# Associativity-Based Stable Cluster Formation in Mobile Ad Hoc Networks

S. Sivavakeesar, G. Pavlou
Center for Communication Systems Research
University of Surrey
Guildford GU2 7XH
United Kingdom
{S.Sivavakeesar,G.Pavlou}@eim.surrey.ac.uk

*Abstract*—**This paper presents a framework for dynamically organizing mobile nodes (MNs) and electing a dominating-set in highly spontaneous large-scale mobile ad hoc networks (MANETs) with an aim to support location-based routing protocol. The proposed strategy is known as associativity-based clustering, wherein a node is elected as a cluster head (CH) based on nodes having associativity-states that imply periods of spatial and temporal stability. The heuristic that is used in the clustering process ensures a more dynamic, distributed and adaptive operation of our protocol. Furthermore, this heuristic considers mobility of nodes as the main criterion in the cluster head election process, and hence results in stable cluster formation. The heuristic used in our CH election process ensures that the responsibility of acting as cluster heads is evenly distributed among all the nodes, and hence it is fair. Simulation results demonstrate the performance advantages of our strategy.**

***Keywords-*** *Ad-hoc networking; Hierarchical clustering; Location-based routing..*

## I. INTRODUCTION

Mobile ad hoc networks (MANETs) are an actively evolving technology that allows the establishment of instant communication infrastructures for civilian and military applications. In multihop MANETs, the routing protocol is key to an efficient operation [14]. However, the design of an effective and efficient routing protocol in MANETs is extremely challenging because of mobility, limited battery energy, unpredictable behavior of radio channel, and time-varying bandwidth [8]. The effectiveness of any routing scheme depends on the timeliness and detail of the topology information available to them. However, in MANETs, the node mobility causes the frequent failure and re-activation of links, effecting a reaction from the network's routing algorithm to the changes in topology, thus increasing network control traffic and contributing to congestion. In other words, in ad hoc networks, significant rates of topological change are expected; consequently, the distribution of up-to-date information can easily saturate the network. Therefore, minimizing the exchange of information is crucial for efficient operation. Furthermore, information arriving late due to latency can drive network routing into instability. Since the rate of link failure is directly related to node mobility, greater mobility increases both the volume of control traffic required to maintain routes and the congestion due to traffic backlogs. This routing task becomes extremely challenging when the network grows in size, and where two additional problems such as increasing node-density and large number of nodes have to be faced and tackled [14]. High node-density, where a node is within a radio-range of a large number of neighbors, often leads to superfluous forwarding of routing related control traffic, and large network size necessitates the maintenance of large routing tables. These two features are inter-related and often affect the scalability of routing protocols. Thus, a crucial algorithm design objective to achieve routing responsiveness and efficiency is the minimization of reaction to mobility.

The future generation wireless networks are expected to evolve towards non-authority based, self-organized, large-scale MANETs [1]. In this work we envisage such large-scale deployment of multihop MANETs. Given that MANETs may comprise a large number of MNs, a hierarchical structure will scale better [1][3][4][6][8][13]. This fact has made researchers focus their attention in partitioning the multihop network into clusters, and electing cluster heads (CHs) or a dominating-set. (the dominating-set in our context doesn't strictly follow graph theory principles, and it refers to a set of CHs that can be reached by other neighbours not necessarily by single-hop but by single or k-hops at most). This clustering technique allows only a set of nodes (dominating-set) to handle the routing related information exchange in MANETs, and brings in a number of benefits as stated in [4][8][10][13]. This paper thus considers the dynamic way of organizing mobile nodes into clusters, and the election process of a dominating-set (a set of cluster heads (CHs)) in a multihop large-scale MANET environment. Our strategy differs from other similar approaches in two important aspects: a cluster head is elected based on spatial-associativeness and it is based on the notion of *virtual-clusters* we introduced in [1][2]. Location information may be obtained using the Global Positioning System (GPS), or a self-positioning algorithm as specified in [5]. In a MANET that uses scalable cluster-based services, network performance such as throughput and delay are tightly coupled with the frequency of cluster reorganization, and hence stable cluster formation is a prerequisite [8][9]. Our leader election heuristic

---

This work was undertaken in the context of the UK Engineering and Physical Sciences Research Council (EPSRC) Programmable Ad-hoc Networks (PAN - GR/S02129/01) research project.

takes this into consideration, and tries to elect stable cluster heads and thus form stable clusters.

The rest of this paper is organized as follows. Section II examines related previous work, and presents our motivation. Our novel associativity-based clustering technique is described in section III. Section IV presents the evaluation of the proposed scheme through simulation, and demonstrates that our clustering scheme leads to more stable cluster formation while incurring less control cost in comparison to other similar methods. Section V presents our conclusions and future work.

## II. RELATED WORK AND OUR MOTIVATION

A considerable body of literature has addressed research on clustering in MANETs [1][3][8][13]. While many clustering techniques with CH selection have been proposed, almost none of them consider node mobility as a criterion in the clustering process effectively [1][6][7][8]. As a result, they fail to guarantee a stable cluster formation. Choosing cluster heads optimally is an NP-hard problem [3]. Thus existing solutions to this problem are based on heuristic (mostly greedy) approaches and none of them attempts to retain the topology of the network. The most popular clustering algorithms available in the literature are the lowest identifier (Lowest-ID) and maximum-connectivity [3][6]. But these two, along with others, do not provide a quantitative measure of cluster stability. In the former, a highly mobile lowest ID CH will cause severe re-clustering; in addition, if this CH moves into another region it may unnecessarily replace an existing CH, causing transient instability. In the latter, depending on MN movement and traffic characteristics, the criterion values used in the election process can keep on varying, and hence also result in instability. This is also the case in the Lowest Distance Value (LDV) and the Highest In-Cluster Traffic (ICT) approaches [7]. Another scheme referred to as (∝, t)-clustering focuses on mathematical characterization of the probability of link and path availability as a function of a random walk based mobility model [8]. In the latter, it is considered that a link is active between two MNs at time $t_1 + t_0$ ($t_1 > 0$) given that there is an active link between them at time $t_0$. This scheme leads to ambiguity as to how big $t_1$ is and also it does not consider events that might have happened in the interval $t_1 + t_0$. A clustering scheme based on a mobility-metric is proposed in [9]. Since this bases the CH selection criteria on received power measurements, its accuracy depends heavily on how well a varying channel condition is modeled and, as such, it is not optimal.

As mentioned before, since mobility of nodes is the main cause of uncertainty in MANETs, our strategy considers mobility as the main criterion in the cluster head election process [1]. For this purpose, our CH election heuristic makes use of the concept of spatial-associativeness of a specific mobile node with respect to a particular *virtual-cluster* that we introduced in [1]. The concept of associativity was proposed and used as a routing metric for link reliability in [2]. In this work, the associativity-concept is used to reflect the degree of association stability between two mobile nodes over time and space. Nodes measure the connection stability by actively generating periodic beacons to signify their existence. In our scheme, however, every node tries to measure its spatial-associativity with respect to a specific *virtual-cluster* – as opposed to another node – by passively monitoring its presence in that cluster. It doesn't, however, involve periodic beacon transmissions. The CH election heuristic elects a node that has the highest associativity with respect to a specific *virtual-cluster* as the CH, as will be explained in section III. With this technique, stable clusters are formed, and as a result the frequency of cluster re-organization is minimal. This in turn conserves scarce bandwidth and battery energy. Also, stability is an important issue, since frequent cluster head changes adversely affect the performance of other protocols such as scheduling, routing and resource allocation that rely on it. The key objectives of our strategy are to achieve stable cluster topology with minimal communications overhead, and to operate asynchronously in a distributed manner.

## III. ASSOCIATIVITY-BASED CLUSTERING PROTOCOL

Having taken into account the common deficiencies of other approaches, our algorithm selects a MN as CH, if it satisfies the following criteria: 1) it has the highest spatial-associativity with respect to a specific *virtual-cluster*, in comparison to other MNs within the same cluster (see equation (4) below), and 2) it has the minimum distance from the respective *virtual-cluster* center (VCC). The first requirement tries to ensure that a highly mobile MN is not elected as a CH. The second is to ensure that by being located very close to a VCC, the CH can have a uniform coverage over a specific *virtual-cluster*. This in turn ensures that in subsequent CH changes, the area covered would not be impaired [1].

In order to make our clustering mechanism scalable, we wish to make use of the notion of *virtual-clusters* we introduced in [1]. The idea is that a geographical area is divided into equal regions of circular shape, as depicted in Fig.1, in a systematic way that each MN can determine the circle it resides in if location information is available. In our scheme each *virtual-cluster* is supposed to have a unique identifier based on the geographic location, which can be calculated using a publicly known hash-function. Each MN is supposed to have a complete picture of the locations of these *virtual-clusters* and their centers [1].

Our clustering protocol does not involve any extra control signaling; instead, periodic *HELLO* dissemination – as in other similar approaches – is enough [10]. This clustering is to facilitate electing a dominating-set (CHs). In order to maintain stable clusters, a new associativity-based criterion is used to elect CHs [2]. Accordingly in our clustering scheme; a node is elected as a cluster head, if it has the highest associativity-state with respect to its present *virtual-cluster*, and stays very nearer to its *virtual-cluster*-centre (VCC), when compared to other nodes in the same cluster. This implies period of spatial, temporal, and connection stability. For this purpose, each MN periodically monitors its current speed, and whenever its speed is zero, it will start measuring its "associativity". This is because after an unstable migration period, there exists a period of stability, where a mobile node will spend some "dormant time" or "residence-time" within the *virtual-cluster* before it starts moving again [2].

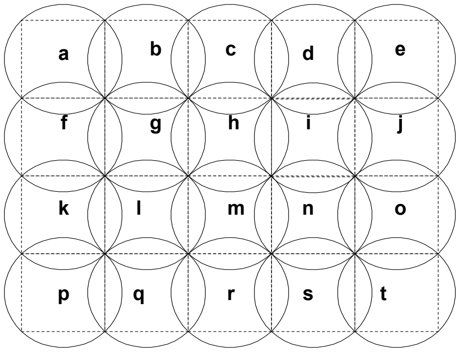

Figure 1. Concept of *Virtual-Clusters*.

When a MN is stationary, it passively measures its associativity with respect to a particular *virtual-cluster* by periodic "ticks" that takes place every ASSOCIATIVITY_TICK_PERIOD. This process doesn't however involve any transmission at all. In this way, any node X within the k^th^ *virtual-cluster* that has its total number of ticks ($n_{xk}$) greater than $A_{threshold}$, will exhibit higher degree of "associativeness", and hence have greater "dormant time". If, however, the speed of the MN is monitored to be greater than $\mu_{TH}$ (a system parameter), its number of ticks will be made to zero. The heuristic that is used by our clustering scheme is given by equation (4). Any node X determines the criterion value ($\Omega_{xk}$) in $k^{th}$ cluster by calculating the following:

- Each MN has to calculate its distance from the centre (VCC) of a particular *virtual-cluster*. Assuming node X, whose location co-ordinates at time 't' are ($x_{xk}(t)$, $y_{xk}(t)$), in the $k^{th}$ *virtual-cluster*, whose center's Cartesian co-ordinates are ($x_{ck}$, $y_{ck}$), its distance at time 't' can be calculated by :

$$d_{xk}(t) = \sqrt{(x_{xk}(t) - x_{ck})^2 + (y_{xk}(t) - y_{ck})^2} \tag{1}$$

- Each MN is supposed to store the "residence-time" or "dormant- time" in the last m number of clusters it has visited. This is basically the time period from the instance at which the MN's velocity is zero within a particular *virtual-cluster* and the instance at which it is more than $\mu_{TH}$. Then the mean "dormant-time" in terms of number of "ticks" ($N_{mean_X}$) is calculated as follows. Assuming that the "dormant-time" of node X in the $j^{th}$ *virtual-cluster* is $R_{X_j}$. Then node X's mean "dormant-time" ($R_{mean_x}$) is determined by considering its "dormant-times" in the last m number of *virtual-cluster* as given by equation (2). Then the mean "dormant-time" in terms of number of "ticks" is derived from equation (3). With this, the clustering criterion value ($\Omega_{xj}$) for node X in *virtual-cluster* j is determined from equation (4).

$$R_{mean_X} = \frac{\sum_{j=1}^{j=m} R_{X_j}}{m} \tag{2}$$

$$N_{mean_X} = \frac{R_{mean_X}}{ASSOCIATIVITY\_TICK\_PERIOD} \tag{3}$$

$$\Omega_{xj} = \begin{cases} \dfrac{N_{mean_X} - n_{xj}}{d_{xj}(t)} & \forall d_{xj}(t) \neq 0, n_{xj} \neq 0 \\ \dfrac{N_{mean_X} - n_{xj}}{d_{\min}} & \forall d_{xj}(t) = 0, n_{xj} \neq 0 \\ \dfrac{1}{d_{xj}(t)} & \forall d_{xj}(t) \neq 0, n_{xj} = 0 \\ \dfrac{1}{d_{\min}} & \forall d_{xj}(t) = 0, n_{xj} = 0 \end{cases} \tag{4}$$

Accordingly, any node X that has the highest value for the clustering criterion $\Omega_x$ is elected in either a centralized way or a distributed way depending on whether the present CH is available or not respectively [1]. The formula, given by equation (4), tries to ensure that the resulting clusters are more stable, and have uniform coverage by the respective CHs. If the CH lies very nearer to a VCC, it can have a uniform coverage, and hence ensures that all member nodes of a *virtual-cluster* are connected to this CH directly or via k-hops, where k is bounded by $D/(2R_{TX})$, and D is the diameter of the *virtual-cluster*, $R_{TX}$ is the transmission radius of a node (the value for D is chosen such that $D \approx 2R_{TX}$). $\Omega_x$ is proportional to expected "residence-time" or "dormant-time", and inversely proportional to distance from respective VCC. The system parameter $d_{min}$ ($\neq 0$) is the minimum value that $d_{xk}(t)$ can take.

Nodes run the CH election heuristic asynchronously; due to the large number of nodes involved, it is desirable to let the nodes operate asynchronously. With this the clock synchronization cost is avoided. Each *HELLO* message, periodically broadcast by the CH – say every CH_HELLO_INTERVAL – carries the ID of the *virtual-cluster* (VID) it covers, the VCC, the cluster's radius and the neighbor-table, the latter being the set of cluster members [10]. Whenever a new MN receives this message from a CH, it can send a *JOIN* message immediately, if it is within the *virtual-cluster*. The new MN includes in the *JOIN* message its heuristic value (as given by equation (4)) with respect to its current *virtual-cluster* 'k', and its location information. Whenever a CH receives a *JOIN* message, it checks first if the MN is within its *virtual-cluster*. If it is, the CH includes it in the cluster, and appends its information to the neighbor-table. If, on the other hand, the MN is not within the *virtual-cluster*, it will simply not be included. In either case, the MN has to wait for at least the next two successive CH_HELLO_INTERVAL periods to check whether it has been included. If not, it has to re-transmit the *JOIN* message. From the periodic neighbor-table that a CH broadcasts, each member of a cluster can build its own neighbor-table. A MN can be a member of up to four maximum adjacent *virtual-clusters*. This specific MN would then behave as gateway or forwarder between those clusters [1][7]. Having become a member, each MN within a particular *virtual-cluster* is supposed to disseminate a *HELLO* message to

0-7803-8784-8/04/$20.00 © 2004 IEEE

its respective CH periodically – say every MN_HELLO_INTERVAL, where MN_HELLO_INTERVAL > CH_HELLO_INTERVAL. In the *HELLO* message, each member specifies whether it is acting as a gateway or as an ordinary node. These control messages are relayed by intermediate MNs only within the *virtual-cluster*. On the other hand, periodic *HELLO* messages by CHs are unicast by gateways between CHs of adjacent *virtual-clusters* to an extent that can be limited for scalability. This is to enable CHs to get the topology information of adjacent clusters. The unique aspect of our protocol is that, whenever a present CH knows that it is going to leave the *virtual-cluster* that it is currently serving, it will select a member node that has the highest value for the criterion value as its successor, and inform about it to the cluster members through the SUCCESSOR packet. In this case, the CH is elected in a centralized manner. The present CH can decide that it is going to leave the cluster, when its monitored speed at a moment exceeds $\mu_{TH}$. If, on the other hand, there is no CH in a specific virtual-*cluster*, because of abrupt failure or error in associativity-based prediction, the cluster head election heuristic is executed in a more distributed way as follows. Accordingly, whenever an ordinary MN notices no *HELLO* message from its CH during two consecutive CH_HELLO_INTERVAL periods. The first MN to notice this will assume duty as the temporary CH, and it will immediately trigger the CH changeover event by broadcasting "*CH Changeover Event*" packet. On seeing this event, each node calculates it current heurist-value within the present *virtual-cluster* and disseminate such information through *HELLO* packets. Accordingly, each MN will become aware of other MNs' **$\Omega$-**values. Each MN then compares its own value with that of each MN of the same *virtual-cluster*, and one that has the highest value for '**$\Omega$**' will be elected as the new primary CH. The new CH will then start broadcasting *HELLO* message as usual. If however, an ordinary MN has not received any of the above control messages for more than four consecutive CH_HELLO_INTERVAL periods, then it will elect itself as the CH. In this algorithm, if more than two MNs have the same value for CH election criterion, **$\Omega$,** the one with the lowest ID will be selected as the new CH. Unlike in any other clustering algorithm, our algorithm has another unique feature in that whenever a CH leaves the *virtual-cluster* it has served, it will loose its CH status. In this way this algorithm ensures that no other visiting MN can challenge an existing CH within a particular *virtual-cluster*, and thus causing transient instability. All aspects of our strategy ensure that stable CHs are elected, and thus stable clustering is resulted in. This clustering scheme is thus fully distributed, where all the nodes share the same responsibility and act as CHs depending on the circumstances.

## IV. EVALUATION THROUGH SIMULATION

The scalability of our clustering protocol is assessed in terms of i) increasing node-count, ii) increasing average node-density, and iii) increasing average node speed. We chose Lowest-ID, maximum-connectivity (Max-Connect), LDV algorithms – the most popular clustering protocols found in the literature – in our attempt to compare the performance of our strategy. For this purpose, we implemented our algorithm along with the other three in GloMoSim [1][12]. The distance between any two VCCs in our scheme is 200m, and the diameter (D) of a *virtual-cluster* is 284m. Each node moves using a random waypoint model, with a constant speed chosen uniformly between zero and maximum speed, which is here taken as 10 $ms^{-1}$. The pause time takes a value that is exponentially distributed with mean 30 seconds. Each scenario was run for a 300 simulated seconds. Lowest-ID, LDV, and maximum-connectivity clustering algorithms form 2-hop clusters. Since it was necessary to ensure that clusters formed by all the schemes cover approximately equal area, the transmission range of each MN is set to 71m. The link capacity takes a value of 2 Mbps. The important simulation parameters are listed in Table 1. The simulation work attempts to compare the performance of our clustering algorithm with the Lowest-ID, maximum-connectivity, LDV clustering algorithms, in terms of the stability of clusters formed and control cost incurred. The cluster instability is measured by determining the number of times each MN either attempts to become a CH or gives up its role as a CH.

In the first-set of simulations, the scalability of the clustering protocols is measured in terms of increasing node-count. In order to properly see the effect of increasing network nodes on the clustering algorithms, the terrain-area is also increased with an increase in the number of nodes, so that the average node-density is kept constant in this set of simulations. The number of nodes in this case is varied from 25, 100, 225, 400 and 625. The terrain-area size is varied such that the average node degree remains the same and accordingly 200X200 $m^2$, 400X400 $m^2$, 600X600 $m^2$, 800X800 $m^2$ and 1000X1000 $m^2$ are selected for each scenario. Fig.2 shows the frequency of CH changes by each MN, and hence measures the (in)stability associated with each clustering algorithm as a function of increasing number of nodes. (The lower the frequency of CH changes, the more stable the cluster is). As it can be seen from Fig. 2, our clustering algorithm leads to more stable cluster formation. The average number of CH changes, which occurred per 100s, increases in the other three algorithms with the number of MNs. On the other hand, in the case of our clustering algorithm this increase is lower.

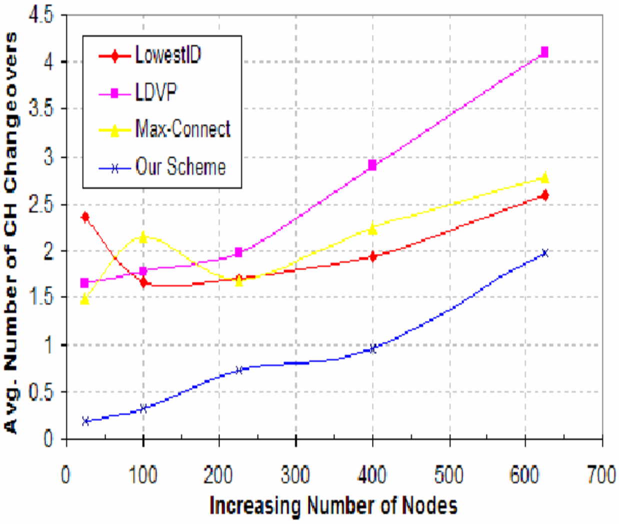

Figure 2. Cluster (In)Stability as a function of Number of Increasing Node-count.

TABLE I. TABLE 1. SIMULATION PARAMETERS

| Parameters | Value |
|---|---|
| Speed Range | 0 – 10 ms$^{-1}$. |
| Transmission Range | 71 m |
| Radius of a *Virtual-Cluster* | 142 m |
| MAC protocol | IEEE 802.11 |
| Simulation time | 300 S |
| ASSOCIATIVITY_TICK_PERIOD | 0.1 S |
| HELLO_INTERVAL by a CH | 3 S |
| HELLO_INTERVAL by a non-CH | 6 S |

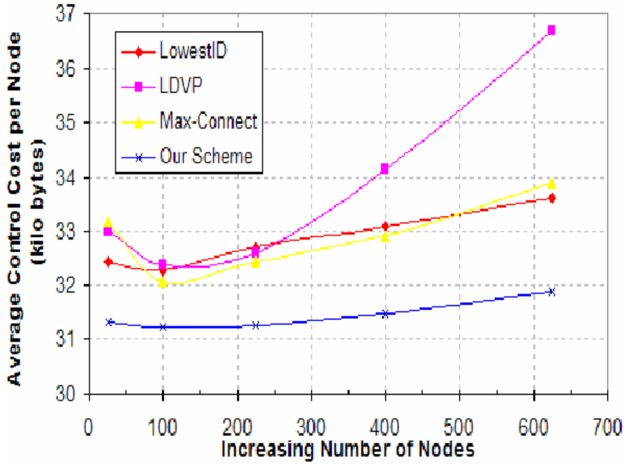

Figure 3. Average Control Cost Incurred per Node as a function of Increasing Number of Node-count.

Fig. 3 tries to compare the Lowest-ID, maximum-connectivity, LDV and our clustering algorithms in terms of the control cost incurred per node in kilo bytes, when the number of node increases. As it can be seen from Fig. 3, the control cost incurred per node in all the schemes tends to increase with increasing number of nodes. But in our scheme this increase is very small, and lower than those of the other three schemes. In the second-set of simulations, the scalability is measured in terms of increasing average node-density. In this case, the terrain-area is kept constant at 1000X1000 m$^2$, while the number of nodes in the given area is increased.

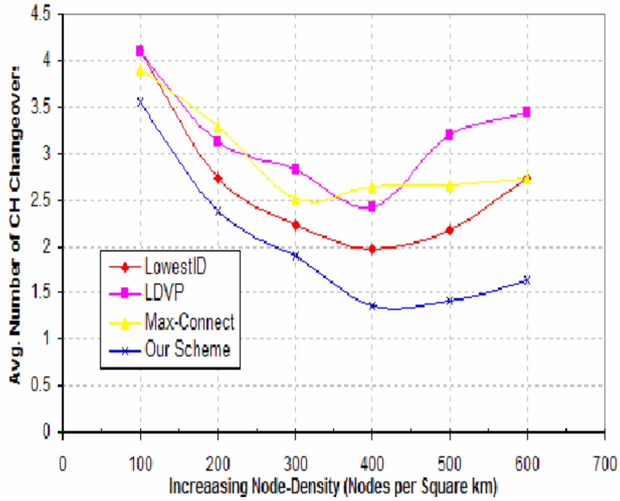

Figure 4. Cluster (In)Stability as a function of increasing number of node-density.

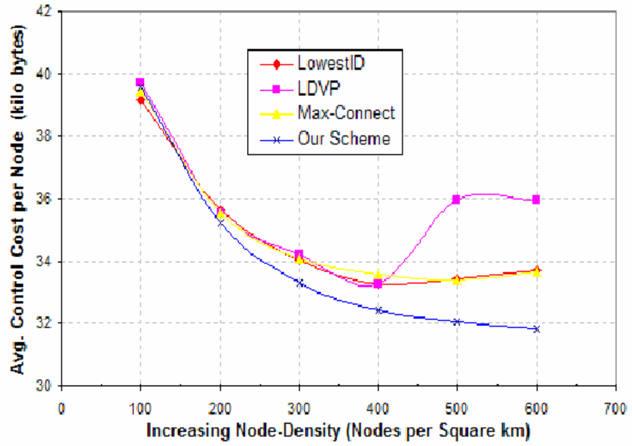

Figure 5. Average Control Cost Incurred per Node as a function of Increasing Node-density.

The measure of clustering stability as a function of increasing node-density is shown in Fig. 4. In all the four schemes, the cluster stability is very low when there is lower node-density. This can be due to the reason that when the node-degree is low, improper nodes are elected as cluster heads and as a result moving nodes may create transient instability. However, when the node-degree is moderate, all the clustering protocols try to converge quickly with the selection of proper heads. On the other hand, when the network becomes denser, the clustering algorithms take longer time to converge due to increased control traffic, and hence this affects the clustering stability. However, in our strategy, since the CH election heuristic takes associativity, and thus mobility, into consideration, only stable nodes are elected as CHs, and hence results in improved cluster stability. Further, in our strategy, there is a limit for the number of messages sent between nodes, and it is bounded by $\Theta(D/2R_{TX})$. This is due to the reason that our CH election heuristic ensures a node that lies very nearer to a *virtual-cluster*-center (VCC) to be elected as a CH with high probability. This condition ensures that any node is away from its respective CH by only a maximum of $D/2R_{TX}$ hops, and hence the number of messages sent from each node is limited to a multiple of $D/2R_{TX}$ in most cases. Messages are relayed by intermediate nodes only if they originate from the same *virtual-cluster* (i.e., relaying is spatially limited). Furthermore, unlike in other schemes, there is a control over the number of cluster heads elected in our scheme, and it is proportional to the number of *virtual-clusters* that we have in a given area. All these desirable features prevent arbitrary improper nodes to become CHs in our scheme, and thus help to improve cluster stability. Fig. 5 shows the clustering cost incurred by a node when the node-density increases. As can be seen, when the network is sparsely-connected, the average control cost incurred by a node tends to be high in all the four schemes. However it tends to decrease as the network becomes denser. While this control cost continues to decrease in our scheme, the same is not expected in the other three schemes. Instead, the control cost starts increasing after the node-density has reached a specific value (425 nodes per square km) in the network considered. The reason for this behavior is again same as the one given for Fig. 4.

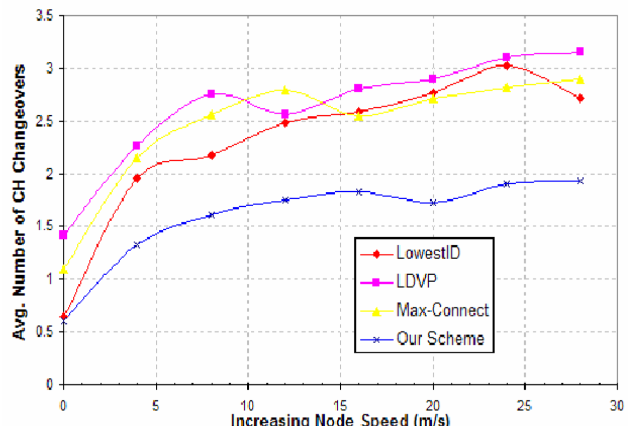

Figure 6. Cluster (In)Stability as a function of node Speed.

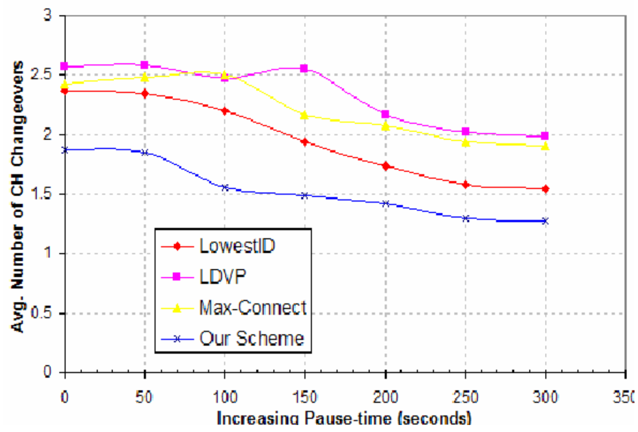

Figure 7. Cluster (In)Stability as a function of increasing Pause-time.

Figures 6 and 7 show the stability and scalability improvement of our strategy in terms of node-mobility. In the case of Fig. 6, the pause-time is exponentially distributed with mean value of 30 seconds, while the maximum speed of a mobile node is increased from 0 to 28 $ms^{-1}$. As can be seen from Fig. 6, although the stability is impaired by increasing node speed, the extent to which it is affected is very low in our scheme. Fig. 7 again tries to measure the stability of all four schemes in terms of node-mobility. However, in this case, the maximum speed of a node is kept constant at 10 $ms^{-1}$, while the pause-time is increased from 0 to 300 seconds (exponentially distributed). As pause-time increases, the stability of a cluster being formed in each scheme tends to increase in all four approaches; however the stability improvement is much high as far as our scheme is concerned.

## V. CONCLUSIONS & FUTURE WORK

In this paper we presented the design and performance of a new clustering scheme known as associativity-based clustering protocol. Our clustering scheme is different from similar approaches in that ours' is geographically-oriented. This way of associating dynamic clusters to geographic locations results in the following benefits: 1) This approach makes the task of mobility management easy, 2) We could predict a specific MN's future positions and packets can be forwarded continuously, 3) The identity of a cluster will not change in subsequent CH changeovers, 4) CH changeover is not frequent, and cluster set up time is minimal. We have demonstrated that this clustering scheme results in more stable clusters while incurring less control cost than those of other well-known schemes. In our future work we have determined to use our clustering scheme to realize a more scalable location-service [11]. In this location-service, we use our CH election heuristic to identify a dominating-set (CHs) that performs periodic location-updates on behalf of others. This way of enabling only a set of nodes to handle location-updates will minimizes inevitable superfluous flooding by every node, and hence leads to less signaling traffic when compared to that of other existing location-service schemes. We plan to report such findings in future papers.

## REFERENCES

[1] S.Sivavakeesar, and G.Pavlou, "Stable Clustering Through Mobility Prediction for Large-Scale Multihop Intelligent Ad Hoc Networks", IEEE WCNC, Mar. 2004.

[2] C-K.Toh, G.Lin, and M.Delwar, "Implementation and Evaluation of an Adaptive Routing Protocol for Infrastructureless Mobile Networks", *Proc. Int'l Conf. on* Computer Communications & Networks (IEEE IC3N), Las Vegas, 2000.

[3] A.D.Amis, R.Prakash, T.H.P.Vuong, and D.T.Huynh, "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks", *Proc.* IEEE INFOCOM'2000, vol. 1, Mar. 2000, pp. 32 – 41.

[4] S.H.Lee and D-H.Cho, "A new adaptive routing scheme based on the traffic characteristics in mobile ad hoc networks", IEEE Vehicular Technology Conference, vol. 6, 2000, pp. 2911 – 2915 .

[5] L.Blazevic, L.Buttyan, S.Capkun, S.Giordono, J-P.Hubaux and J-Y.Boudec, "Self-Organisation in Mobile Ad Hoc Networks: The Approach to Terminodes", IEEE Communications Magazine, June 2001, pp 166 –173.

[6] T-C.Hou, and T-J.Tsai, "An Access-Based Clustering Protocol for Multihop Wireless Ad Hoc Networks", IEEE Journal on Selected Areas in Communications, vol. 19, July 2001, pp 1201 –1210.

[7] J.Habetha, A.Hettich, J.Peetz, and Y.Du, "Central Controller Handover Procedure for ETSI_BRAN HiperLAN2 Ad Hoc Networks and Clustering with Quality of Service Guarantees", Mobile and Ad Hoc Networking and Computing (MobiHOC), 2000, pp. 131 – 132.

[8] B.McDonald, and F.Znati, "A Mobility-Based Framework for Adaptive Clustering in Wireless Ad Hoc Networks", IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, August 1999, pp 1466 –1487.

[9] P.Basu, N.Khan, and D.C.Little, "Mobility Based Metric for Clustering in Mobile Ad Hoc Networks", Workshop on Distributed Computing Systems, 2001, pp. 413 –418.

[10] M.Jiang, J.Li, and Y.C.Tay, "Cluster Based Routing Protocol (CBRP) Function Specifications", IETF Draft, August 1999.

[11] M.Mauve, J.Widmer, and H.Hartenstein, "A Survey on Position-Based Routing in Mobile Ad Hoc Networks", IEEE Network, vol. 15, no. 6, Nov./Dec. 2001, pp. 30 – 39.

[12] X.Zengu, R.Bagrodia, and M.Gerla, "GloMoSim: A Library for Parallel Simulations of Large-scale Wireless Networks", Proceedings of the 12th Workshop on Parallel and Distributed Simulations, May 1998.

[13] R.Ramanathan, and M.Steenstrup, "Hierarchically-organised, multihop mobile wireless networks for quality-of-service support", Mobile Networks and Applications, vol. 3, June 1998, pp. 101 – 119.

[14] X.Hong, K,Xu, and M.Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks", IEEE Network, vol. 16, no. 4, Jul./Aug. 2002, pp. 11 – 21.