Mobility-based d-Hop Clustering Algorithm for Mobile Ad Hoc Networks

Inn Inn ER[2,1] Winston K.G. Seah[1,2]
{stuerii, winston}@i2r.a-star.edu.sg

[1]Institute for Infocomm Research
Agency for Science Technology and Research
Singapore

[2]Department of Computer Science
School of Computing
National University of Singapore

Abstract- **This paper presents a mobility-based d-hop clustering algorithm (MobDHop), which forms variable-diameter clusters based on node mobility pattern in MANETs. We introduce a new metric to measure the variation of distance between nodes over time in order to estimate the relative mobility of two nodes. We also estimate the stability of clusters based on relative mobility of cluster members. Unlike other clustering algorithms, the diameter of clusters is not restricted to two hops. Instead, the diameter of clusters is flexible and determined by the stability of clusters. Nodes which have similar moving pattern are grouped into one cluster. The simulation results show that MobDHop has stable performance in randomly generated scenarios. It forms lesser clusters than Lowest-ID and MOBIC algorithm in the same scenario. In conclusion, MobDHop can be used to provide an underlying hierarchical routing structure to address the scalability of routing protocol in large MANETs.**

Keywords: cluster, mobility-based clustering, mobile ad hoc networks, MANET, mobility pattern.

1. Introduction

Mobile ad hoc network (MANET) consists of a number of wireless hosts that communicate with each other through multi-hop wireless links in the absence of fixed infrastructure. They can be formed and deformed spontaneously at anytime and anywhere. Some envisioned MANETs, such as mobile military networks or future commercial networks may be relatively large (e.g. hundreds or possibly thousands of nodes per autonomous system). The need to store complete routing details for an entire network topology raises scalability issue. The flat hierarchy adopted by most of the existing MANET routing protocols may not be able to support the routing function efficiently since their routing tables could grow to an immense size if each node had a complete view of the network topology. Therefore, clustering algorithms are proposed in MANETs to address scalability issue by providing a hierarchical network structure for routing.

Clustering algorithms can be performed dynamically to adapt to node mobility[2]. MANET is dynamically organized into groups called clusters to maintain a relatively stable effective topology [1]. By organizing nodes into clusters, topology information can be aggregated. This is because the number of nodes of a cluster is smaller then the number of nodes of the entire network. Each node only stores fraction of the total network routing information. Therefore, the number of routing entries and the exchanges of routing information between nodes are reduced[3]. Apart from making large networks seem smaller, clustering in MANETs also makes dynamic topology appear less dynamic by considering cluster stability when they form[2]. Based on this criterion, all cluster members that move in a similar pattern remain in the same cluster throughout the entire communication session. By doing this, the topology within a cluster is less dynamic. Hence, the corresponding network state information is less variable[3]. This minimizes link breakage and packet loss.

Clustering algorithm in MANETs should be able to maintain its cluster structure as stable as possible while the topology changes[1]. This is to avoid prohibitive overhead incurred during clusterhead changes. In this paper, we propose a mobility-based d-hop clustering algorithm (MobDHop) that forms *d*-hop clusters based on a mobility metric suggested by Basu *et al.*[8]. The formation of clusters is determined by the mobility pattern of nodes to ensure maximum cluster stability. We observe that mobile users in MANET may move in groups. This is known as group mobility[10]. Mobile hosts may be involved in team collaborations or activities. They may have a common mission (save victims that are trapped in collapsed building), perform similar tasks (gather information of threats in a battlefield) or move in the same direction (rescue team designated to move towards east side of disaster struck area). Therefore, our algorithm attempts to capture group mobility and uses this information to form more stable clusters.

MobDHop, a distributed algorithm, dynamically forms stable clusters which can serve as underlying routing architecture. First, MobDHop forms non-overlapping two-hop cluster like other clustering algorithms. Next, these clusters initiate a merging process among each other if they could listen to one another through gateways. The merging process will only be successful if the newly formed cluster achieves a required level of stability. As mentioned, most of the existing clustering algorithms form two-hop clusters which may not be too useful in very large MANETs. Therefore, MobDHop is designed to form *d*-hop clusters that are more flexible in cluster diameter. The diameter of clusters is adaptive to the mobility pattern of network nodes. MobDHop is simple and incurs as low overhead as possible. Information exchange during the formation of clusters, clusterhead changes and clusterhead handovers are kept to minimum. The remainder of this paper is organized as follows: We present an overview of clustering algorithms proposed for MANETs in Section 2. Next, details of MobDHop are presented in Section 3. Section 4 discusses our simulation results and analysis. Finally, we conclude in Section 5.

2. Related Work

A number of clustering algorithms have been proposed in literature such as Linked Cluster Algorithm (LCA)[4], Lowest-ID Algorithm (L-ID)[5], Maximum Connectivity Clustering (MCC)[6], Least Clusterhead Change Algorithm (LCC)[7], and MOBIC[8]. LCA[4] was developed for packet radio networks and intended to be used with small networks of less than 100 nodes. LCA organizes nodes into clusters on the basis of node proximity. Each cluster has a clusterhead, and all nodes within a cluster are within direct transmission range of the clusterhead. Gateways are nodes that are located in the overlapping region between clusters. Two clusters communicate with each other via gateways. Pair of nodes can act as gateways if there are no nodes in the overlapping region. LCA was later revised[5] to reduce the number of clusterheads. In the revised version of LCA, a node is said to be covered if it is in the 1-hop neighborhood of a node that has declared itself as clusterhead. A node declares itself to be a clusterhead if it has the lowest id among the non-covered nodes in its 1-hop neighborhood, known as Lowest-ID algorithm.

Parekh suggested MCC in which the clusterhead election is based on degree of connectivity instead of node id[6]. A node is elected as a clusterhead if it is the highest connected node in all of the uncovered neighboring nodes. This algorithm suffers from dynamic network topology, which triggers frequent changes of clusterheads. Frequent cluster reconfiguration and clusterhead reselection incur prohibitive overhead.

LCC[7] is designed to minimize clusterhead changes. A clusterhead change occurs when two clusterheads come within range of each other, or a node becomes disconnected from any cluster. When two clusterheads come into direct contact, one of the clusterheads will give up its role. Some of the nodes in one cluster may not be members of the other clusterhead's cluster. Therefore, one or more of those nodes must become a clusterhead. Such changes propagate across the network, causing a rippling effect of clusterhead changes.

Basu et al.[8] propose a weight-based clustering algorithm, MOBIC, which is similar to L-ID. Instead of node ID, MOBIC uses a new mobility metric, Aggregate Local Mobility (ALM), to elect a clusterhead. The ratio between the received power levels of successive transmissions between a pair of nodes is used to compute the relative mobility between neighboring nodes, which determines the ALM of each node.

All of the above algorithms create two-hop clusters in MANETs. They are more suitable for dense MANETs in which most of the nodes are within direct transmission range of clusterheads. However, these algorithms may form a large number of clusters in relatively large and sparse MANETs. Therefore, two-hop clusters may not be able to achieve effective topology aggregation. . Amis *et al.* generalized the clustering heuristics so that an ordinary node can be at most *d* hops away from its clusterhead[9]. This algorithm allows more control and flexibility in the determination of clusterhead density. However, clusters are formed heuristically without taking node mobility and their mobility pattern into consideration. McDonald and Znati[2] designed a (α,t)-clustering algorithm that adaptively changes its clustering criteria based on the current node mobility. This algorithm determines cluster membership according to a cluster's internal path availability between all cluster members over time.

3.Mobility-based d-hop Clustering Algorithm

A successful dynamic clustering algorithm should achieve a stable cluster topology with minimal communications overhead and computational complexity [2]. The efficiency of the algorithm is also measured by the number of clusters formed [11]. Therefore, the main design goals of our clustering algorithm are as follows:

1. The algorithm minimizes the number of clusters by considering group mobility pattern.
2. The algorithm must be distributed and executed asynchronously.
3. The algorithm must incur minimal clustering overhead, be it cluster formation or maintenance overhead.
4. Network-wide flooding must be avoided.
5. Optimal clustering may not be achieved, but the algorithm must be able to form stable clusters should any exists.

Before introducing **MobDHop**, we first make a few assumptions on the network:

1. Two nodes are connected by bi-directional link (symmetric transmission).
2. The network is not partitioned.
3. Each node can measure its received signal strength.

Through periodic beaconing or hello messages used in some routing protocols, a mobile node can estimate its distance to its neighbor based on the measured received signal strength from that particular neighbor. In the Friss transmission equation, the received power over a point-to-point radio link is given by:

$$P_r = P_t * G_t * G_r * \frac{\lambda^2}{(4 * \pi * d)^2}$$

where P_r = received power, P_t = transmitted power, G_t = antenna gain of the transmitter, G_r = antenna gain of the receiver, λ = wavelength (c/f), and d = distance.

This shows the familiar inverse square-law dependence of received power with distance, i.e. $P_r \, \alpha \, 1/d^2$. Therefore, we derive the estimated distance between two nodes from the above equation based on received signal strength. In real world scenario, it may not be possible to obtain an exact calculation of the physical distance between two nodes from the measured signal strength. However, MobDHop does not depend on accurate estimation of distances between two nodes to operate correctly. Instead, we observe the variation of the estimated distances (in other words, fluctuation of the received signal strength) between two nodes over time. From the series of distance variations, we use statistical testing to predict relative mobility pattern between two nodes. We intuitively conclude that two nodes are stably-connected if the received signal strength between them varies negligibly over time. If two nodes are moving together at a similar speed towards the same direction, the variation of their received signal strength should be very small. This serves as one of the metrics we used to group the nodes into its respective cluster.

Based on the above justification, we will not use complex calculation in MobDHop in order to obtain accurate physical distance. Instead we use the received signal strength measured at the arrival of every packet to estimate the distance from one node to its neighbor node. The stronger the received signal strength, the closer the neighbor node. It is important to know that the "closeness" between two nodes is not necessarily measured by their absolute or physical distance. For example, node A may be very close to node B. However, it runs out of energy and transmits packets at lower power. In this case, it behaves like a distanced node from node A. Therefore, absolute distance may not be useful in predicting link stability in this case.

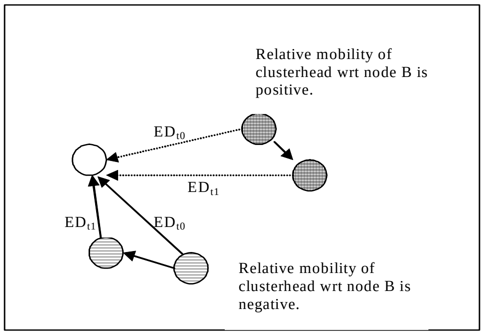

Figure 1. Relative Mobility

Measured signal strength of successive packets is used to estimate the relative mobility between two nodes. We calculate the difference of estimated distance from a neighboring node at two successive time moments. The difference indicates the pair-wise relative mobility as shown in Figure 1. If the new distance is larger than the old distance, the neighboring node is moving away from the measuring node. We group the nodes into two-hop clusters based on their relative mobility in the first stage. Next, we expand the cluster by merging individual nodes with two-hop clusters or merging two or more two-hop clusters based on the previously described metric, i.e. the variation of estimated distance between gateway nodes. Before introducing MobDHop, we give a brief introduction to different terms and metrics used in MobDHop.

3.1 Preliminary Concepts

A node may become a **clusterhead** if it is found to be the most stable node among its neighborhood. Otherwise, it is an **ordinary member** of at most one cluster. When all nodes first enter the network, they are in **non-clustered** state. A node that is able to listen to transmissions from another node which is in different cluster is known as a **gateway**. We formally define the following terms: (1) estimated distance between nodes, (2) relative mobility between nodes, (3) stably-connected node pair, (4) local stability, and (5) estimated mean distance.

Definition 1: Estimated distance between node A and B, $E[D_{AB}]$, is calculated as below.Please note that this formula is not aimed to obtain exact physical distance between two nodes. Instead, it is an approximation to show the "closeness" of two nodes.

$$E[D_{AB}] = \frac{k}{\sqrt{P_r}}, \text{ where k is a constant}$$

Definition 2: Relative mobility between nodes A and B, M_{AB}^{rel}, indicates whether they are moving away from each other, moving closer to each other or maintain the same distance from each other. To calculate relative mobility, we compute the difference of the distance at time, t and the distance at time, $t - 1$. Relative mobility at node A with respect to node B at t is calculated as follows:

$$M_{AB}^{rel} = E[D_{AB}^{t}] - E[D_{AB}^{t-1}]$$

Definition 3: The variation of $E[D_{AB}]$ over a time period, T, VD_{AB}, is defined as the changes of estimated distances between node A and B over a predefine time period. Let's consider node A as measuring node. Node A has a series of estimated distance values from node B measured at certain time interval for n times, $E[D_{AB}]=\{E[D_{AB}]_t,\ t = 0, 1, 2, \dots, n\}$. Therefore we calculate VD_{AB} as the standard deviation of distance variation as follows:

$$VD_{AB} = \sigma(|\, E[D_{AB}]_1 - E[D_{AB}]_0 \,|, \\ |\, E[D_{AB}]_2 - E[D_{AB}]_0 \,|, \dots, |\, E[D_{AB}]_n - E[D_{AB}]_0 \,|)$$

Definition 4: Local stability at node A, St_A, represents the degree of stability at node A with respect to all its neighbors. Local stability is the standard deviation of relative mobility values of all neighbors. Therefore it is calculated as follows:

$$St_A = \sigma\,(VD_{AB_1}, VD_{AB_2}, \ldots, VD_{AB_m})$$

Definition 5: Estimated Mean Distance (EMD) for cluster, C_1, indicates the mean distance from each neighbor to the clusterhead, CH_1 of cluster, C_1. The EMD is calculated as follows:

$$EMD_{C_1} = \mu\;(E[D_{CH_1N_1}], E[D_{CH_1N_2}], \ldots, E[D_{CH_1N_m}])$$

3.2 Discovery Stage

This is an initial setup stage for two-hop clusters when the network is first initialized. All nodes periodically broadcast Hello messages, including their local stability value (initialized to infinity at the beginning of operation). Each node measures the received signal strength of every received Hello message and estimates the distance with each neighbor. After receiving at least two successive Hello messages, each node calculates relative mobility with its neighbor at time t using equation in **Definition 3**. After a discovery period, T_D, each node assumes that it has the complete knowledge of its neighborhood. Then it computes its local stability value using equation in **Definition 4**. Then, it broadcasts Hello messages with the computed local stability value. Thus, each node knows the local stability of their neighbors. After an assignment period, T_A, each node compares its own local stability value with those of its neighbors. If a node has the lowest value of local stability among all its neighbors, it assumes the status of a clusterhead. Its local stability value becomes group stability (GS).

Then, the clusterhead computes EMD with respect to all cluster members (one-hop neighbors of clusterhead). The *EMD* is computed to capture another characteristic of the network if the nodes are moving in groups. This characteristic is suggested by Reference Point Group Mobility (RPGM) model[10]. The RPGM model suggested that a group center is used to characterize the movement of its corresponding group members, including their direction, speed, and distance from group center. This is similar to the real life group communication in which group leader guides the movement of its group members. Therefore, group members will not move too far away from the group leader. Their movement area is usually bounded. EMD is used as one of the metrics in the merging process to allow a new cluster member to join the cluster.

If a cluster member is able to hear hello messages from another node from another cluster, it assumes the role of a gateway. Otherwise, it declares itself to be a cluster member. If two neighboring nodes in non-clustered state have the same value of local stability, the clusterhead assignment is deferred for a back-off period. The local stability will be recomputed at the end of back-off period. This is to ensure the clusterhead is the most stable node among its neighborhood. Hence, it has the greatest potential to be a real group leader in real life scenarios.

3.3 Merging Stage

After the discovery stage, all nodes are covered by two-hop clusters. There are two cases that may initiate a merging process:

a) A non-clustered node requests to join the neighboring clusters.
b) Two neighboring gateways request to merge their clusters.

In the first case, a non-clustered node initiates the merging process. In the second case, two neighboring gateway nodes, G_1 and G_2 from C_1 and C_2 respectively, which are in transmission range of each other, initiate the merging process. Nodes initiating merging process start collecting samples for estimated distance between them. From the samples of estimated distances, they compute mean of estimated distance, $E[D_{G1G2}]$, and variation of distance over time, VD_{G1G2}. Apart from this, they also calculate their relative mobility with respect to each other. To be successfully merged, both gateway nodes must fulfill the following two criteria at the end of sampling period, T_S:

1) $VD_{G1G2} \leq min\{St_{C1}, St_{C2}\}$, and
2) $\mu(E[D_{G1G2}]) \leq max\{EMD_{C1}, EMD_{C2}\}$

The first criterion ensures that the variation of estimated distance between two merging nodes is less than or equal to the minimum value of group stability among two clusters. This indicates that the link between two nodes is at least as stable as other links in one of the clusters which is more stable. The second criterion tells us that the distance between two nodes conforms to the distance characteristic of the larger cluster. Therefore both clusters have higher probability to be originated from the same group of real life situation as suggested in RPGM. In most of the group communication applications, members belong to the same group tend to remain in each other transmission ranges over time by maintaining a constant distance from group leader.

3.4 Maintaining Stage

We first consider two cases that may cause topology changes in MANET and thus invoke cluster maintenance stage:

1) A node switches on and joins the network.
2) A node switches off and leaves the network.

When a node switches on, it will initiate the merging process in the same manner as described in Section 3.3. It checks all the links with its neighboring nodes and collects samples for estimated distance from each neighbor. Then it computes the variation of distance over time, *VD*, with each neighbor. At the end of sampling period, it chooses the neighbor with lowest *VD*, and joins its cluster.

When a node switches off and the node is a clusterhead, this will cause its cluster members to lose the clusterhead and fail to receive cluster advertisements for a predefined period. The immediate neighbors of the clusterhead will initiate the discovery process as described in Section 3.2 in which a new clusterhead will be elected. The information of the new clusterhead will then be propagated to other cluster members, which are further away from it. However, during the clusterhead election period, other cluster members which are at least 2 hops away from the old clusterhead may detect the loss of clusterhead and decide to join neighboring cluster if the merging criteria specified in Section 3.3 can be met. If a node found itself in non-clustered state, it will initiate merging with neighboring clusters whenever possible. Otherwise, it will declare itself to be a clusterhead of a one-node cluster. From time to time, it will try to merge with other clusters if possible.

4. Simulation Results and Discussions

The performance of MobDHop is evaluated via simulations using NS-2 with CMU wireless extensions [12]. The scenarios were generated with input parameters as listed in Table 1, such as network size, speed, transmission range, broadcast interval, clusterhead contention interval and simulation time. The movement of mobile nodes is randomly generated and continuous within the whole simulation period. We implemented MobDHop as described in Section 3. The local stability value, group stability value, node status, node clusterhead id, and cluster EMD are added into “Hello” messages. “Hello” messages have been widely used in on-demand routing protocols to maintain neighbor connectivity. Each node broadcasts “Hello” messages at certain broadcast interval to tell the neighbors of its existence. MobDHop does not use additional control packets for information exchange to form or maintain clusters.

Figure 2 and 3 show the performance of MobDHop for MANETs which are different in number of nodes and transmission ranges. The mobile nodes are moving continuously at 20m/sec throughout the entire network simulation period (300 seconds). We note that the average number of clusters is relatively high when the transmission range is small (10 - 20 m). For small ranges, most nodes tend to be out of each other’s transmission range and the network may become disconnected. Therefore, most nodes form one-node cluster, which only consists of itself. Due to our algorithm design, which require one-node clusters to attempt to merge with neighboring clusters whenever possible, clusterhead will switch their status to non-clustered state in order to merge with their neighbors (if any). This causes the high rate of clusterhead changes in disconnected networks. However, we argue that this will not affect network performance as this will only occur when the network is disconnected (A disconnected network is unable to function too).

When transmission range increases, more nodes can hear each other. The average number of clusters formed decreases and the clusters become larger in size. Since the transmission range is large, mobile nodes tend to remain in the range of their neighbors. Therefore, clusters are less dynamic and the number of clusterheads changes also decreases.

We also compare the performance of MobDHop with the Lowest-ID algorithm and MOBIC in a 50-node MANET under constant mobility (20m/sec). In Figure 4, we note that there is a small difference between Lowest-ID and MOBIC with respect to the average number of clusters formed. This is because both algorithms are variations of a local weight based clustering technique that forms two-hop clusters. MobDHop forms less clusters in the similar scenario since it forms variable-diameter clusters based on node mobility pattern. This is one of the desirable properties in clustering algorithm especially when the scalability is the main concern.

Table 1. Simulation Parameters for MobDHop

Parameter	**Meaning**	**Value in Our Simulation**
N	Number of Nodes	25, 50, 75, 100
m x n	Network Size	500 m^2
MaxSpeed	Maximum Speed of node movement	20 m/sec
Tx	Transmission Range	10 m – 125 m
PT	Pause Time	0 sec
BI	Broadcast Interval	0.75-1.25 sec
TD	Discovery Interval	BI * 10
TA	Assignment Interval	BI * 2
TM	Merge Interval	BI * 5
TC	Contention Period	BI * 2
S	Simulation Time	300 sec

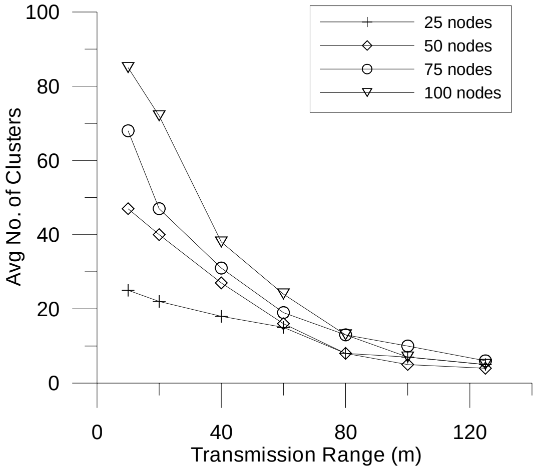

Figure 2. Average number of clusters

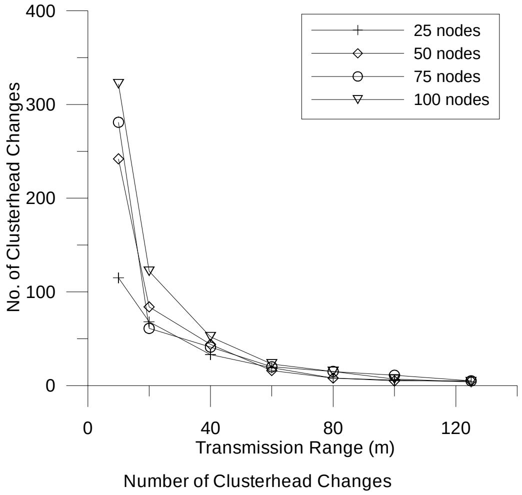

Figure 3. Number of clusterhead changes

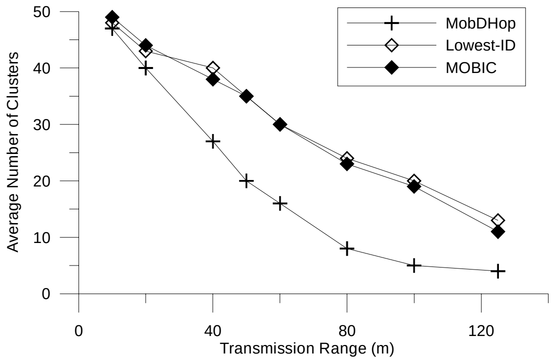

Figure 4. Comparisons between different clustering algorithms in a 50-node MANET.

5. Conclusions

Clustering can provide large-scale MANETs with a hierarchical network structure to facilitate routing operations. In this paper, we proposed a distributed clustering algorithm which forms variable-diameter clusters that may change its diameter adaptively with respect to mobile nodes' moving patterns. Inspired by Basu et. al[8], we proposed two mobility metrics based on the relative mobility concept: (1) variation of estimated distance between nodes over time and (2) estimated mean distance for cluster, in order to measure the stability of a cluster. These metrics are used to decide cluster memberships. Therefore, the formation of clusters in MobDHop is determined by the mobility pattern of nodes to ensure maximum cluster stability.

To achieve the desired scalability, MobDHop forms variable-diameter clusters, which allows cluster members to be more than two hops away from their clusterhead. The diameter of clusters is dependent on the mobility behavior of nodes in the same cluster. As long as the nodes are moving towards the same direction in a stable behavior, they can be grouped into the same cluster. This is justified by the assumption of group movement, in which members of a group tend to move towards a similar destination in real-life scenarios.

We have simulated MobDHop and presented some preliminary results in Section 4. In conclusion, the performance of MobDHop is comparable to other existing algorithms. It also creates lesser and more stable clusters in order to achieve high scalability. The clusterhead change is relatively low. However, we will perform extensive simulation-based comparisons between existing clustering algorithms and MobDHop to evaluate different aspects of performance such as cluster stability, overhead consumption, latency and others. We may use other mobility models which are more realistic such as RPGM in our simulations. Finally, designing a multicast routing protocol which can work on-top of MobDHop in order to address scalability issues in MANET is part of our ongoing research.

References:

[1] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 15(7):1265-1275, Sept. 1997.

[2] A. B. McDonald and T. F. Znati. A mobility-based framework for adaptive clustering in wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1466-1486, Aug. 1999.

[3] C. E. Perkins, editor. Ad Hoc Networking. Addison-Wesley, 2001.

[4] D. J. Baker and A. Ephremides. The architectural organization of a mobile radio network via a distributed algorithm. *IEEE Transactions on Communications*, 29(11):1694-1701, 1981.

[5] A. Ephremides, J. Wieselthier, and D. Baker. A design concept for reliable mobile radio network with frequency hopping signaling. In *Proceedings of IEEE 75*, pages 56-73, 1987.

[6] A. K. Parekh. Selecting routers in ad hoc wireless networks. In *ITS*, 1994.

[7] C.-C. Chiang, H.-K. Wu, W. Liu, and M. Gerla. Routing in clustered multihop, mobile wireless networks with fading channel. *IEEE Singapore International Conference on Networks (SICON)*, pages 197-211, Apr. 1997.

[8] P. Basu, N. Khan, and T. D. C. Little. Mobility based metric for clustering in mobile ad hoc networks. *Workshop on Distributed Computing Systems*, pages 413-418, 2001.

[9] A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh. Max-min d-cluster formation in wireless ad hoc networks. In *Proceedings of IEEE INFOCOM '00*, Vol. 1, pages 32-41, Mar. 2000.

[10] X. Hong, M. Gerla, G. Pei, and C. Chiang. A group mobility model for ad hoc wireless networks. In *Proceedings of ACM/IEEE MSWiM*, Seattle, WA, Aug.1999.

[11] F. G. Nocetti, J. S. Gonzalez, I. Stojmenovic, "Connectivity based k-hop clustering in wireless networks," *Telecommunication Systems* 22 (2003) 1-4, 205-220, 2003.

[12] K. Fall, and K. Varadhan, "The ns Manual," http://www.isi.edu/nsnam/ns/, 2002.