International Journal of Enhanced Research in Management & Computer Applications
ISSN: 2319-7471, Vol. 4 Issue 7, July-2015

Investigate Clustering in Mobile Ad Hoc Networks

Abolfazl Akbari

Department of Computer Engineering, Ayatollah Amoli Branch, Islamic Azad University, Amol, Iran

ABSTRACT

Clustering of nodes provides an efficient means of establishing a hierarchical structure in mobile ad hoc networks. In mobile adhoc networks, the movement of the network nodes may quickly change the topology resulting in the increase of the overhead message in topology maintenance; the clustering schemes for mobile ad hoc networks therefore aim at handling topology maintenance, managing node movement or reducing overhead. This paper presents the reasons for clustering algorithms in ad hoc networks, as well as a short survey of the basic ideas and priorities of existing clustering algorithms.

Keywords: clusteiheed; member cluster; clustring; mobile ad hoc networks.

INTRODUCTION

MOBILE AD HOC networks (MANETs) provide an efficient method for a dispersed set of nodes to establish communication without the need for an infrastructure. Although dynamic routing can be established for this purpose, a simple flat structure based on reactive or proactive routing protocols may prove to be quite inefficient in large-scale MANETs with a high number of mobile nodes due to the higher link and processing overheads [1], [2]. Clustering in MANETs provides an effective method for establishing a hierarchical structure for this purpose. Many clustering schemes have been proposed for MANETs [3], where they each aim to meet certain needs of the system. This could provide a system having low clustering related maintenance costs, or energy efficient clusters to minimize energy consumption suitable for mobile nodes with energy constraints, or for load balancing to distribute the workload of a network. Fig. 1 illustrates the concept of clusters. In such a model, there are usually two main types of nodes, i.e., the cluster head which is in charge of the cluster, and cluster members, which join a cluster and are controlled by the cluster head. In this paper, we consider single-hop (one-hop) clusters. All the member nodes in such a cluster are within the range of the cluster head but not necessarily within range of each other. In the single-hop cluster, any member node is at most within two hops away from any other member node via the clusterhead. This defines the cluster's diameter. The clusterhead is in charge of cluster maintenance, such as resource allocation to members and the acceptance of members into the cluster. Member nodes can join a cluster if the clusterhead accepts their join request.

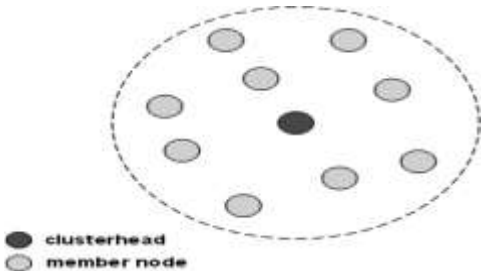

Fig. 1. Concept of clustering in MANETs.

An efficient clustering must elect suitable clusterheads to achieve the clustering scheme's main objectives, and the clusterheads must also accept suitable nodes to become members of their clusters.

RELATED WORK

[4]Recent work in clustering for wireless networks began with the work of Gerla and Tzu-Chieh Tsai [5]. In the algorithms, all nodes start out as clusterheads. In the first version of the algorithm, if a node hears from a clusterhead with a lower ID than itself, it resigns and uses that node as a clusterhead instead. The other version is exactly the same,

except that the degree of the nodes (the number of neighbors each node has) is used instead of ID. Since the degree of a node changes with movements in the network, the clusterheads are not likely to stay clusterheads for very long. On the other hand, using the Lowest-ID algorithm, the nodes with a low ID stay clusterheads most of the time. This is an unfair distribution, that could lead to some nodes losing power prematurely. Alan D. Amis and Ravi Prakash [6] present additions to these clustering mechanisms that helps avoid cluster head exhaustion by providing ”virtual IDs” to the nodes. When all clusters have a radius of size one, all nodes are directly connected to their clusterhead. The advantage with this approach is that the nodes that are neither clusterheads nor gateways can sleep, and fetch messages stored in the clusterhead at any time, since they have a direct connection to the clusterhead.

The disadvantage is that the clusters are limited in size. An algorithm that makes it possible to choose a radius larger than 1 is presented in [7]. In [8], the priority is to create clusters of size between k and 2k−1. The distributed algorithm first creates a rooted spanning tree covering the entire network. The cluster formation is run bottom-up, where subtrees are made into clusters that fit the size requirements. There is no real limit on the radius of the clusters, in theory the diameter is O(k). 1 Unlimited radius can result in problems depending on the application. But if the highest priority is constant cluster sizes, this algorithm is the only one that guarantees the property. In [9], another algorithm presents a cluster structure that is, with high probability, a constant approximation of the optimal solution. In this case, optimal cluster structure is the one that uses the lowest number of clusters to cover the network at this time. A. Bruce McDonald and Taieb Znati present an algorithm that forms clusters of nodes that have sufficient probability to stay connected during a specific time interval [10].

SURVEY OF NEW ALGORITHMS

We Survey of two new clustering algorithms for pseudolinear highly mobile ad hoc networks.

A. Distributed group mobility adaptive (DGMA)

Based on the work by Bai et al. [11], extend and develop the concept of the mobility metric, spatial dependency (SD). SD captures the similarity of the velocities between two nodes that are within their communication range. To use this metric to extract the characteristics of group mobility in MANETs. also have observed that nodes belonging to the same group are more likely to move within the same region over a period of time to complete their tasks. When nodes are moving within a stationary region, the group is considered to have no movement[12]:. For instance, a group of rescuers may occasionally concentrate their searching in a specific area, or a group of people stays together during a networking session in a conference or to engage in a discussion while visiting an exhibition. All of these scenarios reflect the group mobility characteristics with possibly stationary group while nodes are moving constantly. While moving forward toward a destination, a node may move around within a region frequently in an uninterrupted trajectory, say to carry out a search operation. As a result, only minor change has been made to the linear distance from a node’s starting point. Such oscillating movement of individual’s featured in group mobility does not contribute much to the group displacement and the vector summation of individual instant velocity may not reflect the group motion either. Based on above analysis, the use of the linear distance D as measurement of node displacement. With this term, also can identify points where “significant” changes in location have been made by a node. A node is moving along with a group which is in motion only if a node has changed its physical location by more than a threshold Dthr. Thus, each node has two velocity properties: the instant velocity and the velocity calculated based on D over time ΔT. Let Δx_T and Δy_T be the increment of the linear distance in x and y coordinates since the last significant movement was recorded, and $T = T_i$ be the moment when the last history information is recorded for node i. To calculate the linear displacement of a node, have:

$$\Delta x_{Ti} = (x_{ti} - x_{Ti}) \qquad (1)$$

$$\Delta y_{Ti} = (y_{ti} - y_{Ti}) \qquad (2)$$

where t is the current time and xt_i, yt_i, xT_i, yT_i are the coordinates of the node i at time t and T respectively. Thus, the linear distance D can be calculated by:

$$D = \sqrt{\Delta x_{Ti}^2 + \Delta y_{Ti}^2} \qquad (3)$$

If D $\geq$ Dthr, the most recent speed and direction of a node that will be entered to the history record are:

$$\tilde{S}_i = \frac{D}{t - \tilde{T}_i} \qquad (4)$$

$$\tilde{\theta}_i = \begin{cases} \varphi \cdot \mathrm{sgn}(\Delta y_n) & \Delta x_n > 0 \\ \pi / 2 \cdot \mathrm{sgn}(\Delta y_n) & \Delta x_n = 0 \\ (\pi - \varphi) \cdot \mathrm{sgn}(\Delta y_n) & \Delta x_n < 0 \end{cases} \qquad (5)$$

Where

$$\tan\varphi = \left|\frac{\Delta y_n}{\Delta x_n}\right| \text{ and } \tilde{\theta}_i \in (-\pi, \pi)$$

The time T_i , the speed S_i the direction i $\tilde{\theta}_i$, and the present location of the node will be recorded in the history cache for node i with

$$x_{Ti} = x_{ti} \qquad (6)$$

$$y_{Ti} = y_{ti} \qquad (7)$$

$$\tilde{T}_i = t \qquad (8)$$

Hence, at time t, a node's location is compared with the most recent record in the history cache such that a node's "micro" movement trajectory within the diameter of Dthr will not trigger any updating of the movement. Based on above, each node will periodically check its own mobility information at every time interval T_0. If its linear distance D is greater than Dthr, it updates its mobility history information correspondingly. Based on the information in the history cache, a node calculates its total spatial dependency (TSD) with the following steps:

Step 1: It exchanges its history information speed S_i and direction $\tilde{\theta}_i$ with its directly connected neighbors.

Step 2: A node calculates its relative direction (RD) and speed ratio (SR) with each of its directly connected neighbors. For example, for nodes i and j, the RD of these two nodes is the cosine of the angle between i and j at time t,

$$RD(i, j, t) = \cos(\tilde{\theta}_i(t) - \tilde{\theta}_j(t)) \qquad (9)$$

and the SR between two nodes i and j is given by

$$SR(i, j, t) = \frac{\min(\tilde{S}_i(t), \tilde{S}_j(t))}{\max(\tilde{S}_i(t), \tilde{S}_j(t))} \qquad (10)$$

Step 3: Calculate the Spatial Dependency,

$$SD(i, j, t) = RD(i, j, t) * SR(i, j, t) \qquad (11)$$

Thus, if a mobile node moves away from its neighbor, it will have a small SD between these two nodes. That means their mobility is independent from each other. On the other hand, if a node's movement is closely related to another node or is influenced by the nodes in its neighborhood, such that they move in similar direction and speeds, then they are expected to have a higher value for SD.

Step 4: Assuming the number of its directly connected neighbors is n, after exchanging the SD with each neighboring node, the node takes the summation of all SD it has and gets the Total Spatial Dependency (TSD) by

$$TSD\ (i, t) = \sum_{j=1}^{n} SD\ (i, j, t) \qquad (12)$$

A higher TSD value implies that node i has a larger neighbor set and it has a similar mobility pattern with its neighbors. The speed and direction may be strongly related to others. Thus, a node with a higher TSD value is entitled to represent and to reflect the mobility pattern of the group.

[12]In this section describe DMGA clustering scheme for group mobility in MANETs. Our design objectives are as follows:

1. Prolonged lifetime of cluster heads,
2. Applicable in highly mobile environment, and
3. Reduce the number of re-clustering.

The revised SD is used as the mobility metric in DGMA to avoid capturing the instantaneous changes of speed and direction. This makes DGMA scheme more adaptive to the highly mobile environment which is proved by the simulation result as shown in section IV. Coloring method is introduced in DGMA that a varieties of colors represent

nodes in different roles. So each node has a color property which would be any one of white, yellow and red. White indicates the initial color of each node or implies the node has not affiliated with any cluster. CHs are colored with red and cluster members are in yellow. first introduce the terminology used in DGMA.

1) T : the duration that two red nodes are directly connected.
2) n_1 : used by a red node to show the ratio of the number of one-hop white neighbors to the number of all onehop neighbors.
- n_{1_Input} is the input threshold value of n_1
3) n_2 : used by a white node to show the ratio of the number of one-hop white neighbors to the number of all onehop neighbors.
- n_{2_Input} is the input threshold value of n_2

DGMA clustering has two main processes: the nomination process and the cluster maintenance process. Each node maintains its own status and no centralized entity is responsible for the maintenance of a whole cluster. DGMA begins with the nomination process that has five steps:

Step 1: Each node begins in white.

Step 2: A white node, if not connecting with any red node, broadcasts a request to its directly connected neighbors to nominate a red node.

Step 3: After feedback of individual's TSD from its neighbors, the applicant (node A) compares its TSD with the received TSD values from each of its white neighbors (node B)
a. If TSD(A, t)< TSD(B, t), nomination process is terminated and it remains as a white node; b. If TSD(A, t)> TSD(B, t), continue until node A has compares with all its neighbors. **Step 4**: If the white node has the largest TSD among all of its white neighbors, it turns to red. Step 5: It broadcasts an invitation to its neighbors and the receivers in white change their color to yellow.

```
PROCEDURE Nomination (μ);
BEGIN
  Precondition:
      Input node: μ
      Neighbor set of μ: Θ
  Effect:
      if μ.color is white
          for each v ∈ Θ
              if ∀ v that v.color is white, μ.TSD > v.TSD is always true
                  μ.color = red
          if μ.color is red
              for each v ∈ Θ
                  if v.color is white
                      v.color = yellow
END
```

Fig.2 Protocol specification of nomination process

For the maintenance process at a node, it is triggered upon detecting the updates of mobility information. Different maintenance steps will be executed according to a node's color. For red nodes (i.e. CH_s): 1) If two red nodes are connected over a time T, the one with smaller TSD turns into yellow;
2) If more than n1-Input of neighbors are white, i.e. $n_1 \geq n_{1\text{-}Input}$, it turns into white.
For yellow nodes(i.e. member nodes):

1) If a yellow node has a bigger TSD than any of connected red nodes, it turns itself into white; or

2) If it fails to find any neighboring red node, it turns into white. For white nodes (i.e. free nodes):

1) If there is a red neighbor that has a larger TSD value than this white node's, it turns to yellow;

2) Else, if the component of white nodes in the neighbor set is greater than n2-Input, i.e. $n_2 \geq n_{2\text{-}Input}$, it triggers the cluster Nomination process.

```
PROCEDURE Maintenance(μ);
BEGIN
  Precondition:
      Input node: μ
      Neighbor set of μ: Θ
  Effect:
      if μ.color is red
          for each v ∈ Θ
              if ∃ v.color is red & connecting_time(μ, v) > T
                  if μ.TSD < v.TSD
                      μ.color = yellow
                  else
                      v.color = yellow
              else if n_1 ≥ n_1-Input
                  μ.color = white
      if μ.color is yellow
          for each v ∈ Θ
              if that ∃ v.color is red is false
                  μ.color = white
              else if ∀ v that v.color is red,
              μ.TSD > v.TSD is always true
                  μ.color = white
      if μ.color is white
          if ∃ v.color is red & μ.TSD < v.TSD
              v.color = yellow
          else if n_2 ≥ n_2-Input
            Nomination(μ);
END
```

Fig. 3 Protocol specification of maintenance process

From Fig.2 and Fig.3, DGMA has a worst case time complexity of O(n) per node, where n is the number of nodes in the network. The proof of the complexity and correctness of the algorithms are omitted due to page limitation.

B. DISTRIBUTED SCORE BASED CLUSTERING ALGORITHM (DSBCA)

[13] Clustering set up: In Distributed Score Based Clustering Algorithm (DSBCA) various important parameters for cluster heads selection is considered. These parameters are Battery Remaining, Number of Neighbors, Number of Members and Stability of node. Each node calculates its score by using a formula that considers all the above parameters. The score of node N is defined as Equation (1).

Score= $((B_r \times C_1) + (N_n \times C_2) + (S \times C_3) + (N_m \times C_4))$ (1)
Where C_1, C_2, C_3, C_4 are the score factors for the corresponding system parameters listed below:

- Battery Remaining (Br): The residual node's energy. The energy is consumed through transmitting/receiving data packets and messages
- Number of Neighbors (N_n): The number of existing nodes which are located in the transmission range.
- Number of Members (N_m): Set of nodes that is handled by each cluster head and is calculated from Equation (2).

Nm= (Mem_V × previous score) (2)
Where Mem_V is the member value and calculated as Equation (3).

$$\begin{cases} Mem_V = (\frac{N_m}{4}) & NM < 4 \\ Mem_V = (\frac{N_m - \delta}{\delta - 4}) & NM \geq 4 \end{cases} \quad (3)$$

Where δ is a pre-defined threshold for the number of nodes that a cluster head can handle ideally.

- Stability(S):

The total time in which the neighbors of a specific node have spent their time beside the node. A Higher stability simply means that the neighbors of a certain node has spent a longer time in its transmission range, conclude that the

mentioned node has a more stable situation. The stability is used to address movements and adjacencies of nodes and is calculated by Equation (4).

$$S = \sum_{i=1}^{n} T_{RF} - T_{RL} \qquad (4)$$

{n is the number of node's neighbors}

Where T_{RF} is the time of_the first packet reception and T_{RL}_is the time of the last packet_reception. Main structure of DSBCA is a five- step protocol:

Step 1. (Updating Neighborhood table) .Each node updates its Neighborhood table as soon as it receives Score-value messages. This process would be held until the node's timer is elapsed. Obviously if Score-value message is not received from a node, which is previously located in Neighborhood table, in specific period of time the node will be dropped from the table.

Step 2. (Calculating Score value). Each node calculates its score according to Equation (1) and broadcasts it by sending a Score-value (my-id, Score)
message.

Step 3. (Selecting cluster head). Each node checks its Neighborhood table to choose the node with the highest score to be its cluster head; furthermore the node announces its leadership to all neighbors through broadcasting a My-Ch (my-id, my-ch-id, init time) message.

Step 4. (Broadcasting messages). Each node broadcasts its type to all_neighbors_through_a_Type (myid, is-border, is-ch) message. Three types of nodes are defined in DSBCA:

- Cluster head (Zone leader node): A node that receives My-Ch message from at least one of its neighbors which its my-ch-id field is equal to node's id.
- Border node: A node which is associated with two following conditions and finds no other nodes with a higher score in the neighborhood region.

1. A node which receives messages from more than one cluster head which are located in its_ neighborhood region.
2. A node which receives messages from those nodes which are in its neighborhood region with different cluster heads

- Ordinary node: Members of existing clusters.

Step 5. (Calculating new timer value). First TV (Indicates the timer value) is set to desire acceptable value ,then calculated as illustrated in Equation (5,6,7).
a) If the node is cluster head then:
TV = ((step duration × b×D) + i) (5)
Where D is the local density calculated as N_{m+1}, N_m is the Number of Members, i and b are the normalization factors.
b) If the node is border then:
TV= Random (1600, current_time of cluster head's timer)(6)
c) If the node is ordinary then:
TV= Random (200, initial time of cluster head'stimer)(7).

CONCLUSION

With the development of MANETs, large-scale MANETs will become one of the future research directions. The routing efficiency and network complexity would be greatly influenced by the number of mobile hosts. Clustering can provide large-scale MANETS with a hierarchical network structure to facilitate network operations. In this paper, we survey of a distributed clustering scheme for group mobility in

MANETs as well as a novel group mobility metric. Based on the new mobility metric, DGMA ensures cluster stability with longer cluster lifetime, longer mean residence time for cluster members, and a smaller number of reaffiliation events. The results reflect that DGMA scheme outperforms the Lowest-ID algorithm in many aspects. With the DGMA clustering scheme, group partition can possibly be predicted by cluster heads based on their mobility information. Efficient routing protocols for mobile ad hoc groups can also be developed based on this hierarchical clustering structure and also In this paper survey of a novel clustering algorithm for MANETs. In DSBCA each node calculated its score by linear algorithm which is based on four important parameters: battery remaining, number of neighbors, number of members and stability. Each node independently chose one of its neighbors with the highest score to be its cluster head and thus the cluster head selection was performed in a distributed manner with most recently gathered information of current state of the neighbors.

REFERENCES

[1] P. Gupta and P. Kumar, "The capacity of wireless networks," IEEE Trans. Inf. Theory, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[2] X. Hong, K. Xu, and M. Gerla, "Scalable routing protocols for mobile ad hoc networks," IEEE Netw., vol. 16, no. 4, pp. 11–21, Jul./Aug. 2002.

[3] J. Y. Yu and P. H. J. Chong, "A survey of clustering schemes for mobile ad hoc networks," Commun. Surveys Tuts., vol. 7, no. 1, pp. 32–48, 2005.

[4] Ehssan Sakhaee, Student Member, IEEE, and Abbas Jamalipour, Fellow, IEEE, Stable Clustering and Communications in Pseudolinear Highly Mobile Ad Hoc Networks, of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 57, NO. 6, NOVEMBER 2008

[5] Mario Gerla, Jack Tzu-Chieh Tsai. Multicluster, mobile, multimedia radio network. ACM Baltzer Journal of Wireless Networks 1(3):255—265, October 1995

[6] A.Amis, R.Prakash. Load-Balancing Clusters in Wireless Ad Hoc Networks. In Proceedings 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology, pages 25—32 Mar. 24—25 2000

[7] Amis, A.D, Prakash, R, Vuong, T.H.P., Huynh, D.T. Max-Min D-Cluster Formation in Wireless Ad Hoc Networks Proceedings of IEEE March 2000, pages 32—41

[8] S.Banerjee, S.Khuller. A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks. Technical Report CS-TR-4103, University of Maryland, College Park, February 2000.

[9] J.Gao, L.J.Guibas, J.Hershberger, L.Zhang, A.Zhu. Discrete Mobile Centers. In Proc. 17th annual symposium on Computational geometry (SCG), pages 188—196 ACM Press, 2001.

[10] A.B.McDonald, T.Znati. A Mobility Based Framework for Adaptive Clustering in Wireless Ad-Hoc Networks. IEEE Journal in Selected Areas in Communications (JSAC), 17 (8):1466—1487, August 1999.

[11] J. M. Ng and Y. Zhang, "A Mobility Model with Group Partitioning for Wireless Ad hoc Networks," Proc. of IEEE International Conference on Information Technology and Applications, ICITA 2005, Sydney, July 2005, pp.289-294.

[12] Yan Zhang, Jim Mee Ng, A Distributed Group Mobility Adaptive Clustering Algorithm for Mobile Ad Hoc Networks, of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings,

[13] Sahar Adabi, Sam Jabbehdari, Amirmasoud Rahmani, Sepideh Adabi A Novel Distributed Clustering Algorithm for Mobile Ad-hoc Networks. of IEEE Journal of Computer Science 4 (2): 161-166, 2008 ISSN 1549-3636 © 2008 Science Publications