International Journal of Computer
Science and Engineering (IJCSE)
ISSN 2278-9960
Vol. 2, Issue 4, Sep 2013, 49-56

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

A REVIEW OF EFFICIENT CLUSTERING ALGORITHMS USED IN MANET

KANIKA GARG & LALIT KUMAR

Assistant Professor, Department of Computer Science and Engineering, Manav Rachna International University, Faridabad, Haryana, India

ABSTRACT

To become commercially successful, the technology must allow networks to support many users. One problem is that addressing and routing in ad hoc networks is not as easy as in the Internet. By introducing hierarchical addresses to ad hoc networks, this problem can be addressed. Clustering provides a method to build and maintain hierarchical addresses in ad hoc networks. In this paper, we survey several clustering algorithms. Clustering algorithms can also be used to build virtual backbones to enhance network quality of service.

KEYWORDS: Clustering Algorithm, Performance Metrics, MANET

INTRODUCTION

An ad hoc network is a multi-hop wireless communication network supporting mobile users without any existing infrastructure. These multi hop packet radio networks also named mobile ad-hoc networks (MANETs) have a dynamic topology due to the mobility of their nodes. A multi-cluster, multi-hop wireless network should be able to dynamically adapt itself with the changing networks configurations [1].A notable amount of energy is utilized every time a signal is sent and received by a mobile node. Many such signals and power are wasted to update the positional information of the nodes in a wireless scenario. Further bandwidth is also wasted by sending control signals rather than using it effectively for data communication [2].

In a cluster, objects are mutually closer to each other than to objects in other clusters [3]. The Cluster structure need to be maintained as the new mobile nodes may enter the network and the existing nodes may move out or lose their battery power [4]. It occurs in the case of both cluster-heads and member nodes. Perkins [14] observed that aggregating routing information is the key to Internet scalability. In particular, a node's IP address contains hierarchical information related to its location that can be used in routing. Due to the mobility of nodes in an ad hoc network, this is not as simple to accomplish. In a multi-hop packet-switched network, intermediate nodes are required to route packets between the source and destination if they (the source and the destination) are not directly connected.

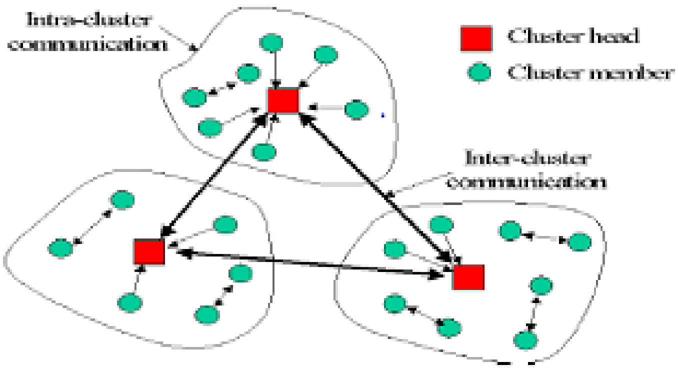

Figure 1: 2-Level Hierarchical Clustering

Cluster communications can be classified into intra-cluster and inter-cluster communications in two-level hierarchical clustering topology shown in Figure 1. In order to guarantee quality of service (QoS) and high throughput, TDMA is adapted for cluster communications by allocating a fixed time slot per a packet to each node over multiple channels.

- In Intra-cluster communication, packet transmission of each cluster member is processed within its cluster. Each cluster member has a packet to a random destination. If its packet destination is located within the same cluster, it transmits the packet to the destination directly (i.e., direct link). Otherwise, it forwards the packet to its own cluster head in order to save battery energy (i.e., uplink).
- In the inter-cluster communication, each cluster head broadcasts packets received from its members to their destination over specific channels of their destination similar to broadcast scheduling methods [6][7].

In mobile ad hoc networks, the movement of the network nodes may quickly change the topology resulting in the increase of the overhead message in topology maintenance.

Protocols try to keep the number of nodes in a cluster around a pre-defined threshold to facilitate the optimal operation of the medium access control protocol. The cluster-head election is invoked on-demand, and is aimed to reduce the computation and communication costs. A large variety of approaches for ad hoc clustering have been developed by researchers which focus on different performance metrics**.** The rest of paper is organized as follow, section II discusses about various clustering techniques, section III discuss about various performance parameters and metrics used in study of these techniques.

STUDY OF CLUSTERING ALGORITHMS

In Existing Algorithms, Clustering is Performed in Two Phases: clustering set up and cluster maintenance. Setup of clusters is accomplished by choosing some nodes to act as coordinators of the clustering process (cluster-heads). Then a cluster is formed by associating a cluster-head with some of its neighbors (i.e. nodes within the cluster-head’s transmission range) that become the member nodes of the cluster. Once the cluster is formed, the cluster-head can continue to be the local coordinator for the operations in its cluster. A common assumption for the clustering set up is that the nodes do not move while the cluster formation is in progress. This is a major drawback, since in real situation; no assumptions can be made on the mobility of the nodes. Once the nodes are partitioned into clusters, the non mobility assumption is released and several techniques are used to maintain the cluster organization in the presence of mobility i.e. clustering maintenance [8]. The existing clustering algorithms differ on the criteria for the selection of the cluster-heads. Many clustering protocols for mobile ad hoc networks (MANETs) have been proposed in the literature. A survey of different clustering algorithms is presented below:

Lowest ID Cluster Algorithm (LIC) [1]

Is an algorithm in which a node with the minimum *id* is chosen as a cluster-head. Thus, the *ids* of the neighbors of the cluster-head will be higher than that of the cluster-head. A node is called a gateway if it lies within the transmission range of two or more cluster-heads. Gateway nodes are generally used for routing between clusters. Each node is assigned a distinct *id*. Periodically, the node broadcasts the list of nodes that it can hear (including itself). A node which only hears nodes with *id* higher than itself is a cluster-head. The lowest-*id* node that a node hears is its cluster-head, unless the lowest-*id* specifically gives up its role as a cluster-head (deferring to a yet lower *id* node). A node which can hear two or more cluster-heads is a gateway else a node is an ordinary node. The Lowest-ID scheme concerns only with the lowest node *ids* which are arbitrarily assigned numbers without considering any other qualifications of a node for election as a cluster-head. Since the node *ids* do not change with time, those with smaller *ids* are more likely to become cluster-heads than nodes with larger *ids*. Thus, drawback of lowest ID algorithm is that certain nodes are prone to power drainage due to serving as cluster-heads for longer periods of time.

Max-Min d-Cluster Formation Algorithm [2]

This method minimizes the amount of data that must be passed from an outgoing cluster-head to a new cluster-head when there is an exchange. This provides a smooth exchange of cluster-heads rather than an erratic exchange.

Due to the large number of nodes involved, it is desirable to let the nodes operate asynchronously. The clock synchronization overhead is avoided, providing additional processing savings. The number of messages sent from each node is limited to a multiple of *d* the maximum number of hops away from the nearest cluster-head, rather than *n* the number of nodes in the network. This guarantees a good controlled message complexity for the algorithm. Additionally, because *d* is an input value to the heuristic, there is control over the number of cluster-heads elected or the density of cluster-heads in the network. The amount of resources needed at each node is minimal, consisting of four simple rules and two data structures that maintain node information over *2d* rounds of communication. Nodes are candidates to be cluster-heads based on their node *id* rather than their degree of connectivity. As the network topology changes slightly the node's degree of connectivity is much more likely to change than the node's *id* relative to its neighboring nodes. If a node *A* is the largest in the d-neighborhood of another node *B* then node *A*, *A* will be elected a cluster-head, even though node *A* may not be the largest in its d-neighborhood.

K-Hop Connectivity ID Clustering Algorithm (KCONID) [3]

Combines Two Clustering Algorithms: The Lowest- ID and the Highest-degree heuristics. In order to select cluster-heads connectivity is considered as a first criterion and lower ID as a secondary criterion. Using only a lower ID criterion generates more clusters than necessary. The purpose is to minimize the number of clusters formed in the network and in this way obtain dominating sets of smaller sizes. Clusters in the KCONID approach are formed by a cluster-head and all nodes that are at distance at most k-hops from the cluster-head. At the beginning of the algorithm, a node starts a flooding process in which a clustering request is send to all other nodes. In the Highest-degree heuristic, node degree only measures connectivity for 1-hop clusters. K-CONID generalizes connectivity for a k-hop neighborhood. Thus, when k = 1 connectivity is the same as node degree. Each node in the network is assigned a pair dID = (d, ID). 'd' is a node's connectivity and ID is the node's identifier. A node is selected as a cluster-head if it has the highest connectivity. In case of equal connectivity, a node has cluster-head priority if it has lowest ID.

Adaptive Cluster Load Balance Method [4]

Distribution of resources and transmission information to all clusters is basic idea of this algorithm. This algorithm can get load balance between various clusters. In HCC clustering scheme, one cluster head can be exhausted when it serves too many mobile hosts. It is not desirable and the CH becomes a bottleneck. So a new approach [4] is given. In hello message format, there is an "Option" item. If a sender node is a cluster head, it will set the number of its dominated member nodes as "Option" value. When a sender node is not a cluster head or it is undecided (CH or non-CH), "Option" item will be reset to 0. When a CH's Hello message shows its dominated nodes' number exceeds a threshold (the maximum number one CH can manage), no new node will participate in this cluster. As a result, this can eliminate the CH bottleneck phenomenon and optimize the cluster structure.

Adaptive Multi-Hop Clustering [5]

Is a multi-hop clustering scheme with load-balancing capabilities. Each mobile node periodically broadcasts information about its ID, Cluster-head ID, and its status (cluster-head/member/gateway) to others within the same cluster. With the help of this broadcast, each mobile node obtains the topology information of its cluster. Each gateway also periodically exchanges information with neighboring gateways in different clusters and reports to its cluster-head. This algorithm sets upper and lower bounds (U and L) on the number of cluster members within a cluster that can be handled by a cluster-head. When the number of cluster members in a cluster is less than the lower bound, the cluster is merged with one of the neighboring clusters. In order to merge two clusters into one cluster, a cluster-head always has to get the cluster size of all neighboring clusters. It prevents that the number of cluster members in the merged cluster is over the upper bound. On the contrary, if the number of cluster members in a cluster is greater than the upper bound, the cluster is divided into two clusters. The upper and lower bounds are decided by network size, mobility etc.

Mobility-Based Frame Work for Adaptive Clustering [8]

The purpose of this strategy is to support a more scalable routing infrastructure that is able to adapt to high rates of topological change. It partitions a number of mobile nodes into multi-hop clusters based on (a, t) criteria. The (a, t) criteria indicate that every mobile node in a cluster has a path to every other node that will be available over some time period 't' with a probability 'a' regardless of the hop distance between them. Cluster framework is based on an adaptive architecture designed to dynamically organize mobile nodes into clusters in which the probability of path availability can be bounded, and the impact of routing overhead can be effectively managed. The cluster organization supports an adaptive hybrid routing strategy that is more responsive and effective when node mobility is low and more efficient when node mobility is high. This is achieved using prediction of the future state of the network links in order to provide a quantitative bound on the availability of paths to cluster destinations. A metric which captures the dynamics of node mobility, makes the scheme adaptive with respect to node mobility.

Passive Clustering [12]

This algorithm is used to reduce control overhead in clustering. A clustering protocol that does not use dedicated control packets or signals for clustering specific decision is called Passive Clustering. In this scheme, a mobile node can be in one of the following four states: initial, cluster-head, gateway, and ordinary node. All the mobile nodes are with 'initial' state at the beginning. Only mobile nodes with "initial" state have the potential to be cluster-heads. When a potential cluster-head with "initial" state has something to send, such as a flood search, it declares itself as a cluster-head by piggybacking its state in the packet. Neighbors can gain knowledge of the cluster-head claim by monitoring the "cluster state" in the packet, and then record the Cluster head ID and the packet receiving time. A mobile node that receives a claim from just one cluster-head becomes an ordinary node, and a mobile node that hears more claims becomes a gateway.

Load Balancing Clustering (LBC) [13]

It provides a nearby balance of load on the elected cluster-heads to reduce power consumption. Once a node is elected a cluster-head it is desirable for it to stay as a cluster-head up to some maximum specified amount of time, or budget. The budget is a user defined restriction placed on the algorithm and can be modified to meet the unique characteristics of the system, i.e., the battery life of individual nodes. In this algorithm each mobile node has a variable, virtual ID (VID), and the value of VID is set as its ID number at first. Initially, mobile nodes with the highest IDs in their local area win the cluster-head role. LBC limits the maximum time units that a node can serve as a cluster-head continuously, so when a cluster-head exhausts its duration budget; it resets its VID to 0 and becomes a non-cluster-head node. When two cluster-heads move into the reach range, the one with higher VID wins the cluster-head role. When a cluster-head resigns, a non-cluster-head with the largest VID value in the neighborhood can resume the cluster-head function.

Power-Aware Connected Dominant Set [14]

Is an energy-efficient clustering scheme which decreases the size of a dominating set (DS) without impairing its function. The unnecessary mobile nodes are excluded from the dominating set saving their energy consumed for serving as cluster-heads. Mobile nodes inside a DS consume more battery energy than those outside a DS because mobile nodes inside the DS bear extra tasks, including routing information update and data packet relay. Hence, it is necessary to minimize the energy consumption of a DS. In this scheme Energy level (el) instead of ID or node degree is used to determine whether a node should serve as cluster-head. A mobile node can be deleted from the DS when its close neighbor set is covered by one or two dominating neighbors, and at the same time it has less residual energy than the dominating neighbors.

Weighted Clustering Algorithm (WCA) [16]

Selects a cluster-head according to the number of nodes it can handle, mobility, transmission power and battery power. To avoid communications overhead, this algorithm is not periodic and the cluster-head election procedure is only invoked based on node mobility and when the current dominant set is incapable to cover all the nodes. To ensure that cluster-heads will not be over-loaded a pre-defined threshold is used which indicates the number of nodes each cluster-head can ideally support. WCA selects the cluster-heads according to the weight value of each node. The weight associated to a node v is defined as:

$$Wv = w1\ \Delta v + w2\ Dv + w3\ Mv + w4\ Pv \quad (1)$$

The node with the minimum weight is selected as a cluster-head. The weighting factors are chosen so that $w1 + w2 + w3 + w4 = 1$. Mv is the measure of mobility. It is taken by computing the running average speed of every node during a specified time T. Δv is the degree difference. Δv is obtained by first calculating the number of neighbors of each node. The result of this calculation is defined as the degree of a node v, dv. To ensure load balancing the degree difference Δv is calculated as $|dv - \delta|$ for every node v, where δ is a pre-defined threshold. The parameter Dv is defined as the sum of distances from a given node to all its neighbors. This factor is related to energy consumption since more power is needed for larger distance communications. The parameter Pv is the cumulative time of a node being a cluster-head. Pv is a measure of how much battery power has been consumed.

Weight-Based Adaptive Clustering Algorithm (WBACA) [19]

The clustering approach presented in WBACA is based on the availability of position information via a Global Positioning System (GPS). The WBACA considers following parameters of a node for cluster-head selection: transmission power, transmission rate, mobility, battery power and degree. Each node is assigned a weight that indicates its suitability for the cluster-head role. The node with the smallest weight is chosen as the cluster-head. The weight of a node N is defined as:

$$WN = w1*M + w2*B + w3*Tx + w4*D + w5/TR \quad (3)$$

Where wl, w2, w3, w4, w5 are the weighing factors for the corresponding system parameters listed below: -

M: Mobility of the node

B: Battery power

Tx: Transmission power

D: Degree difference, and

TR: Transmission rate

This algorithm further allows no two cluster-heads to be one-hop neighbors of each other. Overlapping clusters are connected through Gateways (nodes connecting two cluster-heads). All the ordinary nodes are one-hop from their cluster-heads.

Connectivity, Energy & Mobility Driven Weighted Clustering Algorithm (CEMCA) [20]

The Election of the Cluster Head is Based on the Combination of Several Significant Metrics Such as: The lowest node mobility, the highest node degree, the highest battery energy and the best transmission range. This algorithm is completely distributed and all nodes have the same chance to act as a cluster head. CEMCA is composed of two main stages. The first stage consists in the election of the cluster head and the second stage consists in the grouping of members in a cluster. Normalized value of mobility, degree and energy level is calculated and is used to find the quality (normalized to 1) for each node. The node broadcasts its quality to their neighbors in order to compare the better among them. After this, a node that has the best quality is chosen as a cluster-head. In the second stage the construction of the cluster members set is done. Each cluster-head defines its neighbors at two hops maximum. These nodes form the members of the cluster.

CONCLUSIONS

Several clustering algorithms are reviewed which help organize mobile ad hoc networks in a hierarchical manner and their main characteristics are presented. With this survey it is seen that a cluster-based MANET has many important issues to examine, such as the cluster structure stability, the control overhead of cluster construction and maintenance, the energy consumption of mobile nodes with different cluster-related status, the traffic load distribution in clusters, and the fairness of serving as cluster-heads for a mobile node. These issues provide to perform analysis in these traits of MANETs as research area.

REFERENCES

1. Vincent Bricard-Vieu, Noufissa Mikou," Clustering Algorithm for MANETs based on mobility prediction", 3rd International Conference: Sciences of Electronic, Technologies of Information and telecommunications (SETIT), March 27-31, 2005 – TUNISIA.
2. S. Muthuramalingam, R.RajaRam, Kothai Pethaperumal, V.Karthiga Devi," A Dynamic Clustering Algorithm for MANETs by modifying Weighted Clustering Algorithm with Mobility Prediction", International Journal of Computer and Electrical Engineering, vol. 2, No. 4, pp. 1793-8163, August 2010.
3. Gerla .M., & Tsai .J. T. C. (1995), "Multi-cluster Mobile Multimedia Radio Network" ACM/Baltzer Wireless Networks Journal 95, vol. 1, pp. 255-265.
4. Ali Bokar, Muslim Bozyigit, & Cevat Sener," Scalable Energy-Aware Dynamic Task Allocation", International Conference on Advanced Information Networking and Applications Workshops (2009)
5. Ratish Agarwal, Dr. Mahesh Motwani," Survey of clustering algorithms for MANET", International Journal on Computer Science and Engineering, vol.1 (2), 2009, 98-104.

6. B. Zheng, X. Wu, X. Jin, and D.L. Lee "TOSA: A Near-Optimal Scheduling Algorithm for Multi- Channel Data Broadcast," in Proc. the 6th International Conference on Mobile Data Management (MDM'05), May 9-13, 2005.
7. S. Hameed and N.H. Vaidya "Efficient Algorithms for Scheduling Data Broadcast," *Wireless Networks*, vol. 5, pp. 183-193, 1999.
8. Jean-Fred, Eric-Myoupo —Idrissa Sow, "Clustering in Mobile Ad Hoc Networks With Some Nodes Having the Same Weight", Studia Information University. December 2007.
9. Zhigang Che1, Zhihui Ge, Ming Zhaos "A Load-based Queue Scheduling Algorithm for MANET", Journal of Communication and Computer, ISSN1548-7709, USA, Aug. 2006, Volume 3, No.8 (Serial No.21).
10. J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols", In *Proceedings of ACM/IEEE MOBICOM*, Dallas, TX, October 1998.
11. Byung-Gon Chun, "Evaluation of Scheduling Algorithms in Mobile Ad Hoc Networks", MS Thesis, Stanford University, May 2002.
12. C. Perkins, E. Royer, S. Das, "Ad-hoc on Demand Distance Vector (AODV) Routing", in IETF Internet Draft, February, 2003.
13. S. Papavassiliou, S. Tekinay, K. Malick, K. Walker, "Performance evaluation framework and quality of service issues for mobile ad hoc networks in the mosaic atd," in *Proceedings of* 21st *Century Military Communications Conference*, Oct. 2000, vol. 1, pp. 297–303.
14. C. Perkins, "Ad Hoc Networking", Addison-Wesley, 2001.
15. Bosheng Zhou, Jieyi Wu, Safeng Zhang," A Review of Routing Protocols for Mobile Ad Hoc Networks", Journal of Computer research and Development, 39(10), 2002: pp. 1168-1177.
16. K. Tang, M. Gerla," Fair Sharing of MAC under TCP in Wireless Ad hoc Networks", Proceedings of IEEE MMT'99, Oct., 1999.
17. S. Xu, T. Saadawi, "Does the IEEE 802.11 MAC Protocol Work Well in Multi-hop Wireless Ad Hoc Networks? ", IEEE Communications Magazine, 39(6), Jun., 2001.
18. Longhuang Xiao, Brahim Bensaou, "On Max-Min Fairness and Scheduling in Wireless Ad Hoc Networks: Analytical Framework and Implementation", In IEEE/ACM MobiHOC, Long Beach, CA, USA, 2001.
19. Curt Cramer, Oliver Stanze, Kilian Weniger, and Martina Zitterbart," Demand-Driven Clustering in MANETs".
20. A.Demers, S.Keshav, S.Shenker," Analysis and Simulation of a Fair-queuing algorithm", in *Proceedings of ACM SIGCOMM*, Austin, TX, September 1989.
21. M. Shreedhar and G. Varghese," Efficient Fair Queuing Using Deficit Round Robin", In *Proceedings of ACM SIGCOMM*, Cambridge, MA, October 1995.