual data uploading, which ultimately leads to a data collection tour with both short moving trajectory and short data uploading time.

II RELATED WORKS
A. Clustering Metrics
The literatures have proposed many metrics for clusterhead election to construct an efficient cluster structure. The representative metrics include node identifier (ID) [8], node degree [10], [12], [3], and residual energy [3], [11]. In general, only one clusterhead can exist in a cluster, and each node has a unique ID. This motivates researchers to consider the node ID as the metric for clusterhead election, that is, a node having the largest or smallest ID value among nodes in its proximity is elected as a clusterhead. The Lowest ID Cluster (LIC) algorithm [8] is an ID-based clustering mechanism, where the ID of the selected clusterhead is smaller than that of the other cluster members. Previous studies have proposed an alternative clusterhead election strategy based on the node degree, which indicates the number of nodes in a node’s communication range [10], [12], [13]. Each node periodically exchanges messages (e.g., HELLO message) with all one-hop neighbors, and counts the number of messages received back to measure the node degree. The random-based clusterhead election approach is likely to select a clusterhead that has no neighbors, thereby failing to construct a cluster structure and discover a routing path. Selecting a node with a smaller node degree value to become a clusterhead may create a large number of clusters. This generates a considerable amount of overhead for cluster maintenance and a long delivery delay. Therefore, proposed a Highest Connectivity Clustering (HCC) algorithm, where a node is selected as the clusterhead if its degree (i.e., connectivity) is larger than that of all its neighbors. In case of a tie, the node with the lowest ID prevails. In general, sensor nodes are energy-constrained, and selecting a node with more residual energy as the clusterhead is an intuitive strategy in clustering [3], [14]. The Hybrid Energy-Efficient Distributed (HEED)

Clustering Approach [3] uses a cost function to evaluate the qualification of a node for a clusterhead. Assume that the batteries of all nodes are initially fully charged. The cost function determines the probability of becoming a clusterhead, $C_{prob}$, which can be derived as

$$C_{prob} = \frac{E_{res} \times E_{ini}}{E_{ini}}$$

where $C_{prob}$ is the initial percentage of clusterheads in the network, $E_{res}$ is the estimated current residual energy, and $E_{ini}$ is the initial energy (i.e., maximum energy) of a node. system, the more residual energy a node has, the higher $C_{prob}$ the node derives. That is, the node with the maximum amount of residual energy in its proximity will successfully become a clusterhead.

B. Passive Clustering (PC) Technique
An active clustering techniques, but Kwon and Gerla proposed a passive clustering (PC) technique for construction of a cluster structure [12]. By using ongoing data packets instead of extra explicit control packets, the PC can reduce the control overhead during constructing and maintaining clusters. The PC technique uses five external states to represent a node’s role in a cluster, and each node possesses an external state. The external states include initial (IN), ordinary (OD), clusterhead (CH), gateway (GW), and distributed gateway (D_GW). The PC technique also introduces two internal states, clusterhead ready (CH_R) and gateway ready (GW_R), to represent the tentative role of a node. When a node in the external state receives data packets, it may change its current state. A node in the internal state must enter the external state when it sends out a data packet. For the lack of space, the rules of state transition in the PC technique can be obtained in [12]. The PC technique proposes two innovative mechanisms: First Declaration Wins mechanism and Gateway Selection Heuristic mechanism, which are used to determine CH and GW nodes.

C. Energy-efficient clustering (ec)
A multihop data collection scenario in a WSN with uniformly distributed node locations. Each sensor node makes observations, produces a single data packet, and then transmits this packet to its associated CH. Then, each CH node collects the observation packets from its associated member nodes and combines them to produce a single summary packet representing the cluster. Summary packets travel through the network’s CH-backbone towards the sink in multiple hops. This three-step process is referred to as a single data collection round (DCR) of the entire WSN operation.

1) Trade-offs: Equalization of node energy consumption levels in a multihop data collection scenario has two tradeoffs:

(a) There is a higher traffic load on nodes closer to the data sink in terms of hop-distance.
D. Multihop data collection protocol for WSNs

Energy levels are maintained at different regions. Our task is now to ensure that similar consumption. With this, the lifetimes of all sensors in the region have “approximately” equal rates of energy consumption. Hence, having smaller clusters leads to a larger intercluster communication cost. Therefore, the analysis should take into account the hop distances to the sink node.

2) Hop distances to the sink: The hop distance to a sink node in a network area with length X and width W, where the sink node is located at one edge, forms a wave-like propagation pattern [19] outwards from the sink. Figure 1 illustrates this pattern for a sample randomly deployed network, where nodes at different hop distances to the sink are denoted by different symbols. The area in which nodes of a particular hop distance i reside can approximately be represented by a rectangular region Ri. The widths of these regions may not be equivalent and are random variables depending on node locations and sensor communication range. However, we can denote the average region width by a.

3) Approximate equalization of energy levels: (i): Accumulation of packets from outer regions towards the data sink creates higher traffic loads at closer locations to the sink. Since this load is distributed among the sensors of each region via rotation of the CH-role, sensors in a particular region have “approximately” equal rates of energy consumption. With this, the lifetimes of all sensors in region Ri, are treated as the same, and denoted by γ (i). This is the reason why we claim that our approach provides approximate equalization of node energy levels. Our task is now to ensure that similar energy levels are maintained at different regions throughout the lifetime of the WSN.

D. Multihop data collection protocol for WSNs

An energy-efficient multihop data routing solution for WSNs organized as clusters is briefly outlined. There are three reasons for presenting such a protocol:

1) To complement EC’s energy equalization and conservation features with a protocol that also targets at energy efficiency and reduces its overall energy consumption level via using less control messages.

2) To make comparisons of EC with existing clustering solutions that target at energy efficiency in multihop data delivery.

3) To understand whether EC actually achieves energy efficiency and equalization with its output probability values for CH-selection.

A. Cluster head selection

Cluster formation is performed as a distributed algorithm at the beginning of each data collection round, DCR. This involves election of CH nodes among a set of candidates followed by node-CH associations.

1) Selection of CH-candidates: To determine the CH candidates, a probability scale is assigned to each sensor. According to this value, each sensor decides on becoming a CH-candidate. Basically, the probability to become a CH-candidate, T, is scaled by the ratio of initial sensor energy level to the average initial energy of the network, E0. For a node j in region Ri, the resulting probability becomes

\[ P(j) = \frac{T \cdot E_{0j}}{E_{0}} \]

Computation of \( P(j) \) is performed only once right after network initialization. At the beginning of each DCR, each node j picks a random number on [0 1]. If the number is less than \( P(j) \), then the node is a CH-candidate. With this mechanism, approximately a ratio \( T \) of all nodes are elected as CH-candidates. In simulations, we pick \( T = 10\% \) as in [12].

The candidate selection probability \( P(j) \):

The selection of \( P(j) \) would be more up-to-date if the residual node energy levels \( E_{0j} \) are considered instead of the initial energy levels \( E_{0} \), hence \( P(j) = \frac{T \cdot E_{0j}}{E_{0}} \), where \( E_{0} \) is the average residual energy level within a region. However, this would require each node to notify all others in its region of its energy value, which could only be achieved by region wide broadcasts; a quite high message overhead. Alternatively, a central node in each region could gather the energy levels and then distribute the average value to all sensors in the region, which is a slightly better scheme. Nevertheless, this would add the additional complexity of choosing and replacing such a central node. Another method would be the use of counters at each node to keep track of the number of times they take the CH role. However, this also requires later negotiations among nodes. To avoid all these issues, we use the initial energy levels for selecting the CH-candidate nodes. Since the frequency of being selected as a CH-candidate is proportional to the initial energy levels and the CHs are eventually selected among these candidates, the resulting frequency of having the CH-role and the corresponding energy consumption are on the average approximately proportional to the initial energy levels. Therefore, this choice is a reasonable method towards balancing energy consumption levels while preventing additional overhead. Note that node residual energy levels are taken into account during the selection of the actual CHs, as explained next.

A. Selection of CHs from CH-candidates:

Upon being selected, each CH-candidate in region Ri transmits a “CHAdvertisement” packet and advertises its residual energy level within a
neighborhood of radius \( r_i \) is determined by the EC algorithm and is related to \( p_i \) by Equation 1. A CH-candidate monitors advertisements from other candidates and defers from acting as a CH if a higher energy level is reported by another. Eventually, the candidates with the highest residual energy among their neighboring CH-candidates become the CHs during that particular DCR. (If a CH-candidate receives no advertisement packets for a period of \( T_{wait} \), it automatically becomes a CH node.) This mechanism enables the choice of the actual CH nodes to be based on the most recent sensor energy stocks. The pseudocode of the algorithm for CH selection is available.

### B. Cluster formation

After the CHs are elected, each CH transmits a “CH-announcement” packet within an area of transmission radius \( \alpha r_i \) and informs other sensors of its availability as a CH. This CH-announcement range is a multiple of \( \alpha r_i \)

\[ \Pi\sigma p_i = \alpha r_i, \]

selected to ensure that each non-CH node receives at least one announcement packet and can associate to a CH. To ensure reception of announcement packets by other nodes, a straightforward method is to send region-wide broadcasts. However, this potentially causes high transmission energy cost; a fine-tuned value is required. Thus, \( \alpha \) is a system parameter tuned to achieve high CH-association probability for non-CH nodes while avoiding an unnecessarily large transmission range. Considering a uniform distribution of CH nodes in each region, the number of nodes in a given area has a Poisson distribution. Hence, the probability that a non-CH node has at least one CH neighbour within a circular area of radius \( \alpha r_i \) in region \( R_i \) is

\[ 1 - e^{-\alpha p_i \pi r_i^2}. \]

To ensure a high rate of CH-association, we seek at least 99% average connectivity probability. Each sensor may collect announcement packets from multiple CHs and selects the CH that has generated the announcement packet with the highest RSSI as the ideal CH to associate to. Nodes associate to CHs via sending a “CH-association” request and upon reception of a subsequent “CH-confirmation”. At the end of the clustering formation phase, there may still be a few sensors that have not joined any clusters as they may have not received any announcement packets. To recover from such cases, a sensor with no CH-association gradually increases its transmission range and seeks the closest CH to associate to.

### C. Message complexity of Clustering

In a WSN of \( N \) nodes, \( NT \) nodes advertise as CH-candidates, producing a total of \( NT \) messages. Eventually, \( M \) CH nodes are selected, which then announce their role as a CH with a total of \( M \) CH-announcement messages. Sensor nodes choose a CH to join and send CH-association requests, incurring an additional cost of \( N - M \). For each request, a CH-confirmation message is generated. As a result, the total message complexity in cluster formation is approximately

\[ NT + M + 2(N - M) = (2 + T)N - M = 0(N). \]

### D. Distributed inter-cluster routing

The routing algorithm is based on two ideas: (1) Reducing the overhead in route discovery, (2) Balancing energy consumption among all CHs. To achieve these goals, a simple scheme is used: Basically, a CH node in region \( R_i \) chooses its next hop towards the sink in the neighbor rectangular region, \( R_{i-1} \). The CH transmits a route request packet with a range of \( \sqrt{W_2 + 4\pi z} \), sufficiently large to cover \( R_{i-1} \). Each receiving CH in \( R_{i-1} \) generates a reply packet and starts a route reply timer with an expiration time inversely proportional to its residual energy level. The first node that has an expired timer actually makes the transmission of a route reply packet back to the requester CH in \( R_i \), while the rest quietly cancel their timers upon hearing this reply. This guarantees that a single reply packet is sent and thus prevents excessive message overhead. Furthermore, by considering the residual energy levels, priority is given to nodes with higher resources. A policy towards balancing energy consumption in the entire network.

### III. LEACH Protocol Architecture

The requirements of wireless microsensor networks, we developed LEACH, an application-specific protocol architecture [10], [11]. The application that typical microsensor networks support is the monitoring of a remote environment. Since individual nodes’ data are often correlated in a microsensor network, the end user does not require all the (redundant) data; rather, the end user needs a high-level function of the data that describes the events occurring in the environment. Because the correlation is strongest between data signals from nodes located close to each other, we chose to use a clustering infrastructure as the basis for LEACH. This allows all data from nodes within the cluster to be processed locally, reducing the data set that needs to be transmitted to the end user. In particular, data aggregation techniques can be used to combine several correlated data signals into a smaller set of information that maintains the effective data (i.e., the information content) of the original signals [9]. Therefore, much less actual data needs to be transmitted from the cluster to the base station (BS). For the development of LEACH, we made some assumptions about the sensor nodes and the underlying network model. For the sensor nodes, we assume that all nodes can transmit with enough power to reach the BS if
needed, that the nodes can use power control to vary the amount of transmit power, and that each node has the computational power to support different. These assumptions are reasonable due to technological advances in radio hardware and low-power computing. For the network, we use a model where nodes always have data to send to the end user and nodes located close to each other have correlated data. Although LEACH is optimized for this situation, it will continue to work if it were not true. In Section V, we discuss ways in which LEACH may be improved when these assumptions do not hold. In LEACH, the nodes organize themselves into local clusters, with one node acting as the cluster head. All non-cluster head nodes transmit their data to the cluster head, while the cluster head node receives data from all the cluster members, performs signal processing functions on the data (e.g., data aggregation), and transmits data to the remote BS. Therefore, being a cluster head node is much more energy intensive than being a noncluster head node. If the cluster heads were chosen a priori and fixed throughout the system lifetime, these nodes would quickly use up their limited energy. Once the cluster head runs out of energy, it is no longer operational, and all the nodes that belong to the cluster lose communication ability. Thus, LEACH incorporates randomized rotation of the high-energy cluster head position among the sensors to avoid draining the battery of any one sensor in the network. In this way, the energy load of being a cluster head is evenly distributed among the nodes.

IV SIMULATION RESULTS

The following table 1.1 shows the, and the Network lifetime, for different percentages of CHs in network and average energy consumption in a network.

<table>
<thead>
<tr>
<th>%Cluster formation</th>
<th>First Node Dead in Round Number</th>
<th>% Overhead energy consumed</th>
<th>Energy consumed in 1000 rounds (J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>905</td>
<td>74</td>
<td>182.1914</td>
</tr>
<tr>
<td>5</td>
<td>874</td>
<td>22</td>
<td>198.0256</td>
</tr>
<tr>
<td>10</td>
<td>937</td>
<td>21</td>
<td>198.4349</td>
</tr>
<tr>
<td>15</td>
<td>915</td>
<td>24</td>
<td>209.8809</td>
</tr>
<tr>
<td>20</td>
<td>880</td>
<td>26</td>
<td>220.8879</td>
</tr>
<tr>
<td>25</td>
<td>819</td>
<td>30</td>
<td>235.4284</td>
</tr>
<tr>
<td>30</td>
<td>738</td>
<td>31</td>
<td>243.5268</td>
</tr>
</tbody>
</table>

Table 1.1 Average Overhead Energy of the network

From the results it can be seen that the average overhead energy consumed increases when the number of CH formed increases or decreases from an optimum value of CH percentage. Also the number of rounds in which the first node die starts increasing with the increase of CH percentage but decreases after the optimum value of CH percentage. This optimum value of CH percentage in this case is 5 to 10 percent. This is because when number of CHs increases more than an optimum value say 10% of the total number of node then there are other things which start affecting the energy utilizations like number of broadcasts and transmission of control packets in contention period therefore energy consumed in overhead also starts increasing.

V. CONCLUSION

The energy consumption in node a hierarchal and random cluster head selection protocols of WSN had been considered as an insignificant overhead in the previous literatures. The results show that load balancing algorithm can greatly reduce energy consumptions by improving routing problems on nodes and balancing capacity among cluster heads, which achieves 20 per cent less data gathering time compared to SISO mobile data.
gathering and over 60 per cent energy saving on cluster heads. The energy overhead and discovered the results with different numbers of cluster heads in the structure.

REFERENCES


