ADAPTIVE LOAD BALANCED CLUSTERING SCHEME FOR MOBILE AD-HOC NETWORKS

Naghma Khatoon¹ and Amritanjali²

¹Dept. of Computer Science and Engineering, BIT Mesra, Ranchi, India
²Dept. of Computer Science and Engineering, BIT Mesra, Ranchi, India

ABSTRACT:

Mobile ad-hoc networks (MANETs) are wireless networks without any fixed infrastructure where nodes move randomly and are interconnected via wireless media. In recent years MANET become one of the most prevailing area of research because of its eminent characteristics. Clustering in MANET is the most efficient technique to improve scalability and network lifetime. However, load balancing is still a major concern for optimizing energy consumption and packet loss. In this paper, an adaptive load balanced clustering scheme has been proposed where we are emphasizing not only on the problem of appropriate cluster head election but also on assigning mobile nodes to cluster heads efficiently based on the current weight of cluster heads. Thus, our proposed algorithm revolves around three benefactions for adaptive load balanced clustering i.e. the selection of most appropriate node to serve as cluster head, minimizing the overall load to cluster head and maintaining size of each cluster which improve network lifetime with minimal overhead.

Keywords: ad-hoc networks, clustering, cluster head, load balancing.

[1] INTRODUCTION

Mobile ad-hoc networks (MANETs) in last decades have gained wide acceptance. Due to its inevitable characteristics of self-configuration, flexibility, low deployment cost and distributed nature, it is used in a wide range of applications like search and rescue operations, disaster
recovery, military services and communication between vehicles and roadside equipments as vehicular ad-hoc networks. Such networks may operate by themselves or may be connected to the larger Internet [Figure-1].

A MANET can be modelled as an undirected graph \( G = (V, E) \), where \( V \) is the set of mobile nodes and \( E \) is the set of links that exist between the nodes. There exist a bidirectional link \( E_{ij} \) between the nodes \( v_i \) and \( v_j \) when the distance between the nodes, \( \text{dist}(v_i, v_j) < t_{range} \) (transmission range) of the nodes. Clustering can be thought as a graph partitioning problem with some added constraints. As the underlying graph does not show any regular structure, partitioning the graph optimally (i.e., with minimum number of partitions) with respect to certain parameters becomes an NP-hard problem.

![Figure 1. A mobile ad-hoc network](image1)

Clustering is an important approach which solve many problems in MANET and provide network scalability and increases its lifetime. Here nodes are divided into virtual groups called clusters with a cluster head (CH) in each cluster which serves as a local coordinator for its cluster [Figure-2]. Communication from source to destination is done via CHs and gateway nodes which are within the transmission range of more than one CHs and thus conserve energy of other nodes. Also cluster based MANET improve network management as route setup is localized with clusters, thus reduces the routing table of other nodes.

![Figure 2. An illustration of clusters in MANET](image2)
However, CHs bear extra work load of intra-cluster and inter-cluster transmission. This results in early depletion of energy and death of CHs which ultimately partition the network and degrade network lifetime. Thus, designing an efficient clustering algorithm becomes one of the most focused area of communication research in previous few years in MANET. Many researchers have proposed various clustering algorithms where CHs are selected based on the combination of different parameters like node degree, remaining battery energy, mobility and the sum of distances with all the neighbors. The election of the most eligible node to serve as CH doesn’t only ensure solution to increase network lifetime in such networks where mobility and battery constraints are the most challenging issues. Improper creation of clusters may result in overloading of some CHs. So, balancing loads among CHs at the time of cluster formation itself should be taken into consideration. For load balancing, a number of works have been proposed in [1-5] which effectively improve the network performance and it became an extensive interest in the communication research world. In this paper we are emphasizing on the problems of appropriate CH selection, load balanced cluster creation and periodic cluster maintenance for prolonging network lifetime with better performance. The main beneficences of our proposed Adaptive Load Balanced Clustering (ALBC) scheme for MANET are as follows:

- The affiliation of nodes to a CH is on the basis of the current weight of CH which adaptively make nodes to join a CH with minimum weight for balancing loads on them.
- Minimizing load on each CH increases their lifetime which in turn improve network lifetime.

[2] THE PROPOSED WORK

The proposed ALBC basically consists of three phases i.e. CH election, cluster formation and cluster maintenance [Figure-3]. In the CH election phase, each node (say $n_i$) calculates its combined weight using equation (1) on the basis of its degree-difference ($\Delta d_{n_i}$), sum of distances with all its neighbors ($D_{n_i}$), mobility ($M_{n_i}$) and energy consumed by it ($E_{n_i}$) [6]. A node with minimum combined weight is selected to act as a CH.

$$W_{n_i} = c_1 \Delta d_{n_i} + c_2 D_{n_i} + c_3 M_{n_i} + c_4 E_{n_i}$$

Here, $c_1, c_2, c_3$ and $c_4$ are the weighing factors such that $c_1 + c_2 + c_3 + c_4 = 1$. The value of these factors depend upon the corresponding system parameters.
Once a CH is decided, the second phase is cluster formation. Each time a node is added to its corresponding CH, the new weight of the CH is calculated. This new weight is the current weight of the CH along with the weight incremented due to the affiliation of a node to it. In the third phase i.e. maintenance phase CHs are monitored periodically so that no cluster should get overloaded or under loaded. If the size of any cluster is very small which depends upon the network scenario, their members are merged with neighboring clusters to form a cluster with ideal number of nodes. The system model and detailed description of the phases of ALBC is given in the subsequent sections.

[2.1] SYSTEM MODEL AND TERMINOLOGIES FOR ALBC

We assume a MANET model consisting of mobile nodes that move randomly. The following notations have been used to formulate our proposed work:

- Let \( N = \{n_1, n_2, ..., n_k\} \) be the set of mobile nodes.
- \( \Psi = \{ch_1, ch_2, ..., ch_m\} \) be the set of elected CHs and \( T_{range}(ch_j) \) is the transmission range of CH \( ch_j \).
- \( W_i \) is the weight of node \( n_i \).
- Let \( G_i \) be the set of CHs which are within the communication range of node \( n_i \). For example, if \( G_i = \{ch_1, ch_4, ch_5\} \), it means that node \( n_i \) can be assigned to anyone of CHs \( ch_1, ch_4 \) or \( ch_5 \) depending upon the current load of these CHs.
- \( R_{set} \) is the set of restricted nodes, which can communicate with one and only one CH.
- \( O_{set} \) is the set of open nodes which are in the range of more than one CHs.
- Let \( CW(ch_a) \) be the current weight of a CH \( ch_a \) after the assignment of nodes which are only in the range of \( ch_a \).
- Let \( T_{upper} \) and \( T_{lower} \) be the maximum and minimum number of possible nodes within a cluster respectively.
[2.2] CLUSTER FORMATION PHASE OF ALBC

In the proposed ALBC, the CH election procedure is same as used in weighted clustering algorithm (WCA) [6] using equation (1). Once CHs are selected, our aim is how to assign nodes to them so as to minimize the overall maximum load on them. This is the initialization phase. Each time a node from \( R_{\text{set}} \) of a CH \( c_{h} \) is considered and after its assignment to its corresponding CH, the incremented weight on the CH is calculated using equation (2) as follows:

\[
\Delta W = c_1 \Delta d + c_2 \text{dist} \left( c_{h}, n_i \right) + c_3 \Delta M_{c_{h}} + c_4 \Delta E_{c_{h}} \quad \forall \ n_i \in \ R_{\text{set}}(c_{h})
\] (2)

\[
CW_{c_{h}} = W_{c_{h}} + \Delta W
\] (3)

Here, \( \Delta W \) is the change in weight of CH \( c_{h} \) after the assignment of a node \( n_i \) to it, which is the summation of change in its degree-difference, distance with \( n_i \), change in mobility and energy consumed by it. \( CW_{c_{h}} \) is the current weight of the CH \( c_{h} \) after the assignment of a node \( n_i \) to it and \( W_{c_{h}} \) is its weight before the assignment of node.

The new weight of CH is the sum of its previous weight and the weight incremented due to the assignment of a node from the \( R_{\text{set}} \) of CH as given in equations (3). As nodes in \( R_{\text{set}} \) are in the range of only one CH, it is assigned with its corresponding CH. The nodes which are in the range of more than one CHs i.e. nodes which belongs to \( O_{\text{set}} \) are assigned to the CH which have minimum weight as shown in algorithm in [figure-4].

**Input:** (1) A set of nodes \( N = \{n_1, n_2, \ldots, n_k\} \) and a set of elected CHs \( \Psi = \{c_{h_1}, c_{h_2}, \ldots, c_{h_m}\} \). (2). For each \( n_i \), a set of CHs \( G_i \) which are in the range of \( n_i \).

**Output:** An assignment \( A:n \rightarrow \Psi \) in such a way so as to minimize the maximum load on CHs.

---

**Step 1:** While \( (R_{\text{set}} \neq \text{NULL}) \) do
1.1: Assign successive node \( n_i \) to their corresponding CH \( c_{h} \) and delete \( n_i \) from \( R_{\text{set}} \).
1.2: \( CW_{c_{h}} = W_{c_{h}} + \Delta W \)
end

**Step 2:** Sort the nodes belong to \( O_{\text{set}} \) in increasing order on the number of possible CHs they are in range with.

**Step 3:** While \( (O_{\text{set}} \neq \text{NULL}) \) do
3.1: select subsequent node \( n_i \) from \( O_{\text{set}} \)
3.2: select CH \( c_{h} \) from \( G_i \) with minimum current weight (CW). If two CHs have same CW, randomly select any CH.
3.3: assign \( n_i \), to \( c_{h} \) and delete \( n_i \) from \( O_{\text{set}} \).
3.4: update CW of \( c_{h} \) using equation (3).
end

Stop.

---

**Figure 4.** Cluster formation phase of ALBC
[2.3] MONITORING PHASE OF ALBC

After clusters are created, CHs are monitored periodically. If there are many clusters with only few nodes, it is a better option to merge the nodes of small clusters to create a cluster with ideal number of nodes. For this we are using two predefined thresholds i.e. $T_{\text{upper}}$ and $T_{\text{lower}}$. When a CH finds that its size is less than $T_{\text{lower}}$, it sends “RE-AFFILIATE” message to all its neighbors and dissolve its cluster. On receiving “RE-AFFILIATE” message, all members of that cluster send a “JOIN” message to its neighboring nodes. If any neighboring CH receives a “JOIN” message, it check its cluster information table. It is a table maintained by all CHs that contain information like cluster head ID, size of cluster and member nodes of cluster. If it finds that its size is less than $T_{\text{upper}}$, it send “ACCEPT” message to it and if not it will send a “REJECT” message. If the node doesn’t get “ACCEPT” message from any CH before timeout, it will declare itself as a CH [Figure-5]. This approach ultimately reduces the requirement of re-clustering as loads on CHs are balanced.

| Input: (1) A set of nodes $N = \{n_1, n_2, \ldots, n_k\}$. (2) a set of elected CHs $\Psi = \{c_1, c_2, \ldots, c_m\}$. (3) Thresholds $T_{\text{upper}}$ and $T_{\text{lower}}$.  
Output: Load balanced clusters. |
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1:</strong> Each CH check its size periodically from its cluster information table which is maintained by them.</td>
</tr>
<tr>
<td><strong>Step 2:</strong> if current size of any CH, $ch_a &lt; T_{\text{lower}}$ then</td>
</tr>
<tr>
<td>2.1: $ch_a$ sends a “RE-AFFILIATE” message to all its neighbors and dissolve its cluster.</td>
</tr>
<tr>
<td>2.2: for each of its neighbors and itself do</td>
</tr>
<tr>
<td>2.3: send a “JOIN” message to its neighbors.</td>
</tr>
<tr>
<td>2.4: If any CH, $ch_k$ receives “JOIN” message then</td>
</tr>
<tr>
<td>2.4.1: it check its size from cluster information table.</td>
</tr>
<tr>
<td>2.4.2: If size of $ch_k &lt; T_{\text{upper}}$ then</td>
</tr>
<tr>
<td>2.4.3: it will send an “ACCEPT” message</td>
</tr>
<tr>
<td>2.4.4: else a “REJECT” message</td>
</tr>
<tr>
<td>2.5: else it is an isolated node, so declare itself as CH after timeout.</td>
</tr>
<tr>
<td>\end</td>
</tr>
<tr>
<td>\end</td>
</tr>
</tbody>
</table>

Figure: 5. Maintenance phase of ALBC
[3] CONCLUSION

In this paper we have proposed Adaptive Load Balanced Clustering (ALBC) scheme for MANET. Here, we mainly focused on adaptively creating clusters with load balancing. For this, nodes with good performance parameters are selected to act as CHs. Nodes which are more mobile and having less remaining battery energy will be eliminated from becoming a CH at the initial stage of clustering which reduces computation overhead. After CH selection, clusters are created by assigning nodes in such a way that no CH get overloaded or under loaded. This is achieved in cluster formation and maintenance phase. In cluster formation phase, each time a restricted node is assigned to its corresponding CH, the weight of that CH is updated so that nodes from open set are only assigned to a CH with minimum current weight. In the maintenance phase, the ideal cluster size is periodically maintained. Thus, our proposed algorithm efficiently balanced loads on CHs and increases network lifetime effectively. In the future, we improve the algorithm by considering congestion control for adaptive routing in MANET.

REFERENCES