An Associativity Based Energy Aware Clustering Technique for Mobile Ad Hoc Networks

Tahrima Rahman and A.M.A. Elman Bashar

Department of Computer Science & Engineering
Eastern University, Bangladesh
{tahrima,elman}@easternuni.edu.bd

Abstract. Node clustering is a widely used approach to address the scalability issue of large-scale mobile ad hoc networks(MANETs). It eases the implementation of routing and resource management by constructing an abstract hierarchy of the flat network architecture of MANETs. Its effectiveness however depends largely on the clusters stability which is measured by the lifetime of the clusters. The aim of this paper is to propose a fully distributed clustering algorithm that addresses the stability of mobile nodes in terms of neighborhood associativity and remaining energy. The algorithm maximizes the cluster lifetime by choosing those nodes as clusterheads which have the highest neighborhood associativity and remaining energy. It also tries to minimize the number of clusters in the network by giving priority to higher degree nodes. Simulation results show that the algorithm identifies clusters that are stable with respect to cluster lifetime and the frequency of reaffiliations of mobile nodes.

Keywords: Mobile Ad Hoc Networks, Clustering Algorithm, Associativity.

1 Introduction

Mobile ad hoc networks are autonomous systems of mobile hosts that communicate by multi-hop wireless links without any fixed infrastructure or predetermined connectivity. There are no specialized routers for path discovery and packet routing in such networks. Instead intermediate nodes between a source and destination act as routers by relaying packets between them. These intermediate nodes are mobile in a MANET. Mobility of such nodes cause link failures which in turn cause established routing paths between the source and destination to break. Once an established path is broken a new path has to be discovered. Path discovery algorithms incur extra overhead in the network. This problem becomes acute as mobility increases in the network. For this reason, it is always desirable to select a path that is more stable over time. One solution to this problem is to partition the network into group of clusters[1]. By keeping nodes with a stable neighborhood in the same cluster, the topology within a cluster becomes less dynamic. This minimizes link breakage and packet loss. The number

N. Meghanathan et al. (Eds.): NeCoM, WiMoN, and WeST 2010, CCIS 90, pp. 359–368, 2010.

of nodes in a cluster is smaller than the number of nodes in the entire network. Thus each node needs to store only a partial information of the entire network topology. This reduces the number of entries in the routing table and the exchange of routing information between nodes. Thus clustering helps to mitigate topology information and improves network scalability.

Under a cluster structure, mobile nodes may be assigned different roles, such as clusterheads, cluster gateways or cluster members. A clusterhead is elected to serve as a local coordinator for its cluster performing intra-cluster transmission arrangement, data forwarding, and so on. A node that has a packet to send to another node can obtain routing information from its clusterhead. A cluster gateway is a non-clusterhead node with inter-cluster links, so it can access neighboring clusters and forward packets between them. A cluster member is usually called an ordinary node, which is a non-clusterhead node without any inter-cluster links[3].

Since the purposes of forming clusters are to stabilize the end-to-end communication paths and to improve the network scalability, the cluster stability must be considered, which is defined to be the lifetime of the clusters[6]. Unstable clusters could jeopardize both objectives. Many existing clustering algorithms do not consider the cluster stability as the design goal and, therefore, experience frequent cluster changes. This paper proposes a new clustering algorithm. The basis of the algorithm is a scheme that predicts the stability of each mobile host based on the associativity with its neighborhood and its remaining energy. The concept of associativity (relative stability of nodes) is used to get long-lived routes[7]. On the other hand high remaining energy of a node indicates that the node is stable in the energy sense. Choosing associativity and remaining energy in the selection of clusterheads increases the temporal stability of the clusters. In addition to these parameters, node's degree is also taken into consideration. Nodes with degrees greater than a predetermined value are given priority in the clusterhead election process to reduce the number of clusters throughout the network.

The rest of the paper is organized as follows. In Section 2, we summarize related work with a particular focus on a recent algorithm named Distributed Score Based Clustering Algorithm(DSBCA)[2]. In Section 3, we propose the new clustering algorithm. Simulation results are presented in Section 4 while conclusions are drawn in Section 5.

2 Related Work

Many clustering techniques have been proposed over the past years to manage MANETs effectively. Each technique has its own strengths and weaknesses. Some of the well-known algorithms are the Lowest-ID[4] and the Highest-Degree[5] Algorithms, the Weighted Clustering Algorithm(WCA)[10] and the Distributed Weighted Clustering Algorithm(DWCA)[13]. The Lowest-ID algorithm chooses the node with the minimum identifier in the neighborhood as the clusterhead. Since node ID does not change over time, lower ID nodes are prone to power drainage due to serving as clusterheads for longer periods of time resulting in shorter system life time. In the Highest-Degree algorithm the node with the highest degree in the neighborhood is elected as the clusterhead. It performs much worse than the Lowest-ID algorithm in terms of stability of clusters. The WCA algorithm combines system parameters such as degree, transmission power, mobility and battery power with certain weighing factors to calculate the weight of a node. The node with the smallest weight in its neighborhood is chosen as the clusterhead. The overhead induced by WCA is very high.

The DSBCA[2] is a more recent algorithm based on the idea of WCA. The algorithm considers the Battery Remaining, Number of Neighbors, Number of Members, and Stability in order to calculate a node's score with a linear function. After each node calculates its score independently, the neighbors of the node are notified about it. Each node selects one of its neighbors with the highest score to be its clusterhead. The algorithm was compared with WCA and DWCA in terms of number of clusters, number of re-affiliations and the lifespan of nodes in the system. Although simulation results showed that DSBCA achieved better performance than WCA and DWCA it has some inherent deficiencies. The following subsections describe the identified deficiencies of DSBCA.

2.1 Neighboring Time as the Metric of Node Stability

DSBCA defines the stability of a node to be the total time in which the neighbors of a specific node have spent their time beside the node. It is calculated as follows-

$$S_{(DSBCA)} = \sum_{i=1}^{n}(T_{RL} - T_{RF}) \tag{1}$$

Where n is the number of node's neighbor, T_{RL} is the time of the last packet reception from a neighbor and T_{RF} is the time of the first packet reception from the same neighbor. A higher stability simply means that the neighbors of a certain node has spent a longer time in its transmission range. But it can also be seen that a node with many new neighbors (high degree) might have the same stability value as a node with very few old neighbors. This is because the calculation of S takes into account the neighboring time of both stable and unstable neighbors of a node. Thus it becomes a poor representation of stability of a node in some situations.

2.2 Adjacent Cluster Heads

Two clusterheads might be in direct communication with each other in DSBCA. This is because in DSBCA, instead of clusterheads announcing their leadership, nodes send membership messages to their elected clusterheads. As a result clusterheads might become adjacent nodes. This will create inter-cluster interference, which is not desirable.

2.3 Problem with Weight Based Clustering Technique

In DSBCA two neighboring nodes may have the same weight. But it does not mention how to break such tie. As a result a node might elect more than one clusterhead. This increases the number of clusters in the network.

The proposed algorithm in this paper tries to eliminate the above drawbacks of DSBCA. Unlike DSBCA's general concept of weight, our algorithm takes the actual values of the system parameters such as stability, remaining energy and degree into account and quantitatively measures each node's suitability to become a clusterhead.

3 Associativity Based Energy Aware Clustering Algorithm

In this section we formally describe our new clustering algorithm, named Associativity Based Energy Aware Clustering Algorithm. The algorithm works in a distributed manner. As mentioned in the previous section it takes three attribute values of a node-stability, remaining energy and degree. The nodes in the network exchange these attribute values with their neighbors by means of periodic advertisements. Every node makes its own decision to become a clusterhead after having collected the attribute values from each of its neighbors and comparing them against it's own. The following subsections give a detailed description of how these three attribute values are measured.

3.1 Stability

In our algorithm the stability of mobile nodes is defined in terms of their neighborhood associativity. In the networking context, associativity means periods of spatial, temporal, connection and signal stability[7]. The associativity between nodes is measured in associativity ticks that are calculated from broadcasted packets. All nodes in the network periodically send out Advertisement packets to signify their existence. A node counts the number of Advertisement packets received from a neighbor by a counter variable called Associativity Tick(AT). Whenever it receives an Advertisement packet from the neighboring node, it increments the AT value corresponding to that neighbor. On the other hand, if it does not receive any Advertisement packets within a timeout period, it decrements the neighbor's corresponding AT value. When a neighbor's AT value reaches a desired threshold S_{th}, it is marked as *stable.*

As an example, assume there are two mobile nodes A and B each having a transmission and reception range of 10m in diameter[8]. Initially they are not in radio connectivity with each other but each sends an Advertisement packet every two seconds. If B is migrating at 1m/s speed and it starts to enter A's radio range and move through it, then both A and B record at most 5 Advertisements each. Hence this is the associativity threshold. If only 5 or less Advertisements are recorded, then one can assume that the other mobile host is migrating past it, and this situation is viewed as ***associatively unstable***. Otherwise, if the mobile host is moving but is constantly within the radio coverage of its neighbor, then more than 5 Advertisements will be recorded and hence the node will be noted as ***associatively stable***.

The associativity measure of a node is used as the primary basis for electing clusterheads in the proposed algorithm. It will serve as an indication of a node's stability. The associativity of any node 'x' in the network is defined to be the sum of associativity ticks of its *stable neighbors* only whose AT$\geq$$S_{th}$. Neighbors whose AT values are below the threshold S_{th} are not considered. Thus the sum of associativity ticks of a node 'x' is defined as

$$\sum AT(x) = \sum_{v=1}^{S(x)} AT(v) \tag{2}$$

Where S(x) = {v$\subset$N(x) | AT(v) $\geq$ S_{th}}, where N(x) is the neighborhood of x. A node that has the highest $\sum$AT among its neighbors signifies its highest stability and hence the highest priority in becoming the clusterhead. By selecting nodes with high associativity values, a cluster is expected to be long-lived.

3.2 Remaining Energy

The remaining energy of a node plays a subsidiary role in determining its stability. A node with greater remaining energy than its neighbors can serve for a longer period of time and hence can be considered to have greater stability in the energy sense. NS-2's[9]simple energy model is used to calculate the remaining energy of a node. This model states that, every time a packet is received(transmitted), the total energy is decreased by the value:

$$\text{DecEnergy} = P_{rcv}(P_{tx}) \text{ x rcvTime(txTime)}$$

Where $P_{rcv}(P_{tx})$ is the power consumed during receiving(transmitting) a packet and rcvTime(txTime) is the time to receive(transmit) the packet. The remaining energy of a node will be

$$\text{Remaining Energy} = \text{Energy - DecEnergy}$$

3.3 Degree

Degree of a node is defined to be the existing neighbors of the node that are within its transmission range. If two nodes have the same associativity and energy levels but varying degrees then the node with the higher degree is chosen to be the clusterhead. Higher degree nodes are preferred to reduce the number of clusters throughout the network.

Two nodes may have the same values in all three attributes. In such an unlikely case, the node with the lowest identifier is used to break the tie.

3.4 Algorithm Description

The proposed clustering algorithm executes in three different phases. They are

1. Neighborhood Discovery by Advertisement of Attributes
2. Clusterhead election
3. Cluster Formation and Finalization of Roles

The following subsections present these three phases in detail.

3.4.1 Neighborhood Discovery by Advertisement of Attributes

Every node periodically broadcasts Advertisement packets to its neighborhood. A node includes its up-to-date stability value ($\sum$AT), its remaining energy (E) and its degree (D) in the Advertisements. Every node maintains a neighborhood table to keep a record of the received Advertisements. For each neighbor, the table keeps the neighbor ID, its AT, its $\sum$AT, its D, its E, and the time of receiving the latest Advertisement from it. When an Advertisement is received, the corresponding entry in the table is updated and its AT value is incremented by 1. If a node does not receive an Advertisement from an existing neighbor within a time out period, it decrements the AT value of that neighbor. When the AT of a neighbor becomes zero, its entry is deleted from the neighborhood table.

3.4.2 Clusterhead Election

Initially, all the nodes in the network are free or in the UNDECIDED state. The nodes use their neighborhood tables to determine their respective roles. It takes at most three steps to select a clusterhead from a neighborhood table. These step are described below. It is important to note that all the operations of clusterhead election take into account only the stable neighbors of a node(those whose AT$\geq S_{th}$)

Step 1: A node first sorts its neighborhood table from the highest $\sum$AT to the lowest $\sum$AT, including itself. Since cluster stability is the main concern, a node will select the neighbors having the highest $\sum$ATs.

Step 2: The list of nodes from Step 1 is next sorted from the highest energy (E) to the lowest energy (E). Nodes having the highest energy (E) are selected for the next step.

Step 3: The node with the highest degree from step 2 becomes the clusterhead. In case of a tie, the node with the lowest ID becomes the clusterhead.

If a node selects itself as the clusterhead, it switches to the HEAD state and broadcasts a ClusterHead Beacon message; otherwise it remains in the UNDECIDED (free) state.

As an example, consider the neighborhood table of an arbitrary node n (Tab. 1) which is about to select its clusterhead. The associativity threshold (S_{th}) of the network is set to 6. The node has three neighbors 1,8 and 19, all of which are stable. It calculates its own $\sum$AT as follows,

$$\sum AT(n)=AT(1)+AT(8)+AT(19)=6+6+6=18$$

Table 1. Neighborhood Table of Node n

ID	AT	$\sum$AT	D	E	Last time heard from this node
1	6	6	1	90.0	T_1
19	6	18	3	95.75	T_{19}
8	6	18	3	100.0	T_8

Step 1: The node first sorts the records in its Neighborhood table by $\sum$AT, including itself. The result of the sorting is shown in Tab. 2

Table 2. Neighborhood Table of Node n (sorted by $\sum$AT)

ID	AT	$\sum$AT	D	E	Last time heard from this node
19	6	18	3	95.75	T_{19}
8	6	18	3	100.0	T_8
n	–	18	3	100.0	–
1	6	6	1	90.0	T_1

Step 2: Since all three records have the same $\sum$AT, they are next sorted by the field E(remaining energy). Nodes n and 8 have the same highest remaining energy. So selection will continue to step 3 which is based on node degree.

Step 3: Both nodes n and 8 have the same degree. So to break the tie, the lowest ID node will be selected to be the clusterhead.

During the clusterhead selection phase, no new nodes are allowed to join the cluster, even if they reach the required stability level. In that case the node is not marked as stable. This is to help the selection mechanism to be simple and fast.

3.4.3 Cluster Formation and Finalization of Roles

A node that wins the election, switches to the HEAD state and broadcasts a Clusterhead Beacon (CHB) packet to notify its leadership. A node in UNDECIDED state receiving a CHB packet from a neighboring clusterhead will become the member of the corresponding cluster. It will switch from the UNDECIDED state to the MEMBER state and send a Member Beacon (MB) message to the respective head. A node receiving multiple CHB packets from different neighboring heads will assume the role of a gateway and switch to the GATEWAY state. A clusterhead receiving a CHB packet will simply discard it. A cluster member(gateway) maintains a **Clusterhead Table** which keeps the ID(s) of its head(s). Likewise, a head node maintains a **Member Table** containing the IDs of all its members and gateways. Re-clustering only takes place when a head node loses contact with all its members or when a member(gateway) node loses contact with its head(s).

4 Simulation

To simulate the new clustering algorithm and compare its performance against DSBCA, Network Simulator version 2.31[9] was used. The simulations were carried out on an 800m X 800m area with N nodes as in DSBCA. The value of N was varied between 10 and 60. Radio range for each node was set to 250 meters. The nodes in the simulations moved according to the random way point

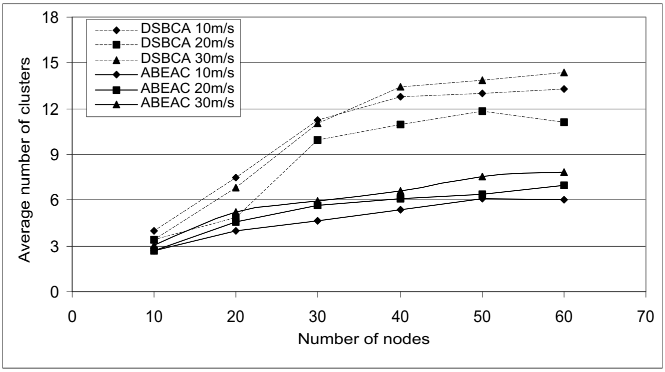

Fig. 1. Comparison of average number of clusters for varying speed and node density

model[11]. The nodes moved randomly in all directions with maximum speed of 10m/s, 20m/s,30m/s. AODV [12] was used as the underlying routing protocol.

4.1 Performance Metrics

Three metrics were chosen to compare the performance of our clustering algorithm with DSBCA: (i)the average number of clusters, (ii)the average number of reaffiliations and (iii)the average lifetime of clusters. Our algorithm showed better performance than DSBCA on all three metrics.

4.1.1 Average Number of Clusters

Figure 1 shows the comparison of the average number of clusters formed in DSBCA and our algorithm. It can be seen that the average number of clusters increase with node mobility in both algorithms. This is because as the mobility of nodes increase, their stability decreases. As a result more clusters are formed. But the important thing to note is that our algorithm forms less clusters in comparison to DSBCA regardless of this node speed, which is desirable.

4.1.2 Average Number of Reaffiliations

Figure 2 depicts the comparison of the average number of reaffiliations in the proposed algorithm with DSBCA for different speed values. According to the result, our algorithm gives less reaffiliations than DSBCA. If a node detaches itself from its current clusterhead and attaches to another clusterhead, then the involved clusterheads update their member list instead of invoking the election algorithm. Re-clustering takes place only when a head loses contact with all its members or when a member(gateway) loses contact with its head(s). A non-clusterhead never challenges the status of an existing clusterhead. Thus the clusterheads change as infrequently as possible resulting in longer duration of stability of the topology.

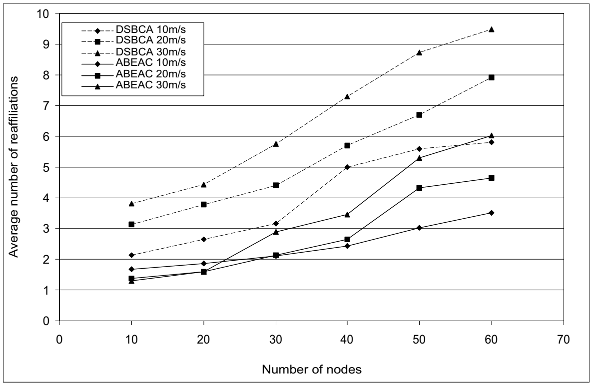

Fig. 2. Comparison of average number of re-affiliations for varying speed and node density

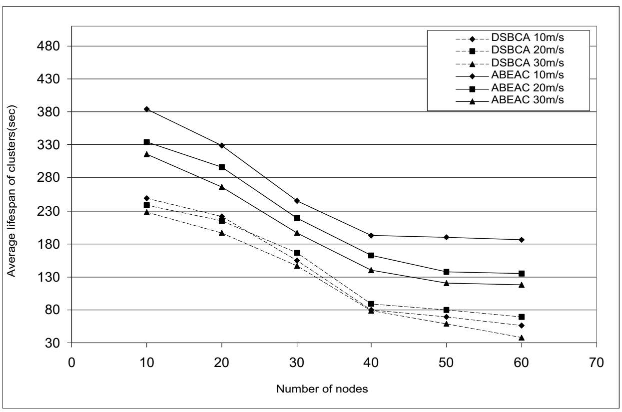

Fig. 3. Comparison of average lifespan of clusters for varying speed and node density

4.1.3 Average Lifespan of Clusters

Figure 3 shows a comparison of the average lifespan of clusters in DSBCA and our algorithm. As the mobility and node density increases, nodes consume more battery power[2]. Consequently the minimum lifespan of nodes decreases in both algorithms. The proposed algorithm provides longer lifespan of clusters. Since the stability of clusters in the proposed algorithm is high and the re-clustering frequency is less, nodes send and receive comparatively less messages than DSBCA. This leads to longer battery lifetime and thus longer lifespan of clusters.

5 Conclusion

In this paper we have proposed a new clustering algorithm and proved by simulations that our algorithm achieved its functional goals. The algorithm showed better performance in terms of smaller number of clusters, a longer lifespan of clusters and much less reaffiliations which is an indication of cluster stability. It is believed that the proposed clustering algorithm will be suited best for small to medium sized networks. In future, an associativity based hierarchical routing protocol can be implementing on top of this clustering technique. The routing protocol's main objective would obviously be to find stable enough links between the source and the destination.

References

1. Lin, C.R., Gerla, M.: Adaptive clustering for mobile wireless networks. IEEE J. on Selected Areas in Communications 15(7), 1265–1275 (1997)
2. Adabi, S., Jabbehdari, S., Rezaee, A., Adabi, S.: A Novel Distributed Clustering Algorithm for Mobile Ad Hoc Networks. J. Computer Science 4, 161–166 (2008)
3. Yu, J.Y., Chong, P.H.J.: A survey of clustering schemes for mobile ad hoc networks. IEEE Communications Surveys and Tutorials 7, 32–48 (2005)
4. Gerla, M., Tzu-Chieh Tsai, J.: Multicluster, mobile, multimedia radio network. Wireless networks 1(3), 255–265 (1995)
5. Parekh, A.K.: Selecting routers in ad-hoc wireless networks. In: Proceedings SBT/IEEE Intl. Telecommunications Symposium, pp. 420–424 (1994)
6. Wu, Y., Wang, W.: MEACA: Mobility and Energy Aware Clustering Algorithm for Constructing Stable MANETs. In: Military Communications Conference, MILCOM 2006, pp. 1–7 (2006)
7. Toh, C.-K.: Associativity-Based Routing For Ad-Hoc Mobile Networks. Kluwer Wireless Personal Communications 4(2), 103–139 (1997)
8. Tenhunen, J., Typpo, V., Jurvansuu, M.: Stability-based multi-hop clustering protocol. In: 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications, PIMRC 2005 (2005)
9. McCanne, S., Floyd, S.: NS network simulator (1995)
10. Chatterjee, M., Das, S.K., Turgut, D.: WCA: A weighted clustering algorithm for mobile ad hoc networks. Cluster Computing 5(2), 193–204 (2002)
11. Johnson, D.B.: Routing in Ad Hoc Networks of Mobile Hosts. In: Workshop on Mobile Computing and Applications (1997)
12. Perkins, C.E., Royer, E.M.: Ad-Hoc on-Demand Distance Vector Routing. In: Proc. of IEEE WMCSA 1999, New Orleans, LA (February 1999)
13. Choi, W., Woo, M.: A Distributed Weighted Clustering Algorithm for Mobile Adhoc Networks. In: AICT/ICIW (2006)