

A Proposed Model for Load Balancing in MANET

Supreet Kaur
Deptt of CSE,
NSIT, Dwarka, India

Varsha Kumari
Deptt of CSE,
NIT, Delhi, India

ABSTRACT

MANET is a multi-cluster, multi-hop packet radio network that should adapt itself to the dynamically changing network configurations. This dynamic reconfiguration is governed by clusterheads, a major set of nodes that have enough resources to be capable of routing packets in the network. This paper proposes a load balancing heuristic to extend the network lifetime. It has two phases. Initial clustering is done for the entire network in start up phase that takes node's degree and energy as input parameters. The maintenance phase keeps the network topology updated through global re-clustering and local updation. Global reclustering is initiated when the network becomes significantly unbalanced i.e. if the variance of degree of the cluster heads in the network is greater than a pre-determined threshold. It is expected to get better result than the time driven algorithms, as implementation is still under progress.

Keywords

Ad-Hoc networks, Clustering, Load Balancing, Reclustering, Energy Conservation

1. INTRODUCTION

A wireless ad-hoc network consists of a set of self-organizing mobile nodes which required no fixed infrastructure and which communicate with each other over wireless links. Properties of mobile ad hoc networks (MANET) like dynamic topology and decentralized connectivity make network management a challenging task.

A clustering is a well known technique for organizing large networks by grouping nodes that are close to each other in the network. Clustering has advantages such as improving bandwidth utilization by reducing communication overheads and reducing energy consumption.

In a clustering structure, each mobile node may be assigned a different status such as cluster head, cluster member, cluster gateway or undecided. Cluster heads are nodes that are vested with the responsibility of routing messages for all the nodes with their cluster. Limiting or optimizing the number of nodes in each cluster can achieve a high throughput of the system.

This paper proposes a load balancing algorithm to insure a fair distribution of load among clusterheads. The proposed algorithm consists of two distinct phases. The start-up phase performs initial partitioning of a network into clusters, while the maintenance phase updates the topology of a network as time goes on. Depending on the situation, algorithm performs either local updation or global reclustering.

The load balancing is accomplished by maintaining a pre-defined threshold on the number of nodes that a clusterhead can cover ideally.

By balancing the load of clusterheads, the average execution time can be minimized and the lifetime of the overloaded (clusterhead) node can be maximized. For small networks, using proactive protocols such as DSR, its possible for each

node to keep the updated information of nodes in the network but for moderate or large network, when reactive protocols are used, it becomes practically impossible for nodes to keep the updated information about other nodes in the network.

Global reclustering is initiated when the network becomes significantly unbalanced. Based on the value of variance, degree of imbalance can be identified and re-clustering can be initiated if required.

The rest of this paper is organized as follows. A brief review of related work is discussed in Section 2. In Section 3, the proposed load balancing algorithm is discussed. The result is given in Section 4. In Section 5, the conclusion is provided.

2. RELATED WORK

Many strategies have been proposed so far to address ad-hoc networks clustering problem.

The Weighted Clustering Algorithm (WCA)[1] employs combined-metrics based clustering : node degree, and mobility or moving speed, are taken into account to calculate a weight factor. Mobile nodes with local minimum weight are elected as clusterheads. WCA does not invoke re-clustering when a member node changes its attaching cluster.

Overlapping clusters algorithm (OCA) proposed in [2] consists of two distinct phases. The start-up phase performs initial partitioning of a network into clusters. In the maintenance phase, the status of the network is periodically monitored and network topology is updated through local and global re-clustering. OCA may be the preferred algorithm in applications that require higher reliability but do not focus much on efficient cluster formation.

A cluster head, apart from supporting its members with the radio sources, has also to route messages for other nodes belonging to different clusters. Therefore, it is not desirable to have any cluster head overly loaded while other are lightly loaded. At the same time, it is hard to maintain a perfectly load balanced system at all times due to frequent detachment and attachment of the nodes from and to the cluster heads.

In [3], it tries to find the best node to share the load but does not consider communication overhead and energy consumption.

Hussein, Yousef and Arabiyat [4] proposed the enhancement on weighted clustering algorithm (EWCA) that leads to a high degree of stability in the network and improves the load balancing. When a cluster size exceeds its threshold/predefined limit, local election procedure is repeated to adjust the number of mobile nodes in that cluster.

A dynamic energy efficient clustering algorithm (DEECA) presented in [5] selects the clusterheads based on energy and mobility and locally alters the topology to increase the network lifetime and reducing their energy consumption while periodically monitoring their energy.

In [7], a load balancing approach based on clustering is proposed where the total execution time of the tasks is minimized by distributed the workload among clusterheads.

Clusterhead Load-Balancing Technique (CLBT) [8] distributes the load between clusterheads. Unloaded clusterhead increases its transmission range while loaded clusterhead decreases its range. All clusterheads share the load and distribute their nodes in a fair manner.

[9] and [10] discusses different load balancing approaches for MANETs.

As an alternative, a new event-driven load balancing algorithm is proposed. The main advantage is that it will increase the network lifetime by invoking re-clustering whenever the network load imbalances.

3. PROPOSED ALGORITHM

Even though the existing clustering algorithms can efficiently divide a network into clusters, only a few of them focus on improving network lifetime and load balancing. Due to this imbalance, some nodes can expire prematurely. The proposed algorithm allows for electing a node to play the role of cluster head taking into consideration its degree and energy. It establishes the basis for an energy efficient environment by allowing each node to monitor its energy expenditure and to strive its utmost during the network lifetime.

3.1 Algorithm preliminaries

Each cluster has exactly one cluster head which is one hop away from all its cluster members. Each node in the network is assigned a status indicator: UNDECIDED(UD), GATEWAY (GW) , CLUSTER MEMBER(CM), or CLUSTER HEAD (CH).

During the cluster formation, nodes that are within the transmission range of two cluster heads are called gateways and are responsible for communication between adjacent clusters. Initially, all nodes are assigned a 'UNDECIDED' status which means that a node is still in search of its cluster. Clusterhead is selected on the basis of node's degree and its remaining energy where, **Degree(d)** of a node is defined as the number of neighbors of a node, and **Energy(b)** is defined as the energy available with the node.

Due to mobility, a node's neighborhood changes with time. Changes in the network topology over time are arbitrary as the mobility of nodes may not be predictable. However, nodes may not be aware of changes in the neighborhood. Therefore, clusters and cluster heads must be updated frequently to maintain accurate network topology. For this, all nodes broadcast their node identity periodically. Data structures have to be maintained for each node in the cluster.

3.2 Clusterhead election procedure

The following is a detailed description of algorithm.

3.3.1 Start-up phase

A node is elected as a cluster head taking into account its degree and energy left. Heterogeneous network is considered so nodes can have different energy available with them. Each node is assigned a weight that indicates its suitability for the clusterhead role. The weighing factors w_1 and w_2 are kept fixed for a given system. The weighing factors also give the flexibility of adjusting the effective contribution of each of the parameters in calculating the combined weight, W .

$$W = (w_1 * (1 - (d_d / \text{MAX_DEGREE}))) + (w_2 * b / \text{MAX_BATTERY}) \quad \text{---- (1)}$$

$$\text{where } w_1 = \alpha \\ w_2 = 1 - \alpha \quad 0 \leq \alpha \leq 1$$

Note that, with this computation, weight value fall in the range of [0..1] because the parameter values and the parameter weight are also normalized.

$$d_d = \text{abs}(d_a - d_i)$$

MAX_DEGREE is the maximum degree among all nodes in its neighborhood and itself.

MAX_BATTERY is the maximum battery among all nodes in its neighborhood and itself.

Degree difference (d_d) is a difference of actual degree of the node, d_a and its ideal degree, d_i .

Node declares itself as a Cluster head if

- (i) Its weight is maximum among all its neighbors.
- (ii) Its weight is same as maximum weight of its neighbors then the conflict is resolved by selecting node with lower ID.

All CH's thus selected informs its cluster members. A node having more than one CH as its neighbor becomes a GW. If a battery of a node is less than cluster head threshold then it will not participate in cluster election process.

If a node is in UD state,

- (i) If it has no neighbor i.e. the node is isolated node, it remains in UD state. It changes its status when it comes in contact with other nodes.
- (ii) If it has neighbors and at least one of them is in UD state, it initiates CH election procedure among those nodes.
- (iii) If it has neighbors but they are all CM's or GW's, it declares itself as a CH.

Once all of the nodes have decided on their status, the network is completely divided into clusters. At this point, the proposed load-balancing algorithm begins its maintenance phase.

3.3.2 Maintenance phase

When network topology is highly dynamic, resulting in frequent cluster topology updates, the control overhead of cluster maintenance increases drastically. Limiting re-clustering situations or minimizing explicit control messages for clustering can maintain the cluster structure maintained well without excessive consumption of network resources [6].

Cluster maintenance algorithm follows an approach whereby clustering is not executed periodically but in an event driven manner. That is, re-clustering process is only invoked when there is significant load imbalance among CHs in the network.

If a cluster head battery is less than minimum threshold then that cluster is dissolved and a local re-clustering takes place. When the network becomes significantly unbalanced i.e. the variance of degree difference of all the nodes is greater than pre-determined threshold, global re-clustering is triggered.

3.3.2.1 Cluster Updation

In this phase, each moving node periodically sends *HELLO* message and updates the neighbor information. The proposed algorithm resolves the contention of two cluster heads that move into the transmission range of each other. The cluster head with the largest weight or lowest ID, if the weights are equal, will keep its role as a cluster head, while the other one will change its status to CM. The node whose CH list size now become zero, they will change their status to UD and will elect their new CH among themselves.

3.3.2.2 Global re-clustering

If the difference between the heaviest loaded and the lightest loaded nodes is minimized, the average work execution time can be reduced, the energy of the nodes will be better exploited and the nodes lifetime can be extended. Therefore, any node in the network whose degree is greater than MAX_DEGREE or less than MIN_DEGREE, it calculates the variance of the degree of all CHs.

If variance is greater than pre-determined threshold value, clustering is initiated in the whole network.

4. RESULTS

The proposed load balancing algorithm will be compared with DEECA[5] through simulations. The algorithm is implemented using GloMoSim 2.03.

To measure the performance of the system, the algorithm focused on *network lifetime*.

Since In DEECA, a cluster head compares its load with the average of its adjacent cluster heads, and when its load is more than the average, it asks its members to affiliate with other neighbouring CH's. Average of nodes can be misleading. Few of the neighbouring nodes may be highly loaded and few of them may be lightly loaded. Whereas in proposed algorithm, standard deviation (variance) of load is considered.

Also, in DEECA, nodes selected as CH will continue as CH's till they reach red threshold. While in the proposed algorithm, during re-clustering phase, each node is given an equal opportunity to be elected as CH.

Hence, it is expected to give better results in terms of network lifetime and packet delivery ratio than DEECA.

The proposed algorithm is under implementation and actual results will be compared as early as possible.

5. CONCLUSION

In this paper, an event-driven load balancing algorithm is proposed, the primary objective of which is to increase the network lifetime by invoking re-clustering whenever the network load imbalances. It is expected to perform better than DEECA.

6. REFERENCES

- [1] Mainak Chatterjee, Sajal K. Das, Damla Turgut. "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks", 2002.
- [2] Nevin Aydin, Farid Na'it-Abdesselam, Volodymyr Pryyma and Damla Turgut, "Overlapping Clusters Algorithm in Ad Hoc Networks", 2010.
- [3] Turgut, D.K. Das, S.K.Elmasri, R., "Balancing loads in Ad Hoc Networks", 2003.
- [4] Abdel Rahman H. Hussein, Sufian Yousef, and Omar Arabiyat, "A Load-Balancing and Weighted Clustering Algorithm in Mobile Ad-Hoc Network".
- [5] Haidar Safa, Omar Mirza & Hassan Artail, "A Dynamic Energy Efficient Clustering Algorithm for MANETs".
- [6] J. Yu, P. Chong, "A survey of Clustering Schemes for mobile ad-hoc networks", IEEE Communication Surveys, 7(1), pp. 32-48, March 2005.
- [7] Rachida Aoudjit, Mustapha Lalam, Abdelaziz M'zoughi, Malika Belkadi and Mehammed Daoui, "Load Balancing : An Approach Based on Clustering in Ad Hoc Networks", Journal of Computing and Information Technology – CIT 17, 2009
- [8] Labdah AL Ghafran, Zulkefli Bin Muhammed Yusof, "Load-Balancing Technique in Clustered Mobile Ad-Hoc Networks", 2013 International Conference on Advanced Computer Science Applications and Technologies.
- [9] Shashank Bharadwaj, Vipin Kumar, Ankit Verma, "A Review of Load Balanced Routing Protocols in Mobile Adhoc Networks", International Journal of Engineering Trends and Technology – July to Aug Issue 2011.
- [10] D.Maheshwari, R.Nedunchezian, "Load Balancing in Mobile Ad Hoc Networks : A Survey", International Journal of Computer Applications, December 2012.