Reliable Scheme for Cluster Head Election in Mobile Ad Hoc Networks

William Asiedu[1], Maxwell Dorgbefu Jnr.[2], Samuel Adu Gyamfi[3]
Department of Information Technology Education, College of Technology Education - Kumasi University of Education, Winneba, Ghana

Abstract

Ad hoc mobile devices are set of consistent heterogeneous devices that move in an independent fashion and communicate with one another over a wireless channel. These devices are presumed to have a non-deterministic mobility pattern. Clusters are formed by clubbing together nodes and heads are elected to manage, serve as a backbone, and route packets to other cluster heads. A lot of policies used in selection of cluster heads are biased in favor of some nodes. The result of the responsibilities of cluster heads, may deplete their energy faster due to higher number of communications with nodes on the network. This situation might cause them to drop out of the network. This paper therefore, proposes a reliable scheme for election of cluster heads by giving all nodes the opportunity to serve as a cluster head and provides enhancement to existing algorithms to minimize the unbalanced distribution of nodes under cluster heads and increase the active life of a node in a network. The scheme elects right cluster head and help nodes to relinquish their headship to other nodes if their power is low.

Keywords — *ad hoc network, load balancing, clustering, communications, policies*

I. INTRODUCTION

Mobile ad hoc network (MANET) is a type of network where nodes communicate with one another using a shared medium. The nodes communicate with each other through multi-hop shared radio channel. Ad hoc networks are easy to setup since no infrastructure is required, and thus a cost effective implementation. The nodes communicate with each other within their transmission range [1]. All nodes in this network take part in the communication process. The nodes in this type of network move freely, and topologies can change capriciously. The free movement of notes, and the frequent change of topology usually result in unpredictable topology as there is no heavy network infrastructure. MANETs can operate in standalone mode or connect to gateway devices to give Internet access to other nodes in remote areas. MANETs have numerous advantages as outlined in [3]. The many advantages of MANETs include law enforcement, disaster and relief, industrial and government applications, collaborative and distributed computing, sensor networks, and vehicular communications. In [4], the authors proposed two main forms of MANET; flat and hierarchical MANETs. In flat MANETs, all nodes have to store the topology of the network. This usually causes scalability problem, and only recommended for small networks. Hence robust clustering algorithms are needed to solve scalability problem and provide effective quality of service [2]. Hierarchical MANET on other hand divides the nodes into groups called clusters. The nodes are classified into three; namely cluster head, gateway and ordinary node or member. Cluster head is elected from the cluster and has responsibility of managing the routing and control messages within the cluster. The other nodes associate themselves with their neighboring cluster heads. The gateway nodes are also responsible for communication and forwarding of messages between clusters. Some of the advantages of clustering are efficient handling of mobility management, power management, shared use of applications within the cluster, spatial reuse of resources, virtual client support, better bandwidth utilization, aggregation of topology information, and minimal storage requirements for communication [5]. However, there are inherit traditional problems with wireless networks. These problems are not only limited to the dynamic nature of the network, link failure and packet loss, unpredictable topology, and energy constrained.

This paper proposes a novel clustering algorithm which solves the problems outlined above through an effective way of electing cluster heads in MANETs. This scheme also enables the formation of stable and scalable clusters and uses a technique that maintains the cluster structure thereby reducing messages in the group

In [5, 6], Ad hoc network clustered architecture organizes nodes into clusters, each governed by cluster head. The nodes in the cluster involves in message dissemination and routing. The messages are also forwarded to other cluster heads or base stations in sensor networks. They outlined a typical sensor network where sensor nodes forward messages through a base station which are basically two hops. Cluster network is good for sensor networks because of its inherent suitability for data fusion. Since all data gathered by members of the cluster can be fused at the cluster head and later communicated to a base station or other routers. Sensor networks are self-organizing, and therefore formation and election should be conducted by themselves [27]. The election

of the cluster head should be autonomous; thus every node should be given opportunity to be elected.
The rest of the paper is organised as follows. Related work done in the area of cluster-based routing is reviewed in Section 2. Section 3 discusses the clustering setup. Section 4 presents the proposed enhancement. Section 5 deals with performance evaluation and section 6 concludes the paper.

II. RELATED WORK

A lot of work has been done by researchers in the field on selecting cluster heads in MANETs. In [7] and [8], the authors presented on link cluster algorithm. In their work, the selection of cluster head is based on node with highest identity number among the group members in the network. The maximum connectivity algorithm proposed by the authors in [9], based their selection of cluster head on the maximum number of neighbor nodes, and the throughput. The system performance however decreases with the increase in the number of nodes that make up the network under investigation. The lowest-ID algorithm proposed in [8] is one of the popular algorithms employed in the cluster head selection; based on lowest virtual identification number. There have been several modifications on this algorithm to make selection of cluster heads and management of clusters more suitable, and power efficient. For example, in [10], [11] and [12] the authors affirmed it with the same concepts in their works. However, these schemes are not effective in electing the most suitable node as the cluster head.
Mobility is one of the unique features of MANETs [13, 14]. Mobility always lead to change in topology, cluster head re-election and route invalidation. The impact of mobility in MANETSs is outlined in [15]. In mobility-aware clustering, nodes with same speed are grouped together under same cluster head to ensure stability and collaborative activities. In [16], the authors designed mobility-based algorithm where each member, calculates the relative mobility values with respect to the neighboring nodes which are used for cluster head election Therefore nodes with low speed relative to their neighbors are elected as cluster heads. This leads to large communication overheads and large latency during cluster formation.
Another issue with MANET is extensive power utilization by the cluster head. The cluster head consumes a lot of power since it is involved in every routing and broadcasting in the network. If the elected cluster head keeps controlling the activities on the network for long time, it will deplete all its power which will lead to partitioning of the network and consequent disruption in communication on the network. In [17] and [18], the authors proposed a distributed heuristic clustering scheme for energy conservation in two-tiered MANETs, and energy aware based on DS-Marking algorithm respectively. Usually, the maximum number of hops between two nodes in a cluster is two. With these algorithms, if the network is densely populated the degree of the cluster head and the workload become high for the creation of clusters. In [19] a proposed max-min-D-cluster algorithm, where the cluster head form part of a d-hop dominating set, nonetheless there is no limit to the maximum size of each cluster and resulted in overloaded clusters.
In Distributed Cluster Algorithm (DCA) as proposed in [20], the authors introduced a weight-based scheme for the selection of a cluster head. In their work, they assumed a unique weight for each node in the cluster however, the technique for assigning weights to the nodes was not properly addressed. They only presented a distributed cluster algorithm that partitions the nodes of a multi-hop wireless network into clusters and adapt to changes in the network topology.

III.CLUSTER SET-UP ALGORITHM

To set-up a proficient cluster, a central or distributed (k, r) dominating set finding algorithm is used for selecting the nodes that act as coordinators of the clustering process. In the process of selecting dominating nodes, redundancy is achieved by choosing the value of the parameter k greater than one and parameter r allows increased local availability. The following are concerns that need to be addressed while designing and implementing SCAM clustering algorithm as proposed in [20]:

1. Selection of the minimal number of cluster heads, which yields high throughput but with low latency as possible.
2. Efficiency and stability of the created clusters.
3. Network scalability
4. Mechanism to prevent the clusters from growing too large. If the clusters grow too large, the load on the cluster head becomes too heavy for it to handle.
5. Maintenance mechanism for the existing clusters. Most existing clustering algorithms create new clustering structures from scratch after a specified time interval.

Algorithm1
Data: $G = (V, E)$ *and parameters* r *and* k
Result: DS, *Dominating Set*
Begin
    Initialization;
    $DS = \theta$
for each u ϵ V do
    Compute the suitability_value of node u, u. Suitability_Value;
    u.Required_Ch = k;
    Insert_Priority_Heap(H,u);
End
    While(H≠ θ) do
P=del_root(H);

```
p.Required_Ch=0;
DS=DS U p;
Find_Rhop_Neighbour(p,G,U_prd);
For (u ϵ U_prd) and (uϵH) do
        u.Required_Ch=u.Required_Ch 1;
        if u.Required_Ch ==0 then
        delete(H,u)
        endif
end
find_2rhop_Neighbour(p,G,U_p2rd);
        for each (uϵH) and (uϵU_p2nd) do
        compute u.Suitability_value;
        find_Rhop_Neighbour(u, G, V_yrd);
        for each(wϵV_urd ) do
        u.Suitability_Value=u.Suitability_Value+w.
        Required_Ch;
        end
        end
        makeheap(H)
        end
end
SCAM- (k, r) -DS algorithm
```

IV. PROPOSED ENHANCEMENT

We based our research on the above algorithm in setting up the cluster and incorporating the use of fuzzy logic as proposed in [24] and concept of load transfer in the determination of the cluster head. We propose an enhancement to the whole process of clustering.

The Clustering is divided into three phases:

- **i.** Election of the Cluster Head,
- **ii.** Selection of the Cluster Head, and
- **iii.** Load Transfer from one Cluster Head to another as in [25].

A. *Election of the Cluster Head*

The cluster head plays a major role by coordinating all activities on the network as explained earlier. The elected cluster head acts as a base station by forwarding packets and communicating with other cluster heads. It communicates with other clusters through respective cluster heads or through gateways by sending and receiving data, managing and controlling messages on the network. There have been several algorithms for cluster head selection as discussed in the related works section of this paper. The factors that influence an algorithm in cluster head selection as outlined in [22, 26] include geographical location of the node, stability, mobility of the node, energy, capacity and throughput of the node. The election of the cluster head is done using the Lower Identify (**LID) algorithm**.

Parameters for cluster head selection

- SOCH: set of all the cluster heads
- CHN: Cluster Head Node
- NT: Total number of nodes in the sample area
- CR: The nodes covered under a cluster head
- NR: Node Range (25 Units)
- Ni : Node under consideration
- NC: Set of respective Cluster Head that can take the current node under its own cluster

Implementation: Algorithm II

```
Begin NodeCHK
        for(x=0;x<=NT;x++)
                for(y=0;y<=NT;y++)
                   if(N_x < NR && N_x != CR)
                   N_x = CR for C_x
                  CHN_x = NC_x
                end if
        end for
        end for
end NodeCHK

Begin LIDelect
        for(i=0;i<=N;i++)
                if(NPID = lowest && N_x != CR)
                N_x = SOCH
                N_x = present Cluster Head
                end if
        end for
end LIDelect
```

B. *Selection of the Cluster Head*

The selection of the cluster head is based on a Fuzzy decision [21] made by the nodes that come under different cluster heads. For this fuzzy selection we use a few more parameters, namely work budget left and number of nodes under the cluster head. The nodes will decide their cluster heads based on the values of these two parameters.

Parameters

- SOCH: set of all the cluster heads
- CHN: Cluster Head Node
- NT: Total number of nodes in the sample are
- CVR: The nodes covered under a cluster head
- NR: Node Range (25 Units)
- Ni: Node under consideration
- NC: Set of respective Cluster Head that can take the current node under its own
- cluster
- NUN: Number of Nodes Under the cluster head
- WB: Work Budget left

Implementation: Algorithm III

```
Begin FuzzyCHK
        for(i=0;i<=SOCH; i++)
           A1 = AVG(WB)
           A2 = AVG(NN)
        end for
        for(i=0;i<=SOCH;i++)
           if(Ni belong to NC)
                T1= MAX(WBi from SOCH)
                T2 = MIN(NNi from SOCH)
                M1 = MOD(AVG(WB)-T1)
                M2 = MOD(AVG(NN)-T2)
```

```
                M3= M1*M2
                        if(MIN(M3))
                        Ni = CVR under Ci
                        end if
                end if
        end for
end FuzzyCHK
```

In the above algorithm, the average work budget left is calculated first and the average number of nodes under each cluster head. The maximum work budget left if the cluster heads are retrieved; thus, the maximum battery life span left. We store this value in T1. Similarly, we choose the cluster head with the least number of nodes under it and store it in variable T2. We then calculate the difference between the maximum WB left and NN. Here we used the MOD function since we want the difference between the values and not the actual value. We then multiply these values and store in an array M3. Now choosing the minimum value from this array will give us the desired cluster head for the current node. We multiply because if the WB is high but the difference between NN and AVG (NN) is low, we will get a lower value as compared to the rest of the combinations. The cluster head, having the lowest value will have a relatively higher work budget left and a lower number of nodes under it is selected.

C. *Load Transfer*

This part of the enhancements is done to reduce the effect of increased load when a new node gets admitted into the cluster. The above algorithm works well, but if the node tries to join at a later stage, the cluster heads tend to transfer this node to another cluster head which is relatively under loaded.

Parameters

- SOCH: set of all the cluster heads
- HN: Cluster Head Node
- NT: Total number of nodes in the sample are
- CVR: The nodes covered under a cluster head
- NR: Node Range (25 Units)
- Ni: Node under consideration
- NC: Set of respective cluster head that can take the current node under its own cluster
- NUN: Number of Nodes Under the cluster head
- WB: Work Budget left

Implementation: Algorithm IV

```
Begin TransferCHK
        for(i=0;i<=NC;i++)
                if(Ni belongs to NC)
                    T1= ( WBi from SC)
                    T2 = ( NUNi from SC)
                  M1 = MOD(AVG(T1-Ti))
                  M2 = MOD(AVG(T2-Ti))
                  M3= M1*M2
                  if(MIN(M3))
                        Transfer Ni to CHNi
                  end if
                end if
        end for
end TransferCHK
```

When a new request comes to a cluster head, it looks for all the cluster heads which can take this node under its own cluster. The cluster head compares its existing work budget left and number of nodes under it. So, the cluster head with relatively lower load will accept this incoming node.

V. PERFORMANCE EVALUATION

Simulation Bed

The test bench that we have used are:

- Mobility pattern is non-deterministic.
- Movement not more than 1 the range of a node per unit time.
- 200 x 200 Sq. Unit Areas.
- Size of the cluster is not known.
- Total number of nodes = {25, 50, 75,100}.
- Sample time = 100ms.
- Node Range = 25 Units (fixed).
- Work Budget = 5000 Units.
- Random Work deducted with a max work done of 10.

The NS2 software simulator tool is used to test the performance against other existing protocols. The simulated ad hoc network is composed of slowly changing network topology. The topology has a square area with length 200 and width 200. The network nodes are randomly distributed.

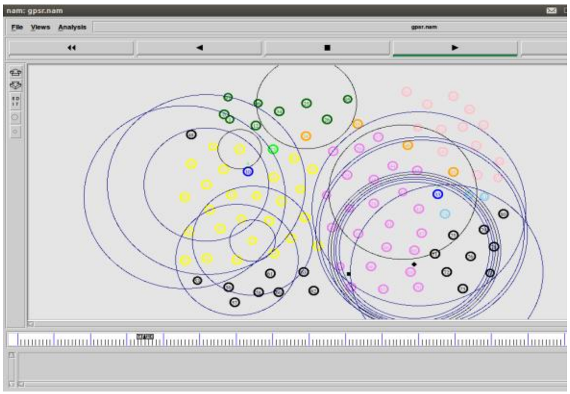

Fig 1. Clustering formation

Initial energy (Work Budget) of all the nodes in the network is 5000. The range of each node is 25 units which is taken as input to the program. The movement speed of a node can vary between 0 and 1 as shown in figure 1. The mobile nodes move according to the “random waypoint" model. Each mobile node begins the simulation by remaining stationary for pause time seconds. It then selects a random destination in the defined topology area and moves to that destination at a random speed. The random speed is distributed uniformly between zero and maximum speed of not more than 1 the communication range of a node. Upon reaching the destination, the mobile node pauses again for pause

time seconds, selects another destination, and proceeds as previously described. This movement pattern is repeated for the duration of the simulation.
One of the major parameters where we can easily asses the quality of the algorithm is the average cluster head time. This gives us a fair idea of the network lifetime. Hence, we compare our proposed algorithm with the existing algorithm.

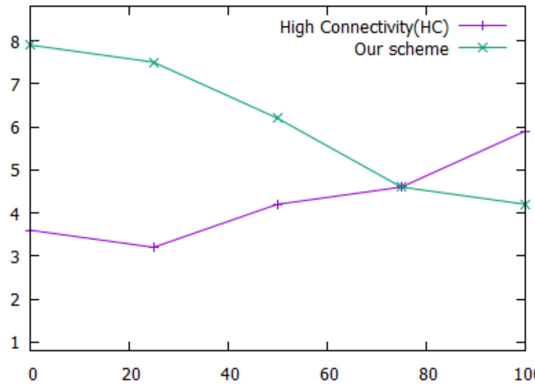

Fig. 2 High connectivity vs our scheme

The network shows low network lifetime in HC initially because the number of nodes are less. Less nodes amount to lower network lifetime. With the enhancement incorporated we see that the initial network lifetime i.e. with less number of nodes has improved because there was a better distribution of nodes under each cluster head.

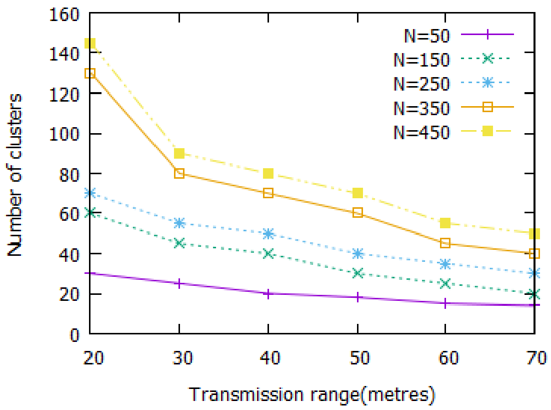

Fig. 3 Transmission range against number of clusters

Fig. 3 shows that the average number of clusters is relatively high when the transmission range is small. The results shown are for varying values of total number of nodes. When the transmission range increases, more and more nodes are connected to the same cluster head resulting in reduced number of clusters created. A smaller backbone is desirable for minimising the routing overhead. Hence, transmission power of a node is also a deciding factor for finding the quality of dominating nodes. When the transmission range is increased from 20 to 40 m, the number of clusters created is reduced considerably. But the rate of reduction in the number of clusters created gets reduced on further increase in the transmission range. The power consumption is high for higher transmission range. Hence, the recommended value of transmission range is between 30 and 40 m.

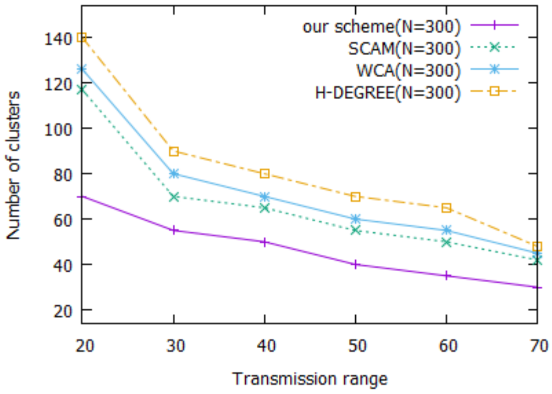

Fig. 4 number of clusters in our scheme, SCAM WCA, and H-degree

Fig. 4 compares the number of clusters formed for Our scheme, SCAM, WCA and H-degree as a function of the transmission range. The results show that the average number of cluster heads selected using our scheme is less compared with SCAM, WCA, and H-degree. This is because they create less number of clusters when the cluster radius increases and make use of the cluster merging process. But increased radius leads to increased cluster size, which adversely affects the performance. So the selection of radius is critical over here.

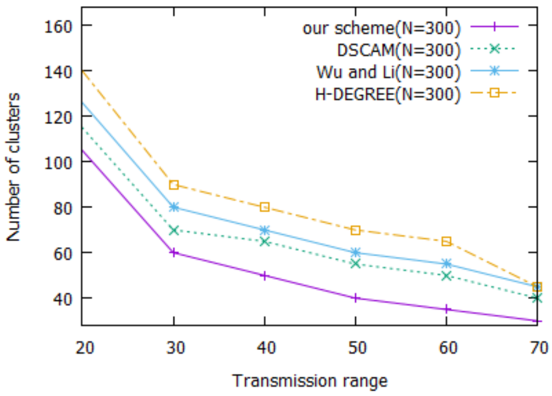

Fig. 5 comparison of number of clusters in our scheme, DSCAM, Wu and Li , and H-degree

Fig. 5 shows the number of clusters formed for our scheme, DSCAM, Wu and Li, and H-degree as a function of the transmission range. Our scheme creates less number of clusters compared with the other protocols. This is because our scheme forms less number of clusters with larger values of radius and uses pruning techniques to reduce the number of cluster heads.

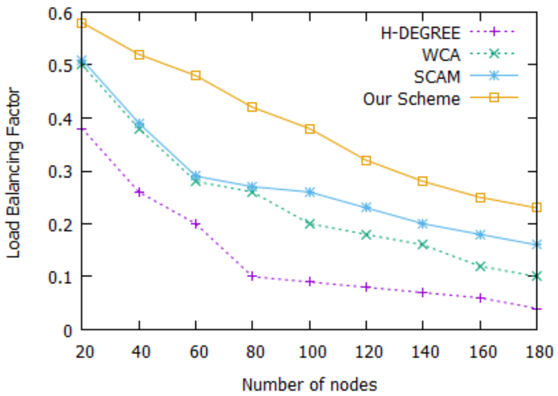

Fig. 6 comparing of load balancing factor with our scheme, SCAM, WCA and Hi-degree

As shown in Fig. 6, our algorithm clearly outperforms H-degree and WCA protocols in terms of Load Balancing Factor because it is a multi-cluster head and bounded distance clustering algorithm.

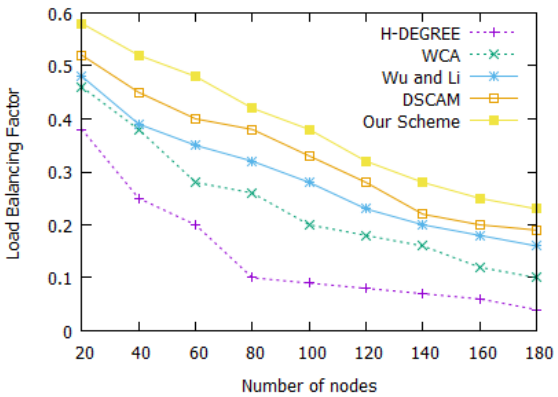

Fig. 7 Comparison of load balancing factor in our scheme, DSCAM, Wu and Li, WCA and Hi-degree

Fig. 7 compares the LBF of our scheme, DSCAM, Wu and Li, H-degree and WCA. One can observe that there is a reduction in LBF with increase in the value of radius. This is because when the value of radius increases more and more nodes are added to the same cluster head and it load gets increased.

VI. CONCLUSION

With the base algorithms being a bit inefficient in clustering in terms of the average cluster head time, we proposed an enhancement. The proposed enhancement helps the existing algorithm; LID to maintain a better network as compared to LID alone by distributing the load over to a relatively under loaded cluster heads and helps in the selection of a cluster head by monitoring two major parameters of the ad hoc network; the work budget left, and the number of nodes serviced by the cluster head. This improvement in the network lifetime is due to the change in the selection and election algorithm, and also to the transfer of node when there is a new admission to the system after it starts running. As the number of nodes in the sample area increases, the network lifetime decreases because, all the cluster heads have to handle greater responsibilities. But this fall in the network lifetime is stabilized as the nodes increase.

REFERENCES

[1] Conti, M., Giordano, S.: ‘Multi-hop adhoc networking: the theory’, IEEE Commun. Mag., 2007, 45, (4), pp. 78–86

[2] Chlamtac, I., Conti, M., Liu, J.-N.: ‘Mobile adhoc networking imperatives and challenges’, Ad Hoc Netw., 2003, 1, (1), pp. 13–64

[3] Akkaya, K., Younis, M.: ‘A survey on routing protocols for wireless sensor networks’, Elsevier J. Ad Hoc Netw., 2005, 3, (3), pp. 325–349

[4] Wu, J., Li, H.: ‘On calculating connected dominating set for efficient routing in ad hoc wireless networks’. Proc. Third ACM Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, pp. 7–14

[5] Basagni, S., Mastrogiovanni, M., Panconesi, A., Petrioli, C.: ‘Localized protocols for ad hoc clustering and backbone formation: a performance comparison’, IEEE Trans. Parallel Distrib. Syst., 2006, 17, (4), 292–306

[6] Er, I.I., Seah, W.K.G.: ‘Performance analysis of mobility-based d-hop (mobdhop) clustering algorithm for mobile ad hoc networks’, Comput.Netw., 2006, 50, (17), 3375–3399

[7] Baker, D.J., Ephremides, A.: ‘The architectural organization of a mobile radio network via a distributed algorithm’, IEEE Trans. Commun., 1981, 29, (11), pp. 1694–1701

[8] Ephremides, A., Wieseltheir, J.E., Baker, D.J.: ‘A design concept for reliable mobile radio networks with frequency hopping signaling’, Proc. IEEE, 1987, 75, (1), pp. 56–73

[9] Parekh, A.: ‘Selecting routers in ad hoc wireless networks’. Proc. IEEE Int. Telecommunications Symp., 1994

[10] Chiang, C.-C., Gerla, M.: ‘Routing in clustered multihop, mobile wireless networks with fading channel’. Proc. IEEE SICON 97, 1997

[11] Yu, J.Y., Chong, P.: ‘3hbac(3-hops between adjacent clusterheads): a novel non-overlapping clustering algorithm for mobile ad hoc networks’. Proc. IEEE Pacrim 03, August 2003, vol. 1, pp. 318–321

[12] Lin, C., Gerla, M.: ‘Adaptive clustering for mobile wireless networks’. IEEE JSAC, September 1997, vol. 15, pp. 1265–1275

[13] Choi, W., Woo, M.: ‘A distributed weighted clustering algorithm for mobile ad hoc networks’. Proc. IEEE ICIW, 2006

[14] Donald, A.A.M., Znati, T.: ‘Mobility based framework for adaptive clustering in wireless ad hoc networks’, IEEE J. Sel. Areas Commun.,1999, 17, (8), pp. 1466–1486

[15] Mahasukhon, P., Sharif, H., Hempel, M., Zhou, T., Wang, W., Chen, H.H.: ‘IEE 802.11b based ad hoc networking and its performance in mobile channels’, IET Commun., 2009, 5, (1), pp. 689–699

[16] Basu, P., Khan, N., Little, T.D.C.: ‘A mobility based metric for clustering in mobile ad hoc networks’. Proc. IEEE ICDCW 01, April 2001, pp. 413–418

[17] Hee Ryu, J., Song, S., Cho, D.-H.: ‘New clustering schemes for energy conservation in two tiered mobile ad-hoc networks’. Proc. IEEE ICC 01, June 2001, vol. 3, pp. 862–866

[18] Wu, J., Dai, F., Gao, M., Stojmenovic, I.: ‘On calculating power – aware connected dominating sets for efficient routing in ad hoc wireless networks’, J. Commun. Netw., 2002, 4, (1), pp. 59–70

[19] Amis, A.D., Prakash, R., Vuong, T., Huynh, D.: ‘Max-min-d-cluster formation in wireless ad hoc networks’. IEEE INFOCOM, 2000, pp. 1293–1302

[20] Basagni, S.: ‘Distributed clustering for mobile ad hoc network’. Proc. 1999 Int. Symp. Parallel Architectures, Algorithms and Networks (I-SPAN’99), 1999, pp. 310–315

[21] Basagni, S.: 'Distributed and mobility adaptive clustering for multimedia support in multi-hop wireless networks'. Proc. Vehicular Technology Conf., 1999

[22] Basagni, S., Turgut, D., Das, S.K.: 'Mobility adaptive protocols for managing large ad hoc networks', IEEE Int. Conf. Commun., 2001, 5, pp. 1539–1543

[23] Ghosh, R., Basagni, S.: 'Limiting the impact of mobility on ad hoc clustering'. Proc. Second ACM Int. Workshop Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitious Network, 2005

[24] Chatterjee, M., Das, S.K., Turgut, D.: 'Wca: a weighted clustering algorithm for mobile ad hoc networks', Cluster Comput., 2002, 5, (1), pp. 193–204

[25] Li, C., Wang, Y., Huang, F., Yang, D.: 'A novel enhanced weighted clustering algorithm for mobile ad hoc networks', Wirel. Commun., Netw. Mob. Comput., 2009, 1, pp. 1–4

[26] Wang, Y.-X., Bao, F.S.: 'An entropy-based weighted clustering algorithm and its optimization for ad hoc networks'. Third IEEE Int. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMOB 2007), 2007

[27] Wang, Y., Chen, H., Yang, X., Zhang, D.: 'Wachm: weight based adaptive clustering for large scale heterogeneous manet'. IEEE Conf.