Clustering Overhead and Convergence Time Analysis of the Mobility-based Multi-Hop Clustering Algorithm for Mobile Ad Hoc Networks

Inn Inn Er[1,2], Winston K.G. Seah[2,1]

[1] Department of Computer Science, School of Computing,
National University of Singapore, Singapore 117543
{erinninn, wseah}@comp.nus.edu.sg
[2] Institute for Infocomm Research,
21, Heng Mui Keng Terrace, Singapore 119613.
{winston, stuerii}@i2r.a-star.edu.sg

Abstract. With the emergence of large mobile ad hoc networks, the ability of existing routing protocols to scale well and function satisfactorily comes into question. Clustering has been proposed as a means to divide large networks into groups of suitably smaller sizes such that prevailing MANET routing protocols can be applied. However, the benefits of clustering come at a cost. Clusters take time to form and the clustering algorithms also introduce additional control messages that contend with data traffic for the wireless bandwidth. In this paper, we aim to analyze a distributed multi-hop clustering algorithm, Mobility-based D-Hop (MobDHop), based on two key clustering performance metrics and compare it with other popular clustering algorithms used in MANETs. We show that the overhead incurred by multi-hop clustering has a similar asymptotic bound as 1-hop clustering while being able to reap the benefits of multi-hop clusters. Simulation results are presented to verify our analysis.

Keywords. MobDHop, clustering algorithm, mobile ad hoc network, time complexity, message complexity.

1 Introduction

Mobile ad hoc networks (MANETs) consist of a number of mobile hosts that communicate with each other via multi-hop wireless links in the absence of fixed infrastructure. The mobility of mobile hosts and the transient nature of wireless links result in a rapidly changing network topology, making the network management, routing, multicasting, and various networking related tasks more sophisticated than those in static wired networks. Some envisioned MANETs, such as mobile military networks or future commercial networks may be relatively large (e.g. hundreds or thousands of nodes per autonomous system) making scalability one of the key issues in MANET research efforts. For example, the flat hierarchy adopted by most of the existing MANET routing protocols may not be able to support the routing function efficiently because their routing tables could grow to an immense size if each node had a complete view of the network topology. Therefore, clustering is proposed to organize mobile nodes into small groups, so called "clusters", in order to provide a hierarchical network structure for efficient routing. Apart from routing, clustering may be used to facilitate the implementation of spatial reuse, location management, network management, and quality of service (QoS) support.

Organizing a MANET into stable clusters is critical to avoid the prohibitive overhead incurred during clusterhead changes [1]. There are some techniques suggested to reduce clusterhead changes, e.g. the Least Clusterhead Change algorithm [2] when applied to the Lowest ID clustering algorithm suggests that a clusterhead change will not occur if another host with lower node id comes into the direct transmission range of the existing clusterhead. However, node mobility of MANETs may still cause frequent failure and activation of links. As a result, clustering algorithms for MANETs are designed to be adaptive towards node mobility, e.g. (α,t)-Cluster Protocol [3]. In the (α,t)-Cluster Protocol, only neighbouring nodes that fulfil a certain probability of path availability bound will be clustered. Therefore, clustering is more dominant in networks with low rates of mobility while the clustering may not be done in networks with high rates of mobility. This scheme is able to make a dynamic topology appear less dynamic.

Another simple, yet promising distributed algorithm, named Mobility-based D-Hop (MobDHop) clustering algorithm that forms *d*-hop clusters, was proposed in [4] based on relative mobility concept suggested by Basu *et al*. [5]. The formation of clusters is determined by the mobility pattern of nodes to ensure maximum cluster stability. This is motivated by the fact that most targeted MANET applications are based on collaborative computing where mobile hosts may be involved in team collaborations. 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). The users of such a MANET are very likely to move in groups. MobDHop captures group mobility and uses this information to form more stable clusters. We choose MobDHop as the basis of our study, comparing it with other similar clustering algorithms.

The benefits of clustering comes at a cost in terms of the time it takes for clusters to form and also the overheads incurred in the additional network traffic generated. This paper aims to provide insights on the theoretical analysis of five clustering algorithms, viz. Lowest-ID [6, 7], Maximum Connectivity Clustering (MCC) [8], Distributed Mobility-Adaptive Clustering (DMAC) [9], Max-Min D-Cluster [10] and MobDHop. We analyze the clustering algorithms with respect to the convergence time and message complexity. These criteria reflect the performance of the algorithm in terms of signalling traffic and its capability to react to topology changes quickly and efficiently. The approach used is inspired by the theoretical analysis of DMAC [11]. The remainder of this paper is organized as follows: The basic operation of MobDHop clustering algorithm is briefly discussed in Section 2. Next, the analysis of MobDHop is discussed in Section 3. Section 4 presents the simulation results and discussion. Section 5 highlights the overhead and convergence time analysis on the Lowest-ID and MCC clustering algorithms. A summary of the previous analysis on DMAC and Max-Min D-Cluster is also presented here. Then a comparison among these algorithms is made. Finally, Section 6 concludes the paper.

2 Mobility-Based *d*-Hop Clustering Algorithm

Mobility-based d-Hop (MobDHop) [4] clustering algorithm dynamically forms stable clusters. MobDHop is designed to form *d*-hop clusters that are more flexible in cluster diameter. Each cluster's members are at most *d*-hops away from its clusterhead, where *d* is a user parameter, which can be adjusted to meet the required clusterhead density. The diameter of clusters, however, is not necessarily 2*d*. Instead, it is adaptive to the group characteristics and mobility pattern of network nodes.

First, MobDHop forms non-overlapping one-hop clusters (each cluster member is at most one hop away from its clusterhead). The election of clusterhead is based on two mobility metrics: (a) variation of estimated distance between nodes over time (VD), and (b) estimated mean distance for cluster (EMD). When the network is first initialized, all nodes periodically broadcast Hello messages. Each node measures the received signal strength of every received Hello message and calculates the Estimated Distance (ED) with respect to each neighbour. After receiving a pre-specified number of Hello messages, each node computes the VD with respect to every neighbour. Based on this information, each node computes a local variability value, i.e. the mean of VD of all neighbours, which implies how stable a node is with respect to all immediate neighbours. If a node has the lowest variability value, i.e. it is the most stable node among its neighbourhood, it assumes the role of clusterhead and announces it with a Hello message; neighbour nodes assume the role of ordinary members upon receiving the announcement from clusterhead. If a cluster member can hear Hello messages from more than one cluster, it assumes the role of a gateway.

Next, a merging process will be initiated by a non-clustered node to join the neighbouring cluster. A node may become non-clustered when it is newly activated or it loses its clusterhead due to node mobility. The merging node will first observe its neighbourhood and choose the neighbour to which it is most stably connected to. Then, it will try to merge into its neighbour's cluster if the following conditions are satisfied:

- Hop count from merging node to its new clusterhead is less than the restricting parameter, d.
- The variability value of the link between the merging node and its chosen neighbour should be lower than the overall variability value of the cluster. Therefore, the link can be added to the cluster without jeopardizing the overall cluster stability.

The second condition ensures that the newly formed cluster achieves a required level of stability by taking their VD and EMD into consideration. After the merging process, a valid cluster structure should be achieved. Such a valid condition is defined by the following properties: (1) every ordinary or gateway node has at least one clusterhead as its d-hop neighbour and (2) clusterheads cannot be direct neighbours of each other.

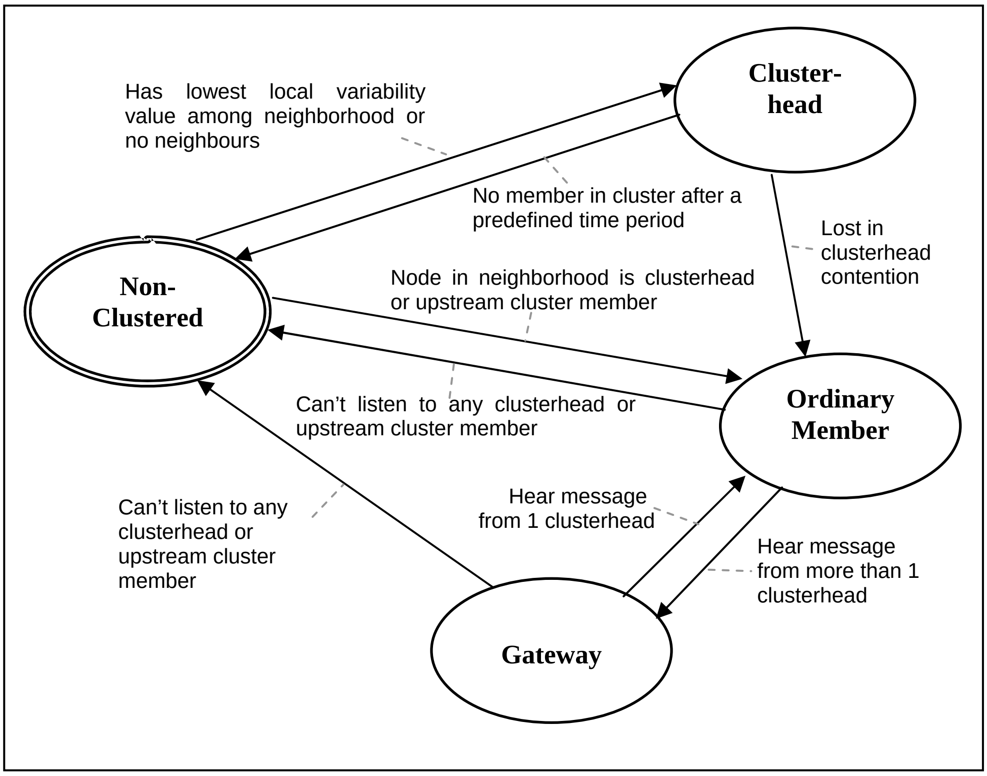

Fig. 1. Node state transition diagram for MobDHop

As a member node moves around, it decides which cluster it currently belongs to and what role it currently plays based solely on the local information. Each node reacts to the changes in the surrounding topology and changes its status or cluster membership accordingly. When the link of an ordinary node to its parent node fails, the ordinary node will first try to merge into neighbouring clusters after ensuring that the stability can be preserved. If this fails, it will determine its new role in the same way as it does during initialization phase. Its children nodes will be notified of the clusterhead loss and react similarly. A stable and valid cluster structure can be re-established after a certain convergence period. If a node finds itself in a non-clustered state, it will attempt to merge with neighbouring clusters. Otherwise, it will declare itself to be a clusterhead of a one-node cluster, and periodically tries to merge with a neighbouring cluster. Fig. 1 shows the node state transition for MobDHop.

3 Analysis of MobDHop Clustering Algorithm

The obvious drawback introduced by almost all clustering algorithms is the additional signalling overhead incurred in order to maintain the cluster structure. Before implementing cluster architecture in a MANET, we must ensure that the benefits from clustering could outweigh the costs. Therefore, it is essential to investigate the amount of signalling overhead. The clustering overhead, OH_C can be classified as follows:

1. Overhead due to Hello Protocol (OH_H)
2. Overhead due to Cluster Formation (OH_{CF})
3. Overhead due to Cluster maintenance (OH_{CM})

Hence, the total clustering overhead is the sum of the above contributing factors.

In most simulation evaluations, the assumption that continuous time is divided into discrete steps is made. Therefore, it is easier to measure duration taken by an algorithm to re-establish a valid cluster structure after a change in the network topology. This is called the convergence time or time complexity and defined as the number of time steps from a topology change until a valid cluster structure is re-established.

The goal of this paper is to evaluate the MobDHop clustering algorithm (as an example of a multi-hop clustering algorithm) with respect to its clustering overhead and convergence time. The following claim is made regarding the average clustering overhead incurred by MobDHop per node per time step:

Claim 1. OH_C = O(1) packet transmissions per node per time step.

3.1 Assumptions and Definitions

A MANET is represented by a connected, undirected graph, $G = (V, E)$, where V is the set of nodes and E is the set of bidirectional links. We assume that nodes are located randomly throughout the network area. To simplify the analysis of link change frequency, the random waypoint mobility model [12], with zero pause time and positive minimum speed is assumed. Two nodes are neighbours if their Euclidean distance between each other is less than their transmission radius, R. The following definitions are useful in the following analysis:

- A = Area of a square (a x a).
- N = the number of nodes in the network.
- r_{hello} = the number of hello messages emitted by a node per time step (hello rate).
- r_{link} = the average number of link state change events occurred per time step.
- μ = average node speed.
- h_i = hop count from clusterhead of node, i.
- m = the average number of cluster members.
- H_{max} = maximum hop count from clusterhead (one of MobDHop parameters that can be adjusted accordingly).
- T_{sample} = the number of time steps taken by a node to collect stability information from neighbours.
- T = the number of time steps taken by the algorithm after a change in the topology to accomplish cluster reorganization (Time Complexity).
- M = the number of messages exchanged between nodes after a change in the topology to accomplish cluster reorganization.

Next, we define the notions of upstream member, downstream member, and peer member. Given node j and node k are members of the same cluster, node j is upstream member of node k if $h_j < h_k$; node j is downstream member of node k if $h_j > h_k$; node j is peer member of node k if $h_j = h_k$.

3.2 Hello Overhead Protocol

The hello messages are broadcasted for every predefined Hello interval during the communication session for nodes to learn its neighbourhood and corresponding stability information in order to compute local variability value which will be used in clusterhead election. Therefore, each node emits a certain amount of Hello messages per time step in order to maintain up-to-date neighbourhood knowledge. This incurs an overhead of $r_{hello}N$ packets per time step. Since r_{hello} is a constant predefined by the protocol and the communication session consists of D time steps, OH_{H} is O(DN).

3.3 Cluster Formation Overhead and Time Complexity

In the first phase of MobDHop (i.e. one-hop cluster formation), each node will first measure relative mobility with respect to all neighbours for a predefined sampling period, T_{sample}. Hence, the number of time steps each node takes before it can decide to be a clusterhead or to join a neighbouring cluster is at least T_{sample}. After the clustering decision is made, each node will broadcast a new Hello message with its latest cluster decision. If a node opts to be a clusterhead, it will broadcast a Hello message to its neighbours that contain its cluster ID and group variability. This is a trivial case and takes only 1 time step. On the contrary, if a node opts to join a neighbouring cluster, it will broadcast its decision to its clusterhead which is at most one hop away from the node. Therefore, this message takes at most 1 time step to reach the clusterhead. In short, time complexity of cluster formation in MobDHop is $T \leq T_{sample} + 1$. Message Complexity, on the other hand is $M = 1$. Since the cluster formation process will only occur once during the cluster setup phase, the cluster formation overhead, $OH_{\mathrm{CF}} = \mathrm{O}(T_{sample}N)$.

3.4 Cluster Maintenance Overhead and Time Complexity

Cluster maintenance in MobDHop is done by continuous inspection on local information via periodical messaging. The approach used to analyze overhead required by cluster maintenance process is greatly inspired by the analysis of DMAC done by Bettstetter and Krausser. In MobDHop, if a topology change is detected, the node will take respective action to maintain the cluster structure. There are three types of events that may cause a topology change in MANETs:

(i) A node joins the network.

(ii) Two nodes move away from each other transmission range (link failure).

(iii) Two nodes move into each other transmission range (link establishment).

Joining of a New Node. After a new node joins the network, it has to make clustering decision, i.e. to decide which cluster to join and what role to play. This process is determined by two factors:

(i) The state of nodes in its neighbourhood.

(ii) Relative mobility with respect to every neighbour.

In MobDHop, a new node, say node a, will first try to merge into neighbouring clusters by measuring its relative mobility with respect to each neighbour for T_{sample} time steps and compute their variation of estimated distance over time. It will choose the neighbour which is relatively most stable, i.e. it yields lowest variation of estimated distance with respect to that neighbour. Denote the neighbour as node *b*. If node *b* is connected to its clusterhead by an unsaturated link (i.e. link which may consist of multiple hops but the hop count is less than H_{max} hops), node *a* joins the cluster successfully. If this condition fails, node *a* will decide its role (clusterhead or ordinary node) by taking all non-clustered neighbours into consideration during a clusterhead election as in cluster setup phase. Therefore, the message and time complexity depend on the configuration of neighbourhood at the time when the topology change occurs.

We denote the number of neighbours of a node, i.e. its degree, as *d*. We identified four kinds of neighbours:

(i) Neighbours that are clusterhead (d_{ch}).

(ii) Neighbours that have joined a cluster and are connected to their clusterhead by an unsaturated link ($d_{us\text{-}mem}$).

(iii) Neighbours that have joined a cluster and are connected to their clusterhead by a saturated link ($d_{s\text{-}mem}$).

(iv) Neighbours that are still not clustered to any cluster (d_{nc}).

Therefore, the total number of neighbours of a node, $deg = d_{ch} + d_{us\text{-}mem} + d_{s\text{-}mem} + d_{nc}$. If a new node, *i* has no neighbours ($d = 0$), a trivial case occurs. It selects itself as clusterhead and broadcasts its decision in the next Hello message ($M = 1$). This process is done in one time step ($T = 1$).

If node *i* has at least one neighbour that is clusterhead, or cluster member that is having an unsaturated link ($d_{ch} + d_{us\text{-}mem} > 0$), node *i* will start to collect information for local variability computation and decide its cluster membership after T_{sample} time steps. After making the decision, it will propagate this decision to its new clusterhead. The time needed for this decision to arrive is at most H_{max} time steps since the clusterhead is at most H_{max} away. Therefore, $T \leq T_{sample} + H_{max}$ and $M \leq H_{max}$.

Table 1. Time and message complexities due to different neighbourhood configuration

Neighbourhood Configuration				**Complexities**		**New Status After Topology Change**
d_{ch}	$d_{us\text{-}mem}$	$d_{s\text{-}mem}$	d_{nc}	**Time**	**Message**	
0	0	0	0	1	1	Clusterhead
> 0	0	Any	Any	$\leq T_{sample} + H_{max}$	1	Ordinary Node
0	> 0	Any	Any	$\leq T_{sample} + H_{max}$	$\leq H_{max}$	Ordinary Node
0	0	0	> 0	$\leq T_{sample} + H_{max}$	1	Clusterhead/ Ordinary Node

In the third case where all neighbours nodes are not yet clustered ($d_{nc} > 0$), node i will perform the similar process as in the cluster formation phase. Therefore, the time and message complexity is the same as those in cluster formation. Since MobDHop adopts Least Clusterhead Change (LCC) [2] mechanism during cluster maintenance, chain reaction caused by any cluster reorganization can be avoided. The time and message complexities for different kinds of neighbourhood configurations are summarized in Table 1.

Link Failure. When two neighbours move away from each other and eventually out of each other's transmission range, a link failure occurs. When a link failure is detected, both affected nodes react to this topology change accordingly.

A link failure between nodes from different clusters or between any two ordinary peer member nodes will not cause any cluster reorganization in MobDHop. Only link failure between an ordinary node and its clusterhead or its upstream ordinary node will trigger the cluster reorganization. In both cases, only the downstream member node will react to this topology change because the clusterhead or upstream ordinary node will simply eliminate downstream member node from their member lists.

We denote the reacting downstream member node as node a. First, we consider the base case when node a is a border node, i.e. it has no downstream members. Three similar cases may happen as in the case where a new node is added into the network. Therefore, the time and message complexity are the same as in Table 1.

In another case when the reacting node has downstream members, each downstream member has to react when they receive messages from their upstream member about status or cluster membership changes. Therefore, this is a chain reaction, which will be reaching an end when the effect reaches the border node of the cluster where the above mentioned base case is executed. Then, cluster reorganization is complete and a valid cluster structure is re-established. In other words, the chain reaction can at most happen at H_{max} - h_a + 1 nodes. Therefore, time complexity and message complexity is upper bounded as shown below:

- $T \leq (T_{sample} + H_{max}) + (H_{max} - h_a + 1)$
- $M \leq H_{max}$ x $(H_{max} - h_a + 1)$

Link Establishment. When two nodes move into each other's transmission range, a new link is established between them. A topology change occurs. When a new link is found, both affected nodes react to this event according to their current state.

A link establishment between two ordinary nodes will not cause any cluster reorganization because both nodes are still connected to their clusterheads. In the case of a new link established between an ordinary node and a clusterhead, no cluster reorganization shall take place because the cluster structure is still valid.

If a new link is established between two clusterheads, clusterhead contention occurs and MobDHop will resolve this contention by making the clusterhead which is more unstable to give up its role and join the other cluster as an ordinary node. If the loser has no members at all, the cluster reorganization is complete. Therefore, the loser node broadcasts its decision in the next Hello message (M = 1) and the process takes one time step (T = 1). If the loser node has cluster members, all members are subject to cluster reorganization. A similar process as in the link failure case will be executed. The base case occurs when the reacting node a is a border node. Three possible cases could happen as in previous section. Therefore, the time and message complexity for the base case are the same as shown in Table 1.

If the reacting node has downstream members, each downstream member has to react when they receive the clusterhead loss notification via Hello messages, indicating clusterhead or status changes. Again, this is a chain reaction that will come to an end when the effect reaches the border node of the loser cluster. Since each member is at most H_{max} hops away from its clusterhead, chain reaction will at most extend to H_{max} hops and may involve all cluster members. In short, the upper bounds of message and time complexity after a link establishment event are listed as below:

- $T \leq (T_{sample} + H_{max}) + (H_{max})$
- $M \leq mH_{max}$

Total Cluster Maintenance Overhead. As analysed above, the upper bound of message complexity is $M = mH_{max}$ per topology change. To quantify the topology change, we adopt the results from Sucec and Marsic [14]. According to this paper, average number of link state change events, i.e. topology changes, per time step is given as:

$$r_{link} = \Theta\left(\frac{\mu}{R} \bullet |E|\right) = \Theta\left(\frac{\mu}{R} \bullet N \bullet \frac{d}{2}\right) = \Theta(N) \quad \textbf{(1)}$$

Therefore, the average number of topology changes in the network grows asymptotically with the number of nodes in the network. $H_{max} < d$ and d is a constant predefined in the algorithm to limit the diameter of cluster formed. Therefore $M = O(m)$ per topology change. The cluster maintenance overhead, $OH_{CM} = O(mND)$.

3.5 Total Clustering Overhead

We summarized our analysis into Table 2. The total clustering overhead, OH_C, is the sum of the following three factors:

(i) Hello Protocol Overhead (OH_H)

(ii) Cluster Formation Overhead (OH_{CF})

(iii) Cluster Maintenance Overhead (OH_{CM})

Therefore, the total clustering overhead incurred by MobDHop clustering algorithm is $O(DN) + O(T_{sample}N) + O(mND)$. Dividing this results by D time steps, the total clustering overhead is $O(N) + O(N) + O(mN)$ in the network. Dividing this result by node count N yields total MobDHop clustering overhead, $OH_C = O(m)$ per node per time step. Since m is the average number of members in a cluster. It is feasible and fairly simple to add a parameter in order to limit the size of each cluster so that the cluster size formed by MobDHop is constrained to a constant value. As suggested in [13], the number and the size of clusters formed should be optimized in order to achieve desired protocol scalability. Clustering effort may not be creditable if a large number of clusters are formed. Conversely, a clusterhead might not be able to handle all the traffic generated by its members if the cluster size is too large. Favourable clustering algorithms should, therefore, form appropriate number of clusters of moderate size. A simple way to limit the cluster size is to set a maximum cluster size constraint as proposed in

[13]. Thus, the total clustering overhead of MobDHop can be constrained to OH_C = O(1) per node per time step as per **Claim 1**.

Table 2. Summary of overhead and convergence time analysis

Overhead Type	Time Complexity	Message Complexity	Total Overhead per Time Step
Hello Protocol	1	$< r_{hello}N$	$O(N)$
Cluster Formation	$< T_{sample} + 1$	$< N$	O(N)
Cluster Maintenance (per topology change)	$< (T_{sample} + H_{max}) + (H_{max})$	$< mH_{max}$	O(N)

4 Simulation Results

Simulations were performed using Qualnet Simulator 3.7 to investigate the message complexity of MobDHop in the presence of mobility. MobDHop was implemented as described in [4]. Maximum cluster size constraint was not imposed in these simulations. The overhead incurred by the Hello protocol is not taken into account in our simulation study since the Hello protocol is widely used in routing protocols as a neighbour discovery mechanism. There are two types of control packet in the MobDHop implementation, i.e. Join-Packet and Leave-Packet. Join-Packet and Leave-Packet are sent to the clusterhead whenever a node joins or leaves a clusterhead which is more than one hop away. The broadcast nature of the wireless medium allows one-hop neighbours to join and leave the cluster implicitly by tagging some additional fields in a Hello message. Since the first phase of MobDHop forms 1-hop clusters, no additional control packets are needed. Therefore, the only MobDHop overhead is the cluster maintenance overhead as discussed in **Section 3.4**.

Each simulation was executed for 900 seconds. H_{max} was set to 2 for all simulations. This value was chosen by considering the simulated network size. If a larger value is chosen, MobDHop will form less but larger multi-hop clusters. Transmission range is homogeneous for every node, i.e. R = 376 meters. The Random Waypoint Mobility Model was used in our simulations. In the first set of simulations, average node speed was varied while network density was fixed. Each scenario consisted of 50 nodes that were moving continuously in a 3000m x 3000m area. In the second set of simulations, network density was varied while average node speed was fixed at 12 m/s in all scenarios. Network size of 50, 100, 150, 200, 250 and 300 were simulated in a 5000m x 5000m area respectively. Fig. 2 shows the topology change rate increased with the average node speed and the number of nodes (network density). These results confirmed the analysis in [14] that the topology change rate under Random Waypoint Mobility Model is influenced by average node speed and the number of nodes in network. To evaluate the percentage of topology change that actually incurs clustering overheads, we measure the ratio of the number of topology changes that caused cluster structural changes to the total number of topology changes in the network. This ratio is named effective topology change ratio. As shown in Fig. 3(a), effective topology change ratio in the first set of simulations (varying average node speed) varied negligibly in the range of 0.3 and 0.4. This implies that, in terms of clustering overhead, MobDHop is less sensitive with respect to mobility since MobDHop is a mobility-adaptive clustering algorithm that forms clusters which are as stable as possible. Therefore, the clustering overhead caused by cluster changes due to mobility can be kept to minimum in MobDHop. Meanwhile, effective topology change ratio decreased sharply with the increase of network density as shown in Fig. 3(b). This implies that the network density does not affect the stability of MobDHop clusters. Although the total number of topology changes in the network increases with the network density, the number of topology changes that actually caused cluster structural change remained the same in networks with different density. In other words, MobDHop is suitable for networks with all kinds of density configuration.

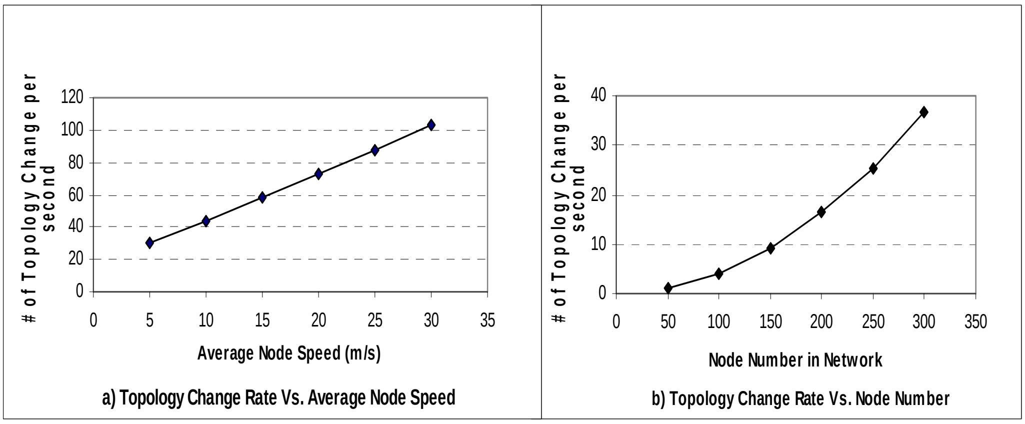

Fig. 2. Impact of average node speed and node density on the topology change rate

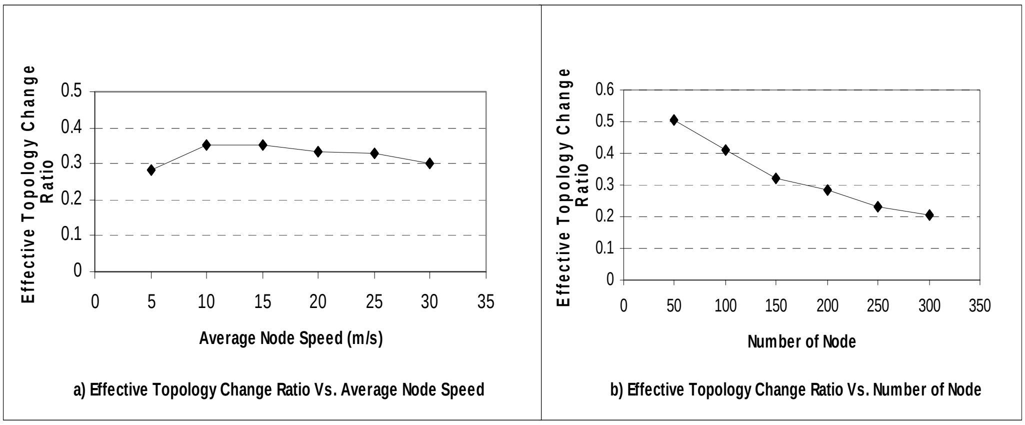

Fig. 3. Impact of average node speed and node density on the effective topology change

Fig. 4(a) and 5(a) show the number of control packets increased with the average node speed and network density respectively. Since a constant number of control packets will be incurred with a topology change, an increase in the number of control packets with average node speed and network density is foreseen. Meanwhile, Fig. 4(b) and 5(b) show the number of control packets per effective topology change was in the range of 1.1 to 1.7 for all scenarios. Effective topology change is the topology change that causes at least one cluster structural change. Our simulation results show that MobDHop incurs a consistent amount of control overhead per topology change. This is consistent with **Claim 1** in our theoretical analysis.

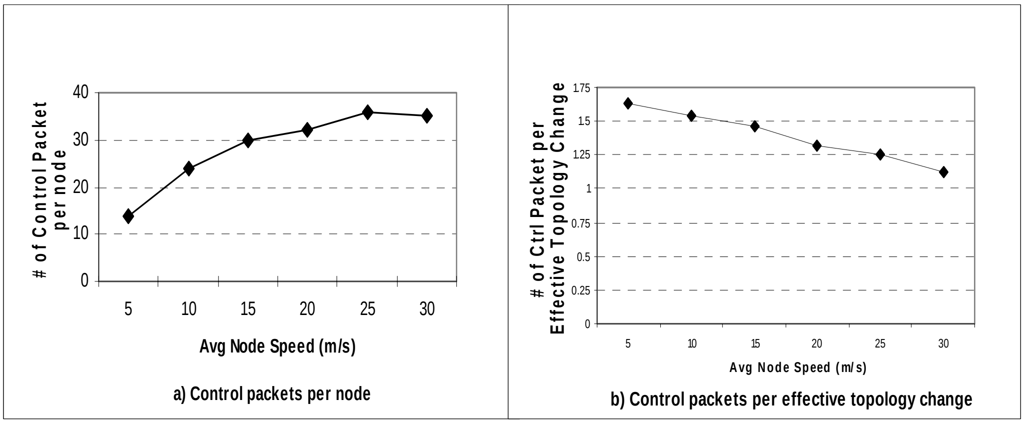

Fig. 4. The impact of average node speed on the MobDhop clustering overhead

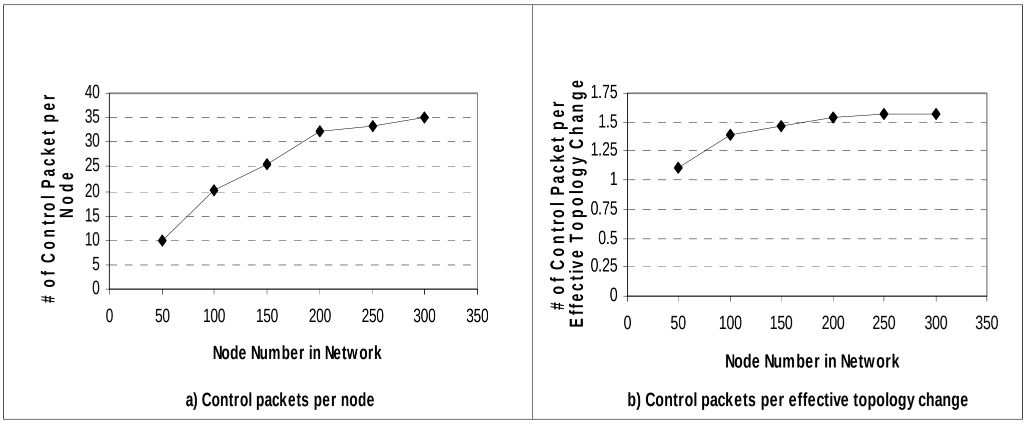

Fig. 5. The impact of node density on the MobDHop clustering overhead

As discussed in **Section 3.5**, total MobDHop clustering overhead is O(m) per node per time step, where m is the average number of members in a cluster. It is suggested to limit the maximum number of members allowed to join a cluster to avoid over-burdening a clusterhead especially when the node density is high. In our simulation, this constraint was not imposed. To examine this issue, the average number of members in clusters formed by MobDHop was plotted against a theoretical upper bound derived based on the expected node degree in a uniformly distributed square area. According to Bettstetter [15], the expected node degree E{D} in a square area where nodes are uniformly distributed is given as:

$$E\{D\} = n\frac{R^2\pi}{a^2}\left(1 - \frac{8}{3\pi}\frac{R}{a} + \frac{1}{2\pi}\frac{R^2}{a^2}\right) \quad \textbf{(2)}$$

There are at most $(E\{D\})^2$ neighbours in the 2-hop neighbourhood of one node. Therefore, a clusterhead may at most have $(E\{D\})^2$ members in its cluster since $H_{max} = 2$. It follows that a cluster may consist of at most $(E\{D\})^2 + 1$ members. As shown in Fig. 6, 50-nodes network was too sparse and most of the network nodes were isolated. Hence, a large number of single-node clusters were formed. When the network density was increased, cluster size also increased. As a result, it is necessary to impose a maximum cluster size in a high-density network in order to avoid high cluster maintenance overhead especially when the node speed is high and cluster contentions might happen in a higher frequency.

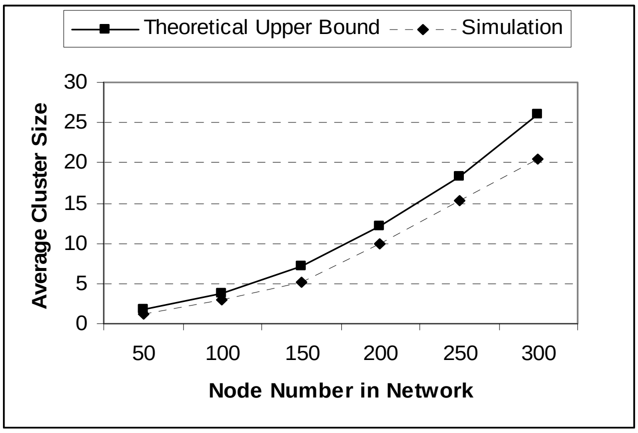

Fig. 6. Average cluster size formed by MobDHop

5 Analysis and Comparison among Five Clustering Algorithms

The time and message complexity of three one-hop clustering algorithms, i.e. Lowest-ID, MCC and DMAC will be presented next. As for multi-hop clustering, the time and message complexity of Max-Min D-Cluster are reviewed. Comparison is then made among these five clustering algorithms.

5.1 Analysis of Lowest-ID, MCC and DMAC and Max-Min D-Cluster

The theoretical analysis of the time and message complexity of Lowest-ID and MCC clustering algorithm is similar to the analysis of DMAC as presented in [11]. However, we assume the LCC improvement is applied on both Lowest-ID and MCC clustering algorithms. Therefore, a tight bound on the time and message complexity can be derived. The results are presented in Table 3.

The overhead of the hello messaging, OH_{H} is $O(N)$ per time step. The total overhead incurred during cluster formation, OH_{CF} is $O(N)$ per time step too. Each topological change will incur at most O(1) overhead. Therefore, the average number of link state change events based on the random waypoint mobility model is $\Theta(N)$. Similar to MobDHop, topological changes between two clusterheads may incur clusterhead contention and then a chain reaction where every cluster member changes their cluster membership. Hence, the upper bound of cluster maintenance overhead per time step is given by $OH_{\mathrm{CM}} = O(mN)$, where m is the cluster size. The total clustering overhead incurred by Lowest-ID or MCC is $O(N) + O(N) + O(mN) = O(N)$ per time step if we impose a similar constraint on the maximum cluster size and therefore m is a constant. Dividing this by the number of nodes yields a total clustering overhead, $OH_{\mathrm{C}} = O(1)$ per node per time step.

The Distributed Mobility-Adaptive Clustering (DMAC) is similar to the Lowest-ID and MCC clustering algorithms except for the clusterhead election criteria. The role of a node is determined by a weight which is associated with every node based on some predefined criteria, e.g remaining power, speed and node ID. Bettstetter and Konig [11] investigated the reaction of the DMAC algorithm towards topology changes in a network and analysed the message and time complexity of this algorithm. They observed the inevitable reclustering chain reactions which resulted from topology changes e.g. node addition, link failure and link establishment. Reclustering chain reaction may happen when a topology change involves a clusterhead with certain neighbourhood configurations that may lead to another clusterhead in its neighbourhood giving up its role. The effect of chain reactions is unpredictable and therefore only lower bounds for time and message complexity were provided. The worst case happens when a chain reaction occurs where a valid cluster structure takes at least 2 time steps to be formed with a message complexity of $1 + deg$.

Max-Min D-Cluster is a heuristic to form d-hop clusters in MANETs. Each node is at most d hops away from its clusterhead. The heuristic can be executed at regular intervals or whenever there is a topology change to maintain valid cluster structure. The heuristic is claimed to elect fewer clusterheads and form larger clusters with longer clusterhead duration on the average than the Lowest-ID algorithm. However, Max-Min D-Cluster does not take mobility pattern into account during cluster formation. The time complexity of Max-Min D-Cluster as presented in [10] is $O(2d + d)$. Whenever the heuristics is executed, each node has to send at least d messages before a cluster can be formed. Since the average number of link state change events, i.e. topology changes, per time step is $\Theta(N)$, the total overhead per time step incurred by Max-Min D is $O(dN)$. Since d is a predefined constant in the algorithm, the total overhead per time step is also $O(N)$.

5.2 Comparison among Five Clustering Algorithms

Table 3 summarizes the comparison of the time and message complexity among MobDHop, Max-Min D-Cluster, Lowest-ID, MCC and DMAC. For DMAC, the worst case lower bound for the time complexity is given in [11] during a new node event where all neighbours are either clusterheads with lower weight or ordinary nodes with lower weight.

Our analysis shows that the total clustering overhead of one-hop clustering or multi-hop clustering are similar in the asymptotic upper bound with respect to the number of nodes in network. MobDHop, Lowest-ID and MCC have a better time complexity than DMAC because the re-clustering chain reaction is avoided by the LCC improvement. LCC improvement provides a better performance in terms of message complexity. We integrate LCC improvements into MobDHop, Lowest-ID and MCC clustering to avoid re-clustering chain reactions. However, chain reaction may occur in MobDHop clusters but it is restricted to H_{max} hops. LCC was not integrated into DMAC in this paper since LCC will force the second property of a valid cluster structure given by Basagni [7] to be violated, i.e. every ordinary node affiliates with the neighbouring clusterhead with the largest weight.

Intuitively, LCC may have negative impact on Lowest-ID and MCC clustering algorithm. LCC may force clusterheads to retain their role until a clusterhead contention occurs. If there is no clusterhead contention, a clusterhead will retain its role until all member nodes leave the cluster. Appropriate re-clustering may not occur at all in the network and this may cause a large numbers of small clusters (consists of 1 or 2 nodes) to dominate the network. This phenomenon can be verified by network simulations and these are parts of the future work.

Table 3. Comparison of the time and message complexity among different clustering algorithms

Algorithm	Time Complexity per topology change	Message Complexity per topology change	Total Overhead per time step
MobDHop	$\leq (T_{sample} + H_{max}) + (H_{max})$	$\leq mH_{max}$	$O(N)$
Max-Min D-Cluster	$O(d)$	$O(d)$	$O(N)$
Lowest-ID	$O(1)$	$O(m)$	$O(N)$
MCC	$O(1)$	$O(m)$	$O(N)$
DMAC	$\geq 1 + deg$	≥ 2	$O(N)$

6 Conclusion

This paper analyzes the message and time complexity of different clustering algorithms and gives an insight into how the algorithm reacts to topology changes. Apart from a few well-established one-hop clustering algorithms (i.e. Lowest-ID, MCC and DMAC), this paper analyzes the performance of a multi-hop clustering algorithm, i.e. MobDHop. We argue that the number of packet transmissions per node per time step required for MobDHop to operate correctly in MANETs is O(1). We also provide the upper bound of time complexities for both cluster formation and cluster maintenance in MobDHop, which shares the same asymptotic bound with other 1-hop clustering algorithms. Simulations were carried out to investigate clustering overhead incurred by MobDHop. Simulation results agree with our analytical conclusion that a constant amount of clustering overhead is incurred by a topology change. Therefore multi-hop clustering should be feasible in mobile scenario. In future research, we will investigate other evaluation methods for clustering algorithms. A possible extension is to apply competitive analysis, which is widely used in online algorithms to compare the performance of different distributed online clustering algorithms in MANETs.

References

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

[2] C-C. Chiang, H-K. Wu, W. Liu, M. Gerla, Routing in clustered multihop mobile wireless networks with fading channel, in: Proceedings of the IEEE Singapore International Conference on Networks (SICON), 1997, pp. 197-211.

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

[4] I.I. Er, W.K.G. Seah, Mobility-based d-hop clustering algorithm for mobile ad hoc networks, in: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), 2004.

[5] P. Basu, N. Khan, T.D.C. Little, Mobility based metric for clustering in mobile ad hoc networks, in: Proceedings of the Workshop on Distributed Computing Systems, 2001, pp. 413-418.

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

[7] A. Ephremides, A. Wieselthier, D.J. Baker, A design concept for reliable mobile radio network with frequency hopping signaling, in: Proceedings of IEEE, 75(1) (1987) 56-73.

[8] A.K. Parekh, Selecting routers in ad hoc wireless networks, in: Proceedings of the SBT/IEEE International Telecommunications Symposium (ITS), 1994.

[9] S. Basagni, Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks, in: Proceedings of the IEEE Vehicular Technology Conference (VTC), 1999, pp. 889-893.

[10] A. D. Amis, R. Prakash, T.H.P. Vuong, D.T. Huynh, Max-min d-cluster formation in wireless ad hoc networks, in : Proceedings of 19th IEEE INFOCOM, 2000, pp. 32-41.

[11] C. Bettstetter, S. Konig, On the message and time complexity of a distributed mobility-adaptive clustering algorithm in wireless ad hoc networks, in: Proceedings of the 4th European Wireless, 2002, pp. 128-134.

[12] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, in: T. Imielinski, H. Korth (Eds.), Mobile Computing, vol. 353, Kluwer Academic Publishers, Norwell, MA, 1996, pp. 153-181.

[13] S. Banerjee, S. Khuller, A clustering scheme for hierarchical control in multi-hop wireless networks, in: Proceedings of the 20th IEEE INFOCOM, 2001, pp.1028-1037.

[14] J. Sucec, I. Marsic, Hierarchical routing overhead in mobile ad hoc networks, IEEE Transactions on Mobile Computing, 3(1) (2004) 46-56.

[15] C. Bettstetter, The cluster density of a distributed clustering algorithm in ad hoc networks, in: Proceedings of the IEEE Intern. Conf. on Communications (ICC), 2004.

Brief biographical notes about authors

Inn Inn Er is a PhD candidate in the School of Computing, National University of Singapore. She completed her B. Sc. in Department of Computer Science and Information System, University of Technology Malaysia, Malaysia in 2002. Her current research interest is in mobile ad hoc networks, specifically on clustering algorithm, unicast routing and multicast routing protocols.

Winston Khoon Guan SEAH received the Dr.Eng. degree from Kyoto University, Kyoto, Japan, in 1997. He is a Lead Scientist in the Networking Department of the Institute for Infocomm Research (I^2R). Prior to I^2R, he had been a Principal Member of Technical Staff, and director of the Internet Technologies programme in the Institute for Communications Research. He is an Adjunct Associate Professor in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Concurrently, he is an adjunct faculty in the Graduate School for Integrative Science and Engineering, and the Department of Computer Science in the National University of Singapore where he lectures in mobile computing. He is actively involved in research and development in the areas of mobile ad hoc and sensor networks, co-developed one of the first quality of service (QoS) models for mobile ad hoc networks (MANET), and has given keynote talks on the area of QoS for MANETs. His latest research focuses on mobility-enhanced protocols and algorithms for C^3 and sensing applications in terrestrial and oceanographic ad hoc sensor networks. He is also on the TPC of numerous conferences and reviewer of papers for many key journals and conferences in the area of MANET and sensor networks. He serves on the Steering Committee of the Asia-Pacific IPv6 Task Force, and is also an associate of the Singapore Advanced Research and Education Network (SingAREN.) He is a Senior Member of the IEEE. [Homepage: http://www1.i2r.a-star.edu.sg/~winston]