Associativity based Clustering Algorithm in Mobile Ad Hoc Networks

Gang Zhao Xiao-Hui Kuang
Beijing Institute of System and Engineering
AnXiang Road, HaiDian District, Beijing
China

Abstract: - This mobility metric is subsequently used as a basis for cluster formation which can be used for improving the scalability of services such as routing, network management and security etc. A novel mobility metric for mobile ad hoc networks (MANET) was presented in this paper that is based on the associativity ticks of the nodes. Based on the use of this mobility metric for selection of clusterheads, a distributed clustering algorithm named MACA was proposed. The simulation results show that MACA outperforms the existing ones and is also tunable to different kinds of network conditions.

Key-Words: - Mobile ad hoc networks, clustering, link stability, relative mobility

1 Introduction

In recent years, mobile ad hoc networks have been received much attention due to their potential applications and proliferation of mobile devices [1]. Specifically mobile ad hoc networks refer to wireless multi-hop networks formed by a set of mobile nodes without relying on a preexisting infrastructure. In order to make an ad hoc network functional, the nodes are assumed to follow a self-organizing protocol, and intermediate nodes are expected to relay messages between two distant nodes.

Obtaining a hierarchical organization is a well-known and studied problem in mobile ad hoc networks [2]. It has been proven effective to solve several problems, such as minimizing the amount of storage for communication information (e.g. routing and multicast tables), reducing information update overhead, optimizing the use of network bandwidth, service discovery, network management and security etc. [1,2,4-9] In the cluster organization, certain nodes known as clusterheads are responsible for the formation of clusters, each consisting of a number of nodes, and also for maintenance of the network topology. The set of clusterheads is known as a dominate set. Due to the dynamic nature of mobile nodes, their association and disassociation to and from clusters perturb the stability of the network and thus reconfiguration of clusterheads is unavoidable. Thus, it is desirable to have a minimum number of clusterheads that can serve the network nodes, and to have a lower re-affiliation of nodes. An optimal selection of clusterheads is an NP-hard problem [10]. Therefore, various heuristics have been designed for this problem.

Mobility is the main cause for the changes of clusterheads and cluster memberships, but most of heuristics for clusterhead selection do not consider relative mobility. Over frequent clusterhead contention is another problem in current clustering algorithms. This paper presents a novel metric that based on link between the node and its neighbors, and presents a new clustering algorithm named MACA (Maximum Associativity Clustering Algorithm), which uses the new link stability metric for cluster formation. And in order to reduce frequency of clusterhead contention and re-affiliation, Clusterhead Contention Principle and Node Join Principle are proposed on the basis of link stability in this paper, which further improve stability of clusters. Simulation results demonstrate that in Random Waypoint Mobility Model, MACA is better than other algorithms such as Lowest-ID, Lowest-SPEED, MIX and Max-Degree in the load balance factor, frequency of re-affiliation and clusterhead update, MACA in Random Waypoint Model has the same good performance as WCA. Easy implementation as well as no needs for other devices to get the heuristic information guarantees that MACA will have wide range of applications. In mobile ad hoc networks, communications are often among teams. The team relationship makes it possible to partition the network into several groups, each with own mobility behavior. So, we simulate the clustering algorithms in most popular group mobility models, Reference Point Group Mobility Model (RPGM). Simulation results demonstrate that in RPGM, the performance of MACA is much better than other clustering algorithms□such as lowest ID, WCA and MIX et al.

The rest of the paper is organized as follows: in Section 2, a brief summary of previous related work is presented. Section 3 describes the approach we have adopted to calculate the link stability metric for each node and outlines the clustering algorithm that is based upon this link stability metric. Simulation results are illustrated in Section 4; and in section 5, we discuss some directions of future research and conclude the paper.

2 Previous Work

Several clustering algorithms have been proposed in mobile ad hoc networks. The Lowest-ID[4] and Max-Degree[7] algorithms are the most popular algorithms, but they do not result in stable clusters: in the former, a highly mobile node with low ID will cause server re-clustering while if the node moves into another region may pose danger to an exiting cluster. In the latter, depending on nodes' movement and traffic, the criterion values used in the selection process can keep on varying for each node, and hence result in instability. The performance of the two algorithms is studied in [3, 12]. C.R Lin et. al. [11] combined the above two algorithms and proposed MIX algorithms which choose the nods with high connectivity first, if degree of nodes are equal, then choose clusterhead based on ID of nodes.

Based on Lowest-ID algorithm, Basagni.[12] proposed a distributed clustering algorithm named DMAC, in which, each node is assigned weights in accordance with its suitability of being a clusterhead. A node is chosen to be a clusterhead if its weight is higher than any of its neighbor's; otherwise, it joins a neighboring clusterhead. Chatterjee et al. developed DMAC and proposed a new clustering algorithm named Weighted Clustering Algorithm (WCA), in which the ideal degree, transmission power, mobility, and battery power of mobile node are taken into consideration. Although DMAC and WCA take into many factors to choose clusterhead, the measurement criterion of them are not unified. So it is difficult to confirm its proportion of factor in weight to get better performance in applications. In addition computing the weights is expensive in the two algorithms. What's more, the Lowest-Speed Algorithm [8] is proposed, in which a node with the lowest speed is chosen to be clusterhead.

Because the above clustering algorithms do not take into relative mobility, when mobility of node increased, the stability of cluster formed by these algorithms become lower. Feng YX et al.[9] propose a clustering algorithm based on token mechanism and lowest ID, which takes relative mobility into consideration, but it does not discuss how to use relative mobility to improve the stability of cluster. Prithwish Basu et al. [12] present a novel mobility metric for choosing clusterhead, which is based on the ratio between the received power levels of successive transmissions measured at any node from all its neighbouring nodes. This metric manifests the relative mobility for the received power level reaches high when two nodes move to each other, while it gets low if two nodes move away. However this mobility metric uses power measurements, its accuracy depends heavily on how well a varying channel condition is modelled, and hence is not practical[14]. Sivavakeesar et al [18] proposed a novel clustering algorithms wherein a node is elected as a cluster head (CH) based on nodes having associativity-states that imply periods of spatial and temporal stability, but it does not validate the performance of the proposed algorithm.

3 Maximum Associativity Clustering Algorithm

Having taken into account the common deficiencies of all other approaches, the basic idea of MACA is that the clustering process should be aware of the mobility of the individual nodes with respect to its neighbors. A node should not be elected a clusterhead if it is highly mobile relative to its neighbors, since, in that situation the probability that a cluster will break and that re-clustering will happen frequently.

MACA selects a node as clusterhead that has maximum stable links. The link stability can be identified by the associativity "ticks", which is updated by data link layer protocol of the node. Every node periodically transmits beacons to identify itself and constantly updates its associativity ticks for every link in accordance with the node sighted in its neighborhood. A node is said to exhibit a high state of relative mobility when it has low associativity ticks. On the other hand, if high associativity ticks of a link are observed, the node is in the stable state with its neighbor even it has a high speed.

Before describing the detail of the algorithm, we first present the network model and the notation that is used throughout the paper.

3.1 Network Model

All the nodes in a mobile ad hoc network are assumed to have an omni-directional antenna and have the same transmission power. All links are bi-directional, i.e. if node A can hear B, then node B also can hear node A. Nodes share the same communication channel (e.g. same frequency band, same spreading code or frequency hoping pattern) to transmit and receive information at the same time. Without loss of generality, partitioning in the network is not allowed, because each partition can be treated as an independent network.

3.2 Notation and Definitions

We model a mobile ad hoc network by an undirected graph $G=(V,E)$ in which V is the set of wireless nodes and there is an edge $\{u,v\}\in E$ if and only if u and v can mutually receive each others' transmission. In this case and v are neighbors. The rest of the notation and the definitions are as follows.

ID_i : Global unique network identifier for node i. Each node is assumed to know its own ID.

$N(i)$: Set of neighbors of node i.

$M(i)$: Set of cluster members, its initial value is empty and only if node i is the clusterhead, it should refresh this set.

ch_i : Clusterhead ID of node i, its initial value is *nil*. If node i is the clusterhead, then $ch_i = ID_i$. If $ch_i = ID_j$, we say that node i belongs to clusterhead j.

T_w, T_c, T_h : Waiting timer, Clusterhead contention timer and beacon timer which are ordered as $T_w > T_c > T_h$.

In the algorithm, the node can be one of three states: *Undecided*, *Member* and *Clusterhead*.

U : Set of nodes in Undecided state.

Th_{member} : Threshold that defines the max number of cluster members in a clusterhead, which should content with its neighboring clusterhead. Its value is relative with density of nodes.

As described above, every node in the network should maintain associativity tick for every link between itself and its neighboring nodes. Let S denote the stability tick, then $S_{(u,v)}$ means the stability tick of link between node *u* and *v*, and it is maintained by node *u*. When a node receives beacon from its neighbor, the corresponding S increases by 1, and when the link is broken, the corresponding S deletes accordingly.

$TickTheshold$: Threshold that sets the minimum tick of stable link. If $S_{(u,v)} > TickTheshold$, the link between *u* and *v* is stable. Its value is relative with speed, transmission power and density of nodes.

w_i □Weight of node i, In MACA, the weight of node is the sum of ticks of stable link it has:

$$w_i = \sum_{j\in N(i)\wedge S_{(i,j)}\geq TickTheshold} S_{(i,j)}$$

3.3 Detail of MACA

This section describes MACA for the setup and the maintenance of a cluster organization in mobile ad hoc network. Since the algorithm is message driven, it is necessary to explain the messages transmitted along with the variation of node states:

- *HELLO*: node in Undecided state periodically broadcasts beacon message to its neighbors. Its format is $\{ID_i, w_i\}$.
- *CH (v)*: when node v is a clusterhead, it broadcasts this message periodically to make its neighbors aware its role. Its format is $\{ID_v, w_v, ID_v, Clusterhead\}$.
- *JOIN (v, u)*: when node u joins to clusterhead v or node u is a cluster member of clusterhead v, then it broadcasts this message periodically to its neighbors, its format is $\{ID_u, w_u, ID_v, \mathrm{Member}\}$.
- *RESIGN (u)*: when some clusterheads are adjacent, the winner in clusterhead contention will broadcast this message to its neighbors. Its format is $\{ID_u, w_u, ID_u, Contention\}$ □

The finite state machine diagram of MACA is shown in Figure 1. Each state in the state diagram is described below.

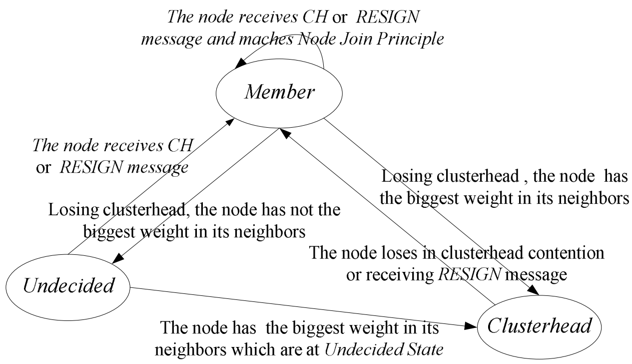

Figure 1 Finite State Machine diagram of *MLSCA*

Undecided State: At the start of clustering process, every node in the network is at Undecided State, and ch of every node is nil and the sets M , N are empty. Every node starts timer T_w , and broadcasts HELLO message by timer T_h , thus sets N and S are constructed by receiving HELLO message from its neighbors. When timer T_w is timeout, if

$$ID_i \in U \wedge \left(\forall ID_l \in (N(i) \cap U) \Rightarrow (w_i > w_l) \vee \left((w_i = w_l) \wedge (ID_i < ID_l)\right)\right)$$

, then node i enters the Clusterhead State and periodically broadcasts CH message. Otherwise it restarts timer T_w .

When node j receives *CH(i)* or *RESIGN(i)* message, if $j \in U$, then it decides whether to join to clusterhead i according to Node Join Principle (describe in section 3.5). if it chooses to join to clusterhead i , node j will broadcasts *JOIN(i, j)* message and set *i* to its ch , then enters the *Member State*. Otherwise, it continues waiting.

If the node in Undecided state receives *HELLO* or *JOIN* message, it only refreshes *N* and corresponding *S*.

Clusterhead State: In this state, every clusterhead broadcasts *CH* message to its neighbors periodically by timer T_h . When a clusterhead for example i receives a *JOIN(i, j)*, if $ID_j \notin N(i)$ then node i adds ID_j to $N(i)$, and if $ID_j \notin M(i)$ then node i adds ID_j to $M(i)$.

Because of mobility of node, some clusterheads will be neighboring. If a clusterhead receives a *CH* message, it starts timer T_c . On expiration of T_c , the node decides whether or not to start clusterhead contention by Clusterhead Contention Principle (describe in section 3.4). The winner retains *Clusterhead State*. The loser in clusterhead contention enters Member state, and its clusterhead is the winner.

Upon reception of an *RESIGN(u)* message, the clusterhead enters *Member State*, and set *u* to its *ch*.

If a clusterhead receives *HELLO* or *JOIN(k, j)* message, and if *k* is not equal to its ID, it only refreshes *N* and corresponding *S*.

Member State: In this state, a node broadcasts *JOIN* message to its neighbors periodically by timer T_h . Because of mobility, the cluster member may not hear its clusterhead. And if the node does not receive *CH* or *RESIGN* message from its clusterhead in timer T_w , or if its clusterhead has changed into other states, it loses clusterhead. At this time, the node with the biggest weight in its neighbors enters *Clusterhead State*; otherwise, it goes into *Undecided State*.

Also in this state, if a node receives *HELLO, JOIN* or *RESIGN* message, it only refreshes *N* and corresponding *S*. If it receives *CH* message, it decides whether to join to new clusterhead according to Node Join Principle (in section 3.5).

3.4 Clusterhead Contention Principle

Many previous solutions did not allow two clusterheads to be neighbors; they forced one of them to resign by clusterhead contention. Although clusterhead contention may decrease number of clusterheads and hops of communications, yet it also decreases the load balance factor of cluster and increases the number of cluster reconstructing and node re-affiliations. DMAC takes in consideration the disadvantage of clusterhead contention, but it only sets the number of clusterhead that are allowed to be neighbors to decrease frequency of clusterhead contention. When the number of neighboring clusterheads is more than the threshold, the clusterhead with the smallest weight will resign. In order to improve the stability of cluster, a clusterhead contention based on link stability is introduced in MACA. In this principle, clusterhead contention happens only if the link between two neighboring clusterheads is stable or if the number of cluster members is lower than threshold. The clusterhead with bigger weight will retain Clusterhead State. The Clusterhead Contention Principle is described below:

Assume u and v are clusterheads, and $u \in N(v), v \in N(u)$. There are three conditions in clusterhead contention:

1□when $|M(v)| \geq Th_{member} \wedge |M(u)| \geq Th_{member} \wedge S_{(u,v)} < TickTheshold$, both clusterheads give up contention. When $S_{(u,v)} > TickTheshold$, a contention occurred.

2□when $|M(v)| < Th_{member} \wedge |M(u)| \geq Th_{member}$, clusterhead *v* triggers the contention, and if $(w_u > w_v) \vee \left((w_u = w_v) \wedge (ID_u < ID_v)\right)$, node *v* gives up Clusterhead State. Otherwise, clusterhead *v* broadcasts *RESIGN (v)* message, and node *u* gives up Clusterhead State.

3□when $|M(v)| < Th_{member} \wedge |M(u)| < Th_{member}$, the node contents clusterhead just like the Lowest-ID algorithm.

3.5 Node Join Principle

The sum of distance within all its neighbors is considered in WCA in order to decrease re-affiliations of nodes. But distance information may be obtained by using the Global Positioning System (GPS), or by received power of packet. The former confines the fields that the clustering algorithm is applied to; the latter is not practical because of its inaccuracy [14]. P Krishna et. al.[15] imposed a rule named LCC on clustering process in order to reduce the number of re-affiliation: if a member of a cluster *C* moves within range of another cluster C' with a higher weight, it do not perform re-clustering unless it is the clusterhead of *C*.

In mobile ad hoc networks, the nodes with same direction and speed have stable link if neighboring. In MLSCA a node tries its best to join the clusterhead, which has stable link connecting them; similar to LCC, it also allows that the weight of member is higher than that of clusterhead after clusters formed, So re-affiliation takes place much less than other algorithms. The Clusterhead Contention Principle is described below:

The node *j* is in *Undecided State*, and receives *CH (i)* or *RESIGN (i)* message, if it come up to any conditions below:

1). $(w_j < w_i) \vee ((w_j \le w_i) \wedge (ID_j > ID_i))$ and (i, j) is stable link.

2).There is not existing node $u \in N(j)$ that matches

$$\left((w_u > w_j) \vee ((w_u = w_j) \wedge (ID_u < ID_j))\right) \wedge$$
$$\left((u \in U \wedge S_{(u,j)} > TickTheshold) \vee (ch_u = ID_u \wedge S_{(u,j)} > TickTheshold)\right)$$

, then node *j* joins to clusterhead *i* . Otherwise, it continues waiting.

As the node *j* in *Member State* receives *CH(i)* or *RESIGN(i)* message and if $(w_j < w_i) \vee ((w_j \le w_i) \wedge (ID_j > ID_i))$ and $(S_{(i,j)} > S_{(m,j)}) \wedge (ch_j = ID_m)$, node *j* joins to clusterhead *i*.

4 Simulation Study and Experimental Results

We implement our algorithm and many previous clustering algorithms, such as WCA, and MIX in a popular network simulator NS-2[16]. The Random Waypoint Mobility Model in ns-2 is used for simulation. To measure the performance of clustering algorithm, we identify two metrics (i) the number of re-affiliations, (ii) the number of clusterheads updates. The re-affiliation count increases when a node gets dissociated from its clusterhead or becomes a member of another clusterhead. The number of clusterheads updates increases when a node becomes a clusterhead or gives up being a clusterhead, at the same time the node gives the number of clusterheads.

In our simulation, the network size is 50 nodes; the size of scenario is calculated according to transmission ranges and network size. There are two parameters studied for varying: the transmission range varies between 30 and 180m, and the maximum speed varies between 0 and 50 m/s in Random Waypoint Mobility Model and average speed varies between 0 and 24m/s in Reference Point Group Mobility Model.

For MACA, Th_{member} is defined as half of the average of neighbors, and $TickTheshold = {2 * range}/{speed}$. In WCA, the weight of degree-difference, sum of distances, mobility and battery power marked respectively 0.7, 0.2, 0.05 and 0.05 respectively; they were used to compare with other algorithms by Chatterjee et. al..

In the Random Waypoint Mobility Model, a node begins by staying in one location for a certain period of time (pause time). Once time expires, the node chooses a random destination in the simulation area and speed. Upon arrival, the node pauses for a specified time period before starting the process again. This mobility model is widely used in many clustering algorithms. In this simulation, the pause time is 2s, and maximum speed is 20 m/s when transmission range varies, and transmission range is 150m when maximum speed varies.

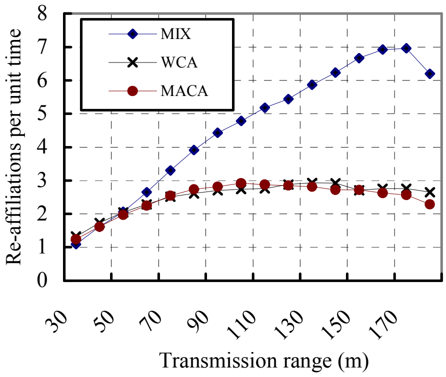

Figure 2. Re-affiliations per unit time vs. transmission range

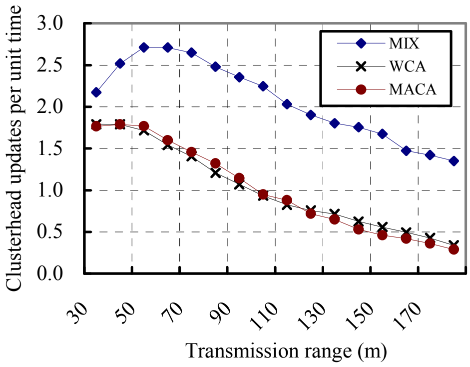

Figure 3. Clusterhead updates per unit time vs. trans. range

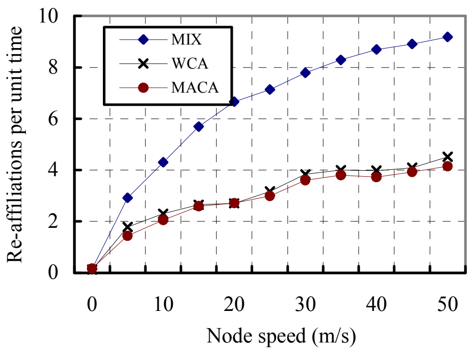

Figure 4. Re-affiliations per unit time vs. node speed

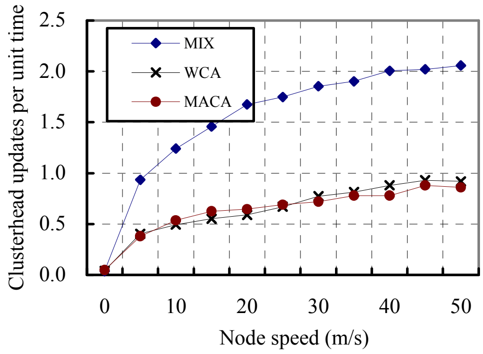

Figure 5. Clusterhead updates per unit time vs. node speed

Figure 2 and figure 3 show the relative performance of MIX, WCA and MACA in terms of number of re-affiliations and clusterhead updates for per unit time which change with transmission range where maximum speed of node is 20m/s. For low transmission range, the nodes in a cluster are relatively close to the clusterhead, and a detachment is unlikely. The number of re-affiliations increases as the transmission range increases and reaches a peak when transmission range is between 150 and 170m. Further increasing the transmission range results in a decrease in the re-affiliations since the nodes tend to stay inside the large area covered by the clusterhead, similar to the number of clusterheads updates; it reaches a peak when transmission range is between 50m and 70m. Figure 4 and 5 indicate the relative performance of these algorithms where maximum speed of node varies between 0 and 50m/s. As the maximum speed becomes larger, the nodes tend to move farther from their clusterhead, detaching themselves from the clusterhead and causing more re-affiliations per unit time and more clusterheads updates.

We can see from the simulation results that the number of re-affiliations and clusterhead updates in WCA and MACA is better than MIX algorithms. The main reason is that the frequency of invoking the clustering algorithms is lower in these two algorithms. MACA performs marginally better than WCA when transmission range or speed increases. However, the distance between nodes is a kind of heuristic in WCA, which is obtained difficultly in wireless network except using GPS, and using additional devices will impose restriction on the application of the clustering algorithm.

5 Conclusions

Many mobile ad hoc applications depend upon hierarchical structure, and clustering is the most popular method to impose a hierarchical structure in the Mobile Ad Hoc Networks. In this paper, we presented a new clustering algorithm, MACA that makes use of link stability as heuristics to choose clusterhead and maintain cluster. We can observe from simulation results that the performance of this algorithm is as good as WCA in Random Waypoint Mobility Model, but our algorithm needs not other device to get the heuristic information and could be implement more easily, therefore it have more profitable applications. Group motion occurs frequently in ad hoc networks and mobility impacts on cluster stability heavily, we evaluate the performance of many clustering algorithms in most popular group mobility model named RPGM, which is described in our another paper. The results manifest MACA outperforms other algorithms.

Our future work is to integrate the secure mechanism into MACA, and consider the key management scheme in larger mobile ad hoc network based on mobility-adaptive clustering.

References

[1] Z.J. Haas, J. Deng and B. Liang. "Wireless Ad Hoc Networks." Encyclopedia of Telecommunications, John Proakis, editor, John Wiley, 2002.

[2] J. Yu, P. Chong, A Survey of Clustering Schemes for Mobile Ad Hoc Networks, IEEE Communications Surveys, 7(1), 2005, pp 32-48.

[3] M. Chatterjee, S.K. Das and D. Turgut. “WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks”, Journal of Clustering Computing IEEE Vol. 5, No. 2, April 2002 pp.193-204

[4] M. Jiang, J. Li, and Y.C.Tay. “Cluster Based Routing Protocol (CBRP) Function Specifications”, IETF Draft, Aug 1999

[5] S. Basagni, K. Herrin and D. Bruschi. “Secure pebblenets.” In Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking and Computing, pages 156163. ACM Press, October 2001

[6] Ulas C. and Leandros Tassiulas. “Network Layer Support for Service Discovery in Mobile Ad Hoc Network” IEEE INFOCOM 2003

[7] A.K. Parekh. “Selecting Routers in Ad-Hoc Wireless Networks”, In Proceeding of the SBT/IEEE International Tele. Symposium 1994.8

[8] Zhang Wan-Yu, Yu Meng, Xie Li. “A Stable Cluster Based Hybrid Routing Protocol for Ad Hoc Mobile Networks” Chinese Journal of Computer 2001 Vol. 24, No. 12 1262-1271

[9] Feng YX, Wang GX. “A Clustering Algorithms Applied to the Management of Mobile Ad Hoc Network” Journal of Software China 2003 Vol.14□No. 1

[10] B. Bollobas, Random Graphs, Academic Press,1985

[11] C.R Lin and M. Gerla, “Adaptive clustering for mobile wireless networks”, IEEE J. on Selected Areas in Communications, 15, 7, 1997, 1265-1275

[12] S. Basagni. “Distributed Clustering for Ad Hoc Networks” Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks, June 1999, pp. 310-315

[13] P. Basu, N. Khan, and T. D. C. Little, "A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks", in Proc. of IEEE ICDCS 2001 Workshop on Wireless Networks and Mobile Computing, pp. 413-418, Phoenix, AZ, April 2001.

[14] Sivavakeesar S., Pavlou G. “A Prediction-Based Clustering Algorithm to achieve Quality of Service in Multi-hop AD HOC Networks” Proceedings of the London Communications Symposium (LCS), London, UK, pp. 17-20, September 2002.

[15] P. Krishna, N.H. Vaidya, M. Chatterjee, D.K. Pradhan. “A Cluster Based Approach for Routing in Ad Hoc Networks” ACM SIGCOMM Computer Communications Review, 1997

[16] Wireless and Mobility Extensions to the ns-2 Network Simulator. CMU Monarch Project. http://monarch.cs.cmu.edu/cmu-ns.html

[17] Tracy Camp, Jeff Boleng and Vanessa Davies “A Survey of Mobility Models for Ad Hoc Network Research” Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications vol. 2, no. 5, pp. 483-502, 2002

[18] Sivavakeesar, S. Pavlou, G. “Associativity-based stable cluster formation in mobile ad hoc networks” IEEE CCNC. 2005, pp196- 201