Analysis of Clustering and Routing Overhead for Clustered Mobile Ad Hoc Networks

Mingqiang Xue[2] Inn-Inn Er[2,1] Winston K.G. Seah[1,2♠]

[1]Networking Department, Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
[2]Department of Computer Science, School of Computing
National University of Singapore, Singapore 117543

Abstract

This paper presents an analysis of the control overhead involved in clustering and routing for one-hop clustered mobile ad hoc networks. Previous work on the analysis of control overhead incurred by clustering algorithms focused mainly on the derivation of control overhead in the Knuth big-O notation with respect to network size. However, we observe that the control overhead in a clustered network is closely related to different network parameters, e.g. node mobility, node transmission range, network size, and network density. This paper presents an analysis that captures the effects of different network parameters on the control overhead. The results of our work can provide valuable insights into the amount of overhead that clustering algorithms may incur in different network environments. This facilitates the design of efficient clustering algorithms in order to minimize the control overhead.

Keywords: Theoretical analysis, performance analysis, control overhead, clustering, mobile ad hoc networks, mobility model.

1. Introduction

Mobile ad hoc networks (MANET) are autonomous systems formed by mobile nodes without any infrastructure support. Routing in MANET is challenging because of the dynamic nature of the network topology. Although numerous routing protocols have been proposed for MANETs, such as DSDV [8] and AODV [9], these routing protocols are not suitable for large MANET because the overhead for maintaining up-to-date routing information at each node quickly becomes unacceptable as network size increases. Clustering is a technique that partitions a network into different groups or clusters, creating a logical hierarchy in the network. Normally, a particular node in a cluster that is elected to manage the cluster information is known as *cluster-head*, and other nodes in the cluster are known as *ordinary nodes* or *cluster-members*. Hierarchical routing in MANET, although may increase network path length, has been shown to be useful for routing in large networks [5]. By partitioning a network into different clusters, both

♠ Corresponding author's email: winston@i2r.a-star.edu.sg

storage and communication overheads for maintaining up-to-date routing information can be significantly reduced. However, clustering still incurs overhead that has yet to be investigated in depth. In this paper, we characterize different clustering overheads involved in a typical clustering algorithm and also investigate the amount of clustering overhead and control messages needed to maintain up-to-date routing information. As a whole, this overhead is known as clustering and routing overhead, or control overhead. Other metrics such as storage overhead, network path length, and convergence time are not considered in this analysis. Control overhead is an important metric for measuring the performance of a clustering algorithm since bandwidth is a limited and valuable resource in MANETs. It is shown by in [1] that the per-node capacity in a random ad hoc network with N nodes is $\Theta(\frac{1}{\sqrt{N \log N}})$, which is a decreasing function with network size N. Thus, as the network size increases, the utilization of bandwidth becomes a very critical factor that affects the overall performance of a network.

Over the past few years, numerous clustering schemes have been proposed [7]. Different clustering algorithms may use different *cluster-head* election rules, form clusters of different sizes, allow different hop distances between a *cluster-member* and its *cluster-head*, or use different schemes for cluster maintenance. Most of the prior work on clustering overhead focuses on the message and time complexity of a clustering algorithm, which is a rough approximation of clustering overhead with respect to the network size. In [19], the time complexity and message complexity for DMAC [21] are analyzed and in [20], comparisons for the clustering overhead in Knuth big-O notation [2] of several clustering schemes such as MobDHop [22], Max-Min [23], Lowest-ID (LID), Highest Connectivity Clustering (HCC) [15] and DMAC is presented. In [3] and [4], the hierarchical routing overhead for a network with O(log N) level of hierarchies using LCA[18] clustering scheme is systematically analyzed, and they conclude that the messages transmissions per node in their network model is O(log N). There is relatively little other work done on the analysis of clustering overhead.

In contrast, our analysis considers several important network parameters that affect the amount of clustering control overhead incurred, including network size, node mobility, node transmission range, and node density. This analysis provides a better model of the clustering overhead that reflects the relationship between the clustering overhead and these network parameters, and could lead to further investigations on the design of a better clustering algorithm for different network conditions. Apart from clustering overhead, our analysis also provides an insight on how these network parameters affect the amount of routing overhead incurred to maintain up-to-date routing information in a proactive manner within every cluster assuming a general hybrid routing protocol is in operation.

The rest of this paper is organized as follows: In section 2, we provide an overview of control messages required for clustering and proactive routing. In section 3, we present the assumptions of our network model and analyze the control overhead for general one-hop clustering algorithms based on this network model. The ratio of *cluster-heads* in a network is viewed as a variable in this analysis which may vary across different one-hop clustering algorithms and Section 4 presents simulation results to verify our analysis. In section 5, we analyze the ratio of *cluster-heads* for a well-known one-hop clustering algorithm, namely LID clustering. This ratio, when substituted into our analysis in Section 3, gives a valuable insight to the amount of clustering overhead incurred by Lowest-ID clustering. In section 6, we highlight the relevance and usage of this analysis in the design of clustering algorithms, and conclude in section 7.

2. Overview on Clustering and Routing Overhead

As the topology of a MANET changes, control messages are generated by nodes to 1) update routing information when route changes occur and 2) update a node of changes to its cluster membership or *cluster-head*. Different clustering algorithms may use different schemes and generally, three types of control messages are needed:

a) Beacon, commonly known as *HELLO* message, for nodes to learn the environment and identities of adjacencies (neighbors).
b) Cluster management (which shall refer to as *CLUSTER* message) for nodes to adapt to cluster changes and update its role.
c) Route management (which shall refer to as *ROUTE* message) for nodes to learn of route changes.

The *HELLO* message is often used. Periodically broadcasted by each node, a node adds a new node to its neighbor-list when it hears the new node's *HELLO*, and a node deletes a node from its neighbor-list when it could not hear that node for some predetermined duration. Ideally, a *HELLO* message should be sent every time a node has a new neighbor.

The *CLUSTER* message is a generalization for a sequence of messages needed by nodes to update cluster information. The execution of clustering algorithms can usually be divided into cluster formation stage and cluster maintenance stage. The *CLUSTER* message only refers to a sequence of messages needed by nodes when cluster change occurs in the cluster maintenance stage. We do not consider initial cluster formation, as we are focusing the long term performance of clustering. Such a sequence of messaging may be implemented in different ways. For example, in LID clustering, a node sends a *CLUSTER* message which indicates the cluster it belongs to when it has decided its own cluster. While in DMAC, two types of messages are sent: CH(v), sent by a node v to inform its neighbors that it is going to be a *cluster-head*, and JOIN(v,u), sent by a node v inform its neighbors that it will join the cluster whose *cluster-head* is u. Some clustering algorithms use implicit clustering, hence, the *CLUSTER* message is not needed, e.g. Passive Clustering [24]. Another important issue is when this *CLUSTER* message is sent: periodically or reactively in response to changes in cluster conditions; our analysis assumes reactive cluster maintenance. There are two main reasons for this. Firstly, reactive cluster maintenance usually requires less control messages; analyzing control overhead using reactive cluster maintenance can provide a lower bound. Secondly, the clustering maintenance scheme is similar for most clustering algorithms, i.e. the same set of events could trigger the *CLUSTER* message. Thus, it can be used for analyzing the control overhead of *CLUSTER* message for all clustering algorithms. The following properties for 1-HOP clustered networks should be ensured and any violation will trigger *CLUSTER* messages at relevant nodes:

P1. No *cluster-heads* are directly connected to each other

P2. Each node should be affiliated to one cluster; i.e. each ordinary node should have only one *cluster-head* and be at most one hop away.

The Least Clusterhead Change (LCC) scheme introduced in [14] is an extension to clustering algorithms, like LID and HCC, which possess the above properties. The *ROUTE* message is periodically exchanged by neighboring nodes to update routing information, and ideally, sent every time route table information changes.

3. Theoretical Analysis

In this section, we present the clustering overhead analysis taking into consideration several network parameters, viz. network size, cluster size, network density, node velocity, and transmission range; only one-hop clustering algorithms are considered. First, the assumptions on the network model, clustering algorithm and mobility model used in our analysis are discussed, followed by a list of notations. We then derive properties of our network model, and for investigating control overhead in a one-hop clustering algorithm.

3.1 Assumptions

There are N nodes in the network, each with transmission range r. If two nodes are within the transmission range of each other, they form a bi-directional link between each other and become neighboring nodes. A node knows of its neighbors through the "HELLO" mechanism.

The clustering algorithm used can be any one-hop clustering algorithm as long as the cluster structure it forms satisfies the two properties *P1* and *P2* in section 2. After the cluster formation stage, each node is assigned a role of either *cluster-member* or *cluster-head* and belongs to some cluster. In the cluster formation stage, the clustering algorithm selects a node as the *cluster-head* with a probability P_{HEAD}. Thus, the expected number of clusters or expected number of *cluster-heads* is $N \cdot P_{HEAD}$. Here, P_{HEAD} varies for different clustering algorithms and it can be viewed as a metric of a particular clustering algorithm, which describes how *cluster-heads* are distributed over the network. In the cluster maintenance stage, each node keeps broadcasting *CLUSTER* messages when cluster changes happen due to violations of *P1* and *P2*. Upon hearing the *CLUSTER* messages from its neighbors, a node updates its role according to the rules defined by the clustering algorithm used. For a good clustering algorithm, the number of clusters should not change drastically in steady state and it can be assumed to remain constant in the maintenance stage.

One of the main purposes for clustering is to enable scalable routing. A typical hierarchical routing protocol in MANETs consists of intra-cluster and inter-cluster routing. In our analysis, we assume a hybrid routing protocol which uses proactive intra-cluster routing and on-demand/reactive inter-cluster routing. In our analysis, control overhead incurred by the clustering algorithm and proactive intra-cluster routing protocol are taken into account. Control overhead incurred by on-demand inter-cluster routing protocol is not considered in this analysis.

3.2 Mobility Model and Node Spatial Distribution

Different mobility models have been developed and adopted in the empirical analysis of MANETs [11]. The Random Waypoint (RWP) and Random Walk (RW) are two popular mobility models commonly used in MANET simulations. However, RWP and RW are known to be unfavorable for theoretical analysis due to the difficulty in capturing their mobility behavior as well as the uneven node distribution they generate. It has being shown in [12] that the node spatial distribution of RWP in steady state is not uniform given the initial node distribution being uniform. In this analysis, we adopt the Constant Velocity (CV) model [10]. It preserves the uniform node spatial distribution and link change rate in the network is mathematically tractable. The CV model requires an infinite number of nodes to be randomly distributed on a boundless plane. However, in our analysis, we propose a variant of CV, called Bounded Constant Velocity (BCV) model, by assuming a bounded area with a network size N:

1) Initially, at time $t = 0$, nodes are uniformly distributed on an infinitely large area with density ρ. All the nodes randomly choose directions from a uniform distribution.
2) At time $t > 0$, each node starts moving with a constant velocity v in the direction chosen.
3) A node does not change the direction and velocity while moving.
4) A bounded square region S is selected in the boundless plane with the average number of nodes within S being N at any point of time

We show later that BCV produces uniform node spatial distribution, and its link change rate is computable. The simulations also show that BCV is a good approximation of the Random Mobility model.

3.3 Symbols and Notations

In this subsection, we briefly discuss the symbols and notations used in the subsequent analysis. The average cluster size is m (the average number of nodes in a cluster, including the *cluster-head*) which is given by $m = \frac{N}{n}$ where n is the number of clusters in the network. Therefore, the probability that a randomly selected node being a *cluster-head* is given by $P_{HEAD} = \frac{n}{N}$. The border length of the square area, a is given by $a = \sqrt{|S|}$ and $r < a$. For each node, the link generation rate (non-neighboring nodes become neighbors) and link break rate (neighboring nodes move away from transmission range of each other) in the network are denoted by λ_{gen} and λ_{brk} respectively, with the total link change rate given by $\lambda = \lambda_{gen} + \lambda_{brk}$.

The size of the *HELLO* message, *CLUSTER* message and one routing table entry (*ROUTE* message) are p_{hello}, $p_{cluster}$ and p_{route} respectively; the corresponding broadcast rates are denoted by f_{hello}, $f_{cluster}$ and $f_{routing}$ respectively. An arbitrary node in the region S has d network neighbors (neighbors outside S are not considered.) Lastly, the control overheard (bits/sec) from *HELLO*, *CLUSTER* and *ROUTE* messages are O_{hello}, $O_{cluster}$ and $O_{routing}$ respectively.

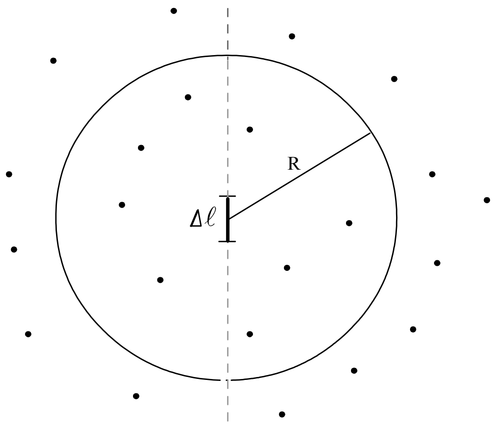


Figure 1: For a randomly selected segment Δl, the rate that nodes cross from one side is the same as the rate that nodes cross from the other side

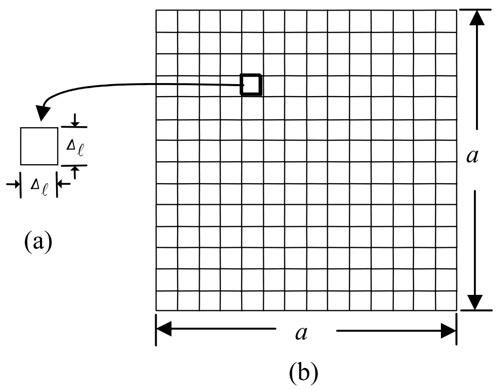


Figure 2: Consider a small square with length a. The rate that nodes enter the rectangle area and the rate that nodes leave the rectangle area from each border are the same.

3.4 Properties of Network Model

The following three claims are made regarding the several important properties of the network model adopted in our analysis**:**

Claim 1: The mobility model preserves uniform spatial node distribution at any point of time.

Proof: The proof can be accomplished in two stages. First, we show that for a randomly selected line segment in the plane, the rate that nodes cross it from one side of the line segment to the other is the same as the rate in the opposite direction. We randomly select a line segment Δl with length $|\Delta l|$. We draw a circle centered at the middle of Δl with radius R, as shown in Figure 1. At time $t = 0$, the number of nodes uniformly distributed in the semicircular region to the left of the line is segment the same as the number of nodes uniformly distributed in the right semicircular region to the right of the line segment. At time $t > 0$, the rate at which some of nodes in the left semicircular region cross Δl can be expressed as a function $F(R)$. Similarly, the number of nodes crossing in the opposite direction can be expressed by the same function $F(R)$ because of reflexivity. As $R \to \infty$ the circular area approaches an infinitely large plane. Hence, the rate of nodes cross from the left of Δl is the same as rate of nodes cross from the right of Δl. In stage two, we observe that a square area with border length $|\Delta l|$, the number of nodes in the square remains the same at any time $t > 0$, as shown in Figure 2(a). This is because the rate of nodes entering the square from each border is the same as the rate of nodes leaving the square from each border. We further observe that any large square area can be divided into many small squares. In Figure 2(b), we divided an $a \times a$ square into a large number of small squares, each with a border length $|\Delta l|$. Thus, the number of nodes in each small square area is the same. When $|\Delta l| \to 0$, each small square becomes a single point in the $a \times a$ square, with the same node density. Hence, we have shown that the BCV mobility model preserves uniform spatial node distribution at any time $t > 0$.

Claim 2: The expected number of network neighbors, d, of a randomly selected node in S is given by:

$$d = (N-1)\frac{r^2\rho}{N}\left(\frac{r^2}{2}\frac{\rho}{N} - \frac{8r}{3}\sqrt{\frac{\rho}{N}} + \pi\right) \tag{1}$$

Proof: It has been shown in [13] that the cumulative distribution function for the link distance between two nodes that are randomly placed in a square area with border length D is:

$$\mathrm{F}_d(\gamma = \xi D) = \begin{cases} 0, & \xi < 0 \\ \xi^2\left(\frac{1}{2}\xi^2 - \frac{8}{3}\xi + \pi\right), & 0 \le \xi < 1 \\ \frac{4}{3}\sqrt{\xi^2 - 1}(2\xi^2 + 1) - \left(\frac{1}{2}\xi^4 + 2\xi^2 - \frac{1}{3}\right) & \\ \quad +2\xi^2[\sin^{-1}(1/\xi) - \cos^{-1}(1/\xi)], & 1 \le \xi < \sqrt{2} \\ 1, & \sqrt{2} \le \xi \end{cases} \tag{2}$$

$\mathrm{F}_d(\xi D)$ gives the probability that two randomly selected nodes with transmission range ξD in the square of border length D are connected. E.g. when $\xi D = 0.1D$, $\mathrm{F}_d(0.1D) = 0.0288$ which denotes that a randomly selected pair is connected with a probability of 0.0288. Thus, in our network model where $r < a$, within an area S the expected number of neighbors of a randomly selected node is $(N\text{-}1)F_d(r)$. Substituting $D = a = \sqrt{\frac{N}{\rho}}$, we obtain Eqn (1).

Claim 3: The expected link generation rate at each node in S with other nodes, λ_{gen}, is $\frac{8dv}{\pi^2 r}$, and the link break rate λ_{brk} is the same as link generation rate. Therefore, the total link change rate at each node with other nodes in the plane is given by:

$$\lambda = \frac{16dv}{\pi^2 r} \qquad (3)$$

Proof: In [10], the authors derived the link change rate for the CV model. Both the link generation and break rates at each node are $\frac{8}{\pi}\rho r v$. Hence, the total link change rate is $\frac{16}{\pi}\rho r v$. The link change rate of CV model is different from link change rate of BCV model because the link change with nodes outside the square region S is not accounted for in BCV. In BCV, the total number of connected neighbors of a node is $\pi r^2 \rho$, but only d of them are within the rectangle region. We assume each established link is equally likely to break no matter whether it is inside or outside S. Thus, among the total link changes at a node in unit time, $\frac{d}{\pi r^2 \rho}$ of link changes happen inside S. Thus, the link change rate for a node in S with other nodes in S is $\frac{16dv}{\pi^2 r}$. The link break rate and link generation rate is half of the link change rate, which is $\frac{8dv}{\pi^2 r}$.

3.5 Control Overhead Analysis

The overhead of each type of the control messages, viz. *HELLO*, *CLUSTER*, and *ROUTE*, is proportional to its rate of broadcasting. Our analysis provides the lower bound for the overheads by assuming that each cluster and route change can be detected. In the following, we estimate the rate of each type of control messages and its overhead:

3.5.1 "HELLO" Overhead

The frequency of *HELLO* message at each node should be equal at least to the rate of its neighbor changes. For a randomly selected node n_0, any link generation between it and another node n_i should be noticed by both nodes and add each other to their neighbor-lists. While any link break between n_0 and its neighboring node n_j should also be noticed by the two entities so that they remove each other from their neighbor-lists. The link generation between two neighbors can be notified by both sending *HELLO* messages, and each of the nodes can hear the *HELLO* message send by the other node. While link break between two nodes cannot be notified via sending *HELLO* messages because the two nodes cannot hear each other. Usually, the link break event is sensed via a soft timer approach. When a node cannot hear its neighbor for a pre-configured time, it removes that neighbor from its neighbor-list. Hence, in order to learn a new neighbor immediately when a new link is formed, the rate of *HELLO* message broadcast at each node should at least equal the link generation rate. Therefore,

$$f_{hello} = \lambda_{gen} \tag{4}$$

and the control overhead at each node due to *HELLO* messages is $p_{hello} \cdot f_{hello}$. We substitute $\frac{8dv}{\pi^2 r}$ for λ_{gen}, and replace d with Eqn (1) in *claim 2*, to obtain the control overhead of hello message at each node O_{hello} with respect to N, r, ρ, p_{hello} and v as

$$O_{hello} = p_{hello}(N-1)\frac{8\rho r v}{\pi^2 N}\left(\frac{r^2}{2}\frac{\rho}{N} - \frac{8r}{3}\sqrt{\frac{\rho}{N}} + \pi\right) \tag{5}$$

3.5.2 Clustering Overhead

CLUSTER messages are sent at relevant nodes when the two properties *P1* and *P2* in section 2 are violated, that is, when two *cluster-heads* become neighboring nodes, or a node does not have a neighboring *cluster-head* and itself is *cluster-member*. *CLUSTER* messages need to be sent at relevant nodes in order to re-adjust cluster structure and re-satisfy the two properties. Besides *cluster-head* changes, changes of membership from one cluster to another also require sending of *CLUSTER* messages. The above cluster changes can be categorized into two link change events: 1) link break between *cluster-members* and their respective *cluster-heads*, and 2) link generation between two *cluster-heads*. All other link change events do not change the clusters and thus no *CLUSTER* message is sent. The two link change events may cause certain *CLUSTER* message overhead, and we will look at the two events respectively:

1) link break between *cluster-members* and their respective *cluster-heads*

This event causes a node to change its cluster, or become a *cluster-head* when it has no neighboring *cluster-heads*. The *CLUSTER* messages due to this type of link change are sent by *cluster-members*. The ratio of such link breaks to total link breaks should be equal to the ratio of links between *cluster-members* and *cluster-heads* divided by the total number of links in the entire network. The total number of links involving *cluster-heads* (with *cluster-head* at one end of the link) should be equal to the total number of *cluster-members*, that is, $N(1\text{-P}_{\text{HEAD}})$, since each *cluster-member* forms a link with its respective *cluster-head*. In a graph, the number of edges in the graph equals half of the sum of degrees of each node. Thus, the total number of links for the entire network is half the sum of network neighbors of all nodes within *S*, which is $(Nd)/2$. Therefore, the rate of *CLUSTER* message at each *cluster-member* due to link break with *cluster-heads* should be equal to:

$$\frac{N(1-\text{P}_{\text{HEAD}})}{Nd/2}\lambda_{brk} = \frac{16v(1-\text{P}_{\text{HEAD}})}{\pi^2 r} \tag{6}$$

The number of *CLUSTER* messages sent by *cluster-members* in the network in unit time due to link break with *cluster-heads* is:

$$N(1-\text{P}_{\text{HEAD}})\frac{16v(1-\text{P}_{\text{HEAD}})}{\pi^2 r} = N\frac{16v(1-\text{P}_{\text{HEAD}})^2}{\pi^2 r} \tag{7}$$

2) link generation between two *cluster-heads*

When a link is generated between two *cluster-heads*, one of the *cluster-heads* needs to give up its *cluster-head* role, which is to be decided by the clustering algorithm. However, the *CLUSTER* control message overhead incurred is the same. When one *cluster-head* drops its role it needs to send one *CLUSTER* message informing that it is no longer a *cluster-head* and the new cluster it belongs to. The original *cluster-members* of this former *cluster-head* lose their original cluster memberships. Thus, each of the nodes needs to send a *CLUSTER* message informing of their new clusters. We need not consider any chain reaction effects in our analysis as it does not affect our lower bound analysis. Every time a link between two *cluster-heads* appears, the number of *CLUSTER* messages generated is the same as the number of nodes in the cluster that needs to undergo re-clustering.

With the *cluster-heads* randomly distributed in the network, the total number of *cluster-heads* is $N\text{P}_{\text{HEAD}}$ and, the density of the *cluster-heads* spatial distribution is $\text{P}_{\text{HEAD}}\rho$. Since each *cluster-head* moves in a randomly selected direction with constant velocity, the link generation rate between two *cluster-heads* follows the analysis in *claim 3*:

$$\frac{8d'v}{\pi^2 r} \tag{8}$$

where

$$d' = (N\mathrm{P}_{\mathrm{HEAD}} - 1)\frac{r^2\rho}{N}\left(\frac{r^2}{2}\frac{\rho}{N} - \frac{8r}{3}\sqrt{\frac{\rho}{N}} + \pi\right) \tag{9}$$

Each of such link change causes *m* *CLUSTER* messages. Therefore, the total number of *CLUSTER* messages sent in the network due to link generation between two *cluster-heads* in unit time is:

$$nm\frac{8d'v}{\pi^2 r} = N\mathrm{P}_{\mathrm{HEAD}}\frac{1}{\mathrm{P}_{\mathrm{HEAD}}}\frac{8d'v}{\pi^2 r} = N\frac{8d'v}{\pi^2 r} \tag{10}$$

Combining (7) and (10), the rate of *CLUSTER* message sent at each node in unit time is:

$$f_{cluster} = \frac{16v(1-\mathrm{P}_{\mathrm{HEAD}})^2 + 8d'v}{\pi^2 r} \tag{11}$$

Therefore, the control overhead due to *CLUSTER* messages at each node is:

$$\begin{aligned} O_{cluster} &= p_{cluster}\frac{16v(1-\mathrm{P}_{\mathrm{HEAD}})^2 + 8d'v}{\pi^2 r} \\ &= p_{cluster}\frac{16v(1-\mathrm{P}_{\mathrm{HEAD}})^2 + 8(N\mathrm{P}_{\mathrm{HEAD}} - 1)\frac{r^2\rho}{N}\left(\frac{r^2}{2}\frac{\rho}{N} - \frac{8r}{3}\sqrt{\frac{\rho}{N}} + \pi\right)v}{\pi^2 r} \end{aligned} \tag{12}$$

3.5.3 Routing Overhead

One of the main benefits of hierarchical routing in MANET with clustering is that each node only needs to keep routing information of the cluster it resides in, while in a flat structure a proactive routing protocol requires nodes to keep global routing information of the entire network which may be substantial when the network size grows. In steady state, a particular node in a cluster should be updated with the routes to other nodes in the cluster and the storage overhead is proportional to the size of the cluster. When a route changes due to link change within a cluster, this information should be propagated through the cluster for every node in the cluster to update their routing tables. Every link change within the cluster will initiate a round of routing information broadcasting to update the routing information at each node.

For example, in Figure 3, nodes A and B are the *cluster-heads* of two clusters, and we call cluster A and cluster B respectively. Node A1 and A2 belongs to cluster A, and node B1 to B5 belongs to cluster B.

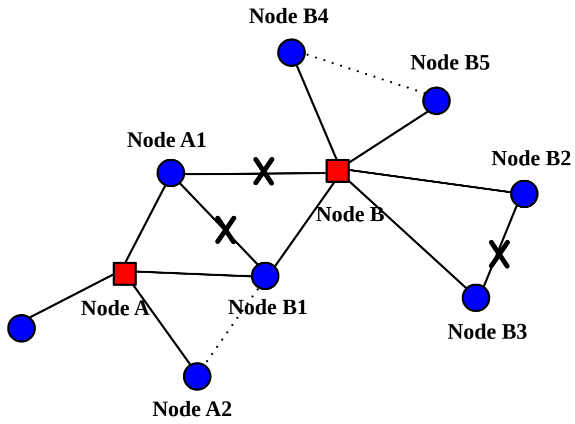


Figure 3: Link changes between nodes in the same cluster that causes routing information changes.

A1 only needs to keep routing information with other nodes in cluster A, and B1 only needs to keep routing information with other nodes in cluster B. The link changes between nodes in different clusters (e.g. break between A1 and B1/B, or new link between A2 and B1) does not cause any routing update, while the link change between nodes in the same cluster (e.g. break between B2 and B3, or new link between B4 and B5) causes routing updates. The portion of link changes that causes routing updates should be proportional to the number of links formed between nodes in the same cluster divided by the total number of links among nodes in the entire network. The frequency of routing information update is (derivation details in the appendix):

$$f_{routing} = \frac{32v((1-\mathrm{P_{HEAD}})^2(-\frac{3\sqrt{3}}{8\pi}+\frac{1}{2})+(1-\mathrm{P_{HEAD}})\mathrm{P_{HEAD}})}{\pi^2 r \mathrm{P_{HEAD}}^2} \qquad (13)$$

and the control overhead due to routing update:

$$O_{routing} = p_{route}\frac{32v((1-\mathrm{P_{HEAD}})^2(-\frac{3\sqrt{3}}{8\pi}+\frac{1}{2})+(1-\mathrm{P_{HEAD}})\mathrm{P_{HEAD}})}{\pi^2 r \mathrm{P_{HEAD}}^3} \qquad (14)$$

From the above analysis of overhead for the three types of control messages, the total control overhead in our network environment in bits per second is $O_{hello} + O_{cluster} + O_{routing}$. This control overhead is closely related to the link change rate among the network as we had shown above. In the following section, we validate these analytical results with simulations.

4. Simulation Studies

To validate our analysis, we simulated the different network scenarios and measured the frequencies of each category of control messages. Having used CV and BCV to model the random mobility model, we adopt a special case of RWP, which has similar properties as BCV in terms of link change rate and node spatial distribution. The RWP model used in the simulation has the following properties: initially N nodes are randomly uniformly distributed in an $a \times a$ square region. Then, following procedures start at time t=0:

1) At an arbitrary time epoch t, each node is static and selects a direction from a uniform distribution.
2) During time interval (t, $t + \tau$), each node moves in the selected direction with the same velocity v. τ is a configurable variable. If a node hits the border of the square region, it reappears at the same position in the opposite border and continues moving without changing its direction.
3) At time $t + \tau$, repeat step 1)

The clustering algorithm used is LID; r, ρ and v are configurable variables of the system, except P_{HEAD} which is determined by the LID clustering algorithm. In our simulations, P_{HEAD} for LID is measured in real time during the simulation. We compare control message frequencies measured from the simulations with the theoretical analysis. Figure 4 shows the changes of frequencies in the control messages by varying r and keeping other variables fixed. Figure 5 shows how control message frequencies change with node velocity. In Figure 6, the relations between control messages and network density are presented. As shown in all the figures, our analytical results for control message frequencies closely approximate in simulations results. Thus, our analysis is a good approximation of control overhead of the clustering algorithm.

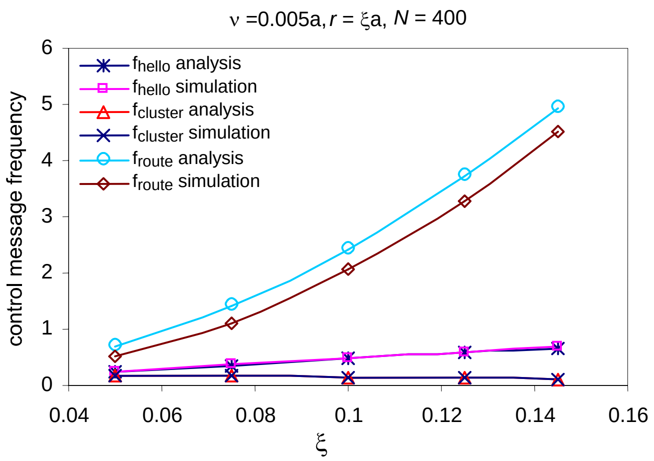


Figure 4: Control message frequencies with increasing transmission range

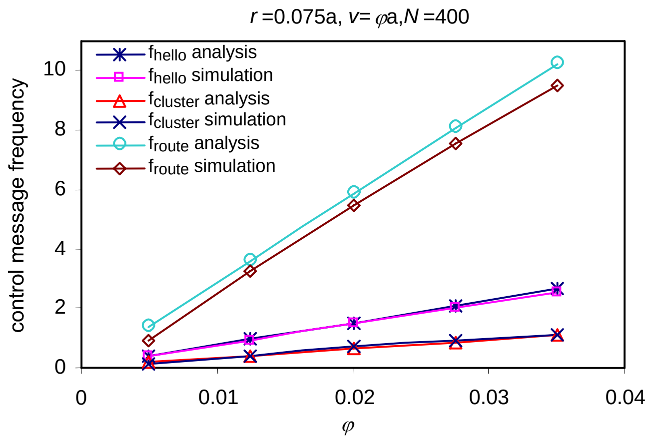


Figure 5: Control message frequencies with increasing node velocity

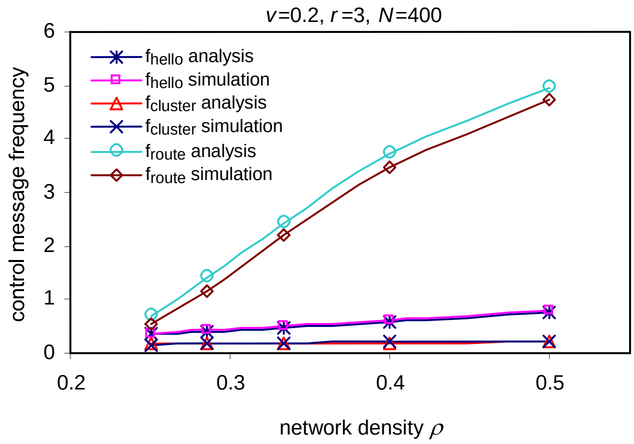


Figure 6: Control message frequencies with increasing network density

5. Case Study: Lowest-ID Clustering Algorithm

In the previous sections, we derived and validated the control overhead for a general one-hop clustering algorithms in terms of N, ρ, v, r and P_{HEAD}. While N, ρ , v, r are configurable network parameters, P_{HEAD} depends on one-hop clustering algorithm used; different clustering algorithms form clusters of different sizes, and thus P_{HEAD} varies accordingly. P_{HEAD} for a particular clustering algorithm can either be empirically measured from simulations or derived from theoretical analysis. In this section, we analytically derive P_{HEAD} for a simple but widely used clustering algorithm, i.e. Lowest-ID (LID) Clustering Algorithm [15][16].

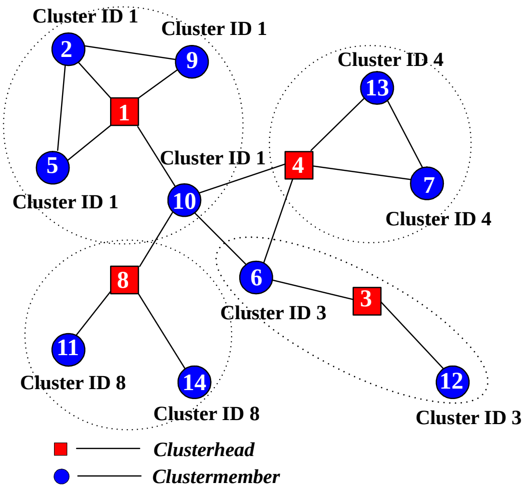


Figure 7: Clustered network using Lowest ID Clustering

LID assumes each node has a unique id and the nodes do not move during cluster formation process. A node is a *cluster-head* if and only if it has the smallest id among nodes in its immediate neighborhood that have not joined any cluster. The clustering algorithm is guaranteed to terminate, and each node is assigned a role of ordinary-node (also known as *cluster-member*) or *cluster-head*. Figure 7 is an example of a network using LID clustering. The *cluster-head* id is also the cluster ID of all nodes in that cluster.

5.1 P_{HEAD} in LID Clustering

Since at the end of the cluster formation phase each node is assigned a role of either *cluster-member* or *cluster-head*, we could define P_{MEMBER} with respect to P_{HEAD} which is the probability that a randomly selected node being a *cluster-member*. Therefore,

$$P_{MEMBER} + P_{HEAD} = 1 \tag{15}$$

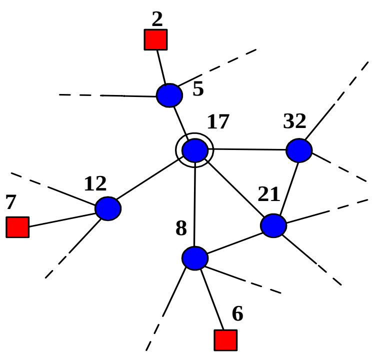


Figure 8: A randomly selected node is clusterhead if and only if all the nodes in its neighborhood that has smaller ids join other clusters as clustermembers

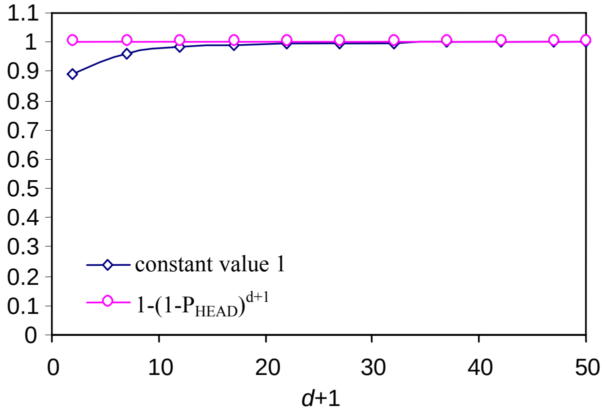
According to the rules of LID, a node is a *cluster-head* if and only if it has smallest id among the nodes in its closed neighborhood that have not being assigned any roles. In other words, if there are nodes with smaller ids in the surroundings of our randomly selected node, these nodes should become *cluster-members* of other clusters. This idea is illustrated in Figure 8. There are three nodes with smaller ids than node 17 in its neighborhood, namely nodes 5, 8 and 12. Node 17 can become a *cluster-head* if and only if these three nodes join other clusters. That is, node 5 joins the cluster with *cluster-head* id 2, node 8 joins the cluster with *cluster-head* id 6, and node 12 joins the cluster with *cluster-head* id 7. In all other cases, node 17 can never become a *cluster-head*. For analysis, we assume that a randomly selected node n_0 is the i-th smallest node in its closed neighborhood, which means there are i-1 nodes with smaller ids than n_0, where $i = 1..d+1$. The number of nodes in the closed neighborhood of n_0 is d+1and thus the largest possible value of i is d+1. When i =1 and i-1=0, node n_0 is the smallest node in its closed neighborhood. When node n_0 is the i-th smallest node in its closed neighborhood, the probability that the node n_0 being a *cluster-head* is thus $\mathrm{P}_{\mathrm{MEMBER}}{}^{i-1}$ which is the probability that all the i-1 nodes that have smaller ids are members of other clusters. A randomly selected node is the i-th smallest node for any i from 1 to d+1 with probability $\frac{1}{d+1}$. Hence, by summing over all possible i and substituting $\mathrm{P}_{\mathrm{MEMBER}}$ with 1-$\mathrm{P}_{\mathrm{HEAD}}$, we get:

$$\begin{aligned} \mathrm{P}_{\mathrm{HEAD}} &= \frac{1}{d+1}\sum_{i=1}^{d+1}(1-\mathrm{P}_{\mathrm{HEAD}})^{i-1} \\ \Rightarrow \mathrm{P}_{\mathrm{HEAD}}{}^{2} &= \frac{1-(1-\mathrm{P}_{\mathrm{HEAD}})^{d+1}}{d+1} \end{aligned} \qquad (16)$$

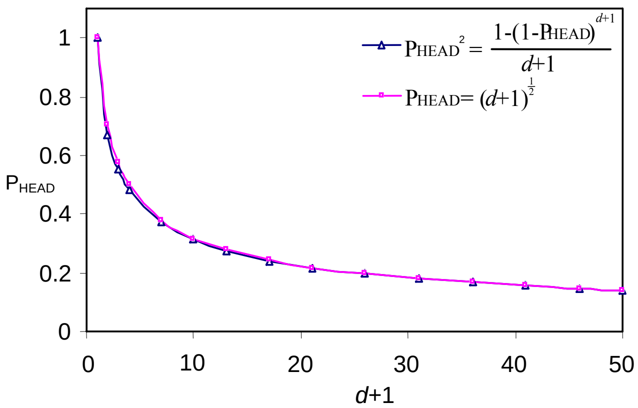
1-$\mathrm{P}_{\mathrm{HEAD}}$ is a positive value between 0 and 1, and d represents the average number of nodes in the transmission range of a selected node, thus $(1-\mathrm{P}_{\mathrm{HEAD}})^{d+1} \to 0$ as d+1 increases. Thus $1-(1-\mathrm{P}_{\mathrm{HEAD}})^{d+1} \to 1$ as d+1 increases, as shown in Figure 9(a). By substituting $1-(1-\mathrm{P}_{\mathrm{HEAD}})^{d+1}$ in Eqn (16) with 1, the expression can be rewritten as:

$$\begin{aligned} \mathrm{P}_{\mathrm{HEAD}}{}^{2} &= \frac{1}{d+1} \\ \Rightarrow \mathrm{P}_{\mathrm{HEAD}} &= \frac{1}{\sqrt{d+1}} \end{aligned} \qquad (17)$$

In Figure 9(b), we plot Eqn (16) and its approximation derived in Eqn (17) to show that the approximation is indeed accurate.



(a) $1\text{-}(1\text{-}P_{HEAD})^{d+1} \rightarrow 1$ as d+1 increases.



(b) P_{HEAD} as a function of d+1

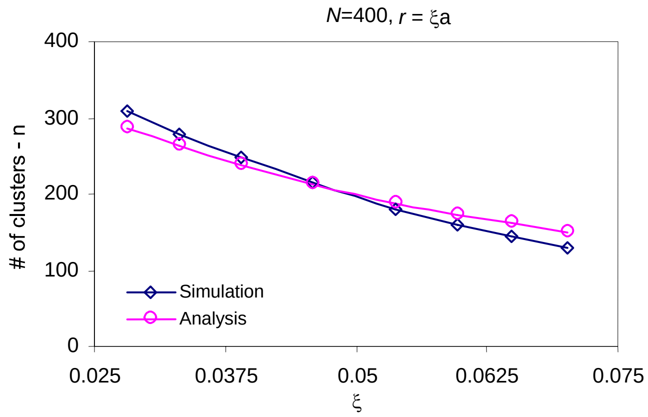
Figure 9: Validation of Eqn (16)

To reflect how P_{HEAD} varies with N, ρ and r, we substitute d with $(N-1)\frac{r^2\rho}{N}(\frac{r^2}{2}\frac{\rho}{N}-\frac{8r}{3}\sqrt{\frac{\rho}{N}}+\pi)$ from *claim 2* to obtain the following equation:

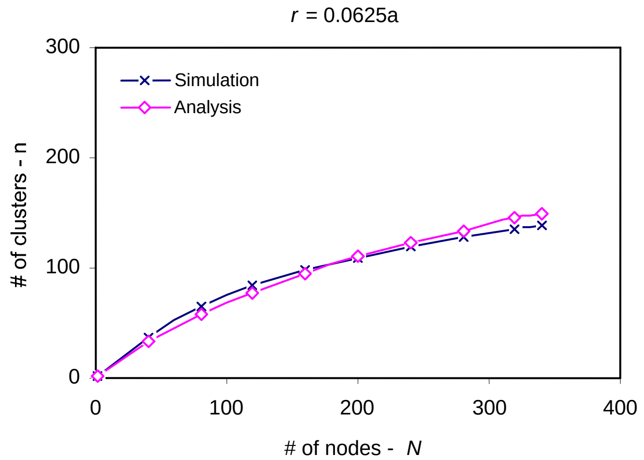
$$P_{HEAD} = \frac{1}{\sqrt{(N-1)\frac{r^2\rho}{N}(\frac{r^2}{2}\frac{\rho}{N}-\frac{8r}{3}\sqrt{\frac{\rho}{N}}+\pi)+1}} \tag{18}$$

5.2 P_{HEAD} Verification

We used GloMoSim [17] to develop simulations to verify our analysis. Figure 10 compares the numerical results from the analysis against the simulation results.



(a). The # of clusters varies with network size *N*



(b).The # of clusters varies with transmission range *r*

Figure 10: Verification of cluster size analysis

Figure 10(a) shows how the number of cluster changes with network size given fixed transmission range and network area while Figure 10(b) shows how the number of cluster changes with transmission range given fixed network size. We observe slight difference between the analysis and simulation plots, and also, they cross each other around the middle. This discrepancy arises due to the simplifying assumptions made in our analysis. For example, in a network with a finite number of nodes, it is not with equal probability that a randomly selected node is the *i*-th smallest ($1 \leq i \leq d+1$) in a closed neighborhood, unless the number of nodes in the network is infinite. Our analysis becomes more accurate as the network size increases. Nevertheless, the analysis is sufficiently close to the simulation results, and demonstrates that our theoretical P_{HEAD} adequately models how the *cluster-head* density changes with node transmission range and network size.

6. Relevance and Usage of Control Overhead Analysis

In above sections, we analytically derived the control overhead for a general one-hop clustered MANET. This control overhead analysis captures the effects of node velocity, transmission range, network size, and network density. We then derived the analytical expression for a key parameter, P_{HEAD}, of a typical clustering algorithm, namely, LID.

6.1 Control overhead w.r.t. network environment and clustering algorithm

We have derived three types of control message overheads in clustering which can be used to evaluate how control overhead changes with the network environment for a particular clustering algorithm. As the results show, the control overhead arising from *HELLO*, and *ROUTE* messages increase with r, v, and ρ. While the volume of *CLUSTER* messages increases with v and ρ. In addition, the control overhead is also influenced by P_{HEAD}, a variable which relates to the percentage of *cluster-heads* in the network. Taking the example of LID clustering in a network with N nodes, we note that P_{HEAD} for LID is a decreasing function of ρ and r given a fixed network size. Intuitively, the more nodes there are in the transmission range of a particular node, the less likely it is to become a *cluster-head*. According to our analytical lower bound for the three types of control overhead, we could derive their lower bounds in Knuth Ω-notation [2] with r, ρ and v as variables. On an infinitely large area, i.e. $a\to\infty$ and $N\to\infty$, all the three types of control messages are $\Omega(1)$ with N. The *HELLO* message overhead ultimately increases at the rate of $\Omega(r)$, $\Omega(\rho)$ and $\Omega(v)$ with r, ρ and v respectively, while the *CLUSTER* message overhead at each node is $\Omega(1)$ with r, $\Omega(\rho^{1/2})$ with ρ, and $\Omega(v)$ with v, and the *ROUTE* message at each node increases at $\Omega(r)$ with r, $\Omega(\rho)$ with ρ, and $\Omega(v)$ with v. *ROUTE* message overhead constitutes the main control overhead, because of its relative high broadcasting rate and large message size.

6.2 Scalability of one-hop clustered network

The control overhead in bits/sec is $O_{hello} + O_{cluster} + O_{routing}$. Given that each node has the same channel bandwidth W, under an optimal circumstance, per node throughput for a static network is $\Theta(\frac{W}{\sqrt{n}})$ [1]. The upper bound of per node throughput is $\sqrt{\frac{8}{\pi}}\frac{W}{\Delta}\sqrt{N}$ for all spatial and temporal scheduling schemes, where

Δ caters for a protocol specified guard zone to minimize contention and interference [1]. Thus the upper bound of the network size is constraint by the equation:

$$O_{hello} + O_{cluster} + O_{routing} \leq \sqrt{\frac{8}{\pi}} \frac{W}{\Delta} \frac{1}{\sqrt{N}}$$

In [6], it is shown that node mobility could actually help to increase throughput at each node, and per node throughput of Θ(1) is achievable at the expense of additional delay.

6.3 Reducing control overhead

We have derived lower bound for control overhead, but it is based on our assumptions that all three types of control messages are needed and the frequency of these messages are chosen such that a route change or a cluster change can always be detected immediately. These requirements can be relaxed in order to reduce the clustering overhead further. One likely approach to reduce control overhead is to limit the explicit sending of messages. For example, in Passive Clustering [24], sending of explicit *CLUSTER* messages is avoided by monitoring data packets with some predefined cluster information embedded within. The cluster infrastructure is in fact constructed as a by-product of user traffic. Another possibility is to reduce the frequency of the control message transmission. When a node does not frequently send or receive data, it is not necessary for every change in route and cluster status be updated in the node. In fact, a node needs to exchange and update its clustering and routing information when it has data to transmit or receive. Therefore, in networks that not all nodes have busy traffic, there is a change of reducing frequency of control messages.

7. Conclusion

While many clustering algorithms have been proposed for mobile ad hoc networks, formal mathematical analysis of the overheads incurred by clustering has been extremely lacking. Much of the analysis is based on the big-O notation and relates to the network size only. This is very inadequate as various other network parameters affect the volume of control overhead generated, e.g. node mobility, node transmission range, and network density. Cluster overheads compete with the data traffic for the network bandwidth, and if not managed properly, can adversely affect the network's performance. In this paper, we have analyzed the clustering overheads for a generic one-hop clustering algorithm taking into consideration node mobility, node transmission range, network size, and network density. By abstracting and representing the clustering algorithm as the probability of an arbitrarily selected node being a *cluster-head*, we also showed how to derive this value for a typical clustering algorithm, the Lowest ID algorithm. Our analysis also makes provision for three key aspects of clustering, viz., the neighbor discovery using periodic broadcast of *HELLO* messages, the clustering management and the routing information management. The work presented here provides a good basis for further analysis on the performance of clustering algorithms for mobile ad hoc networks, in aspects such as scalability and the influence of node mobility patterns.

REFERENCES

[1] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks", *IEEE Trans on Information Theory*, Vol. 46, No. 2, Mar 2000, pp. 388-404.

[2] D. E. Knuth, "Big Omicron and big Omega and big Theta", *ACM SIGACT News*, Vol. 8, No. 2, Apr-Jun 1976, pp. 18-24.

[3] J. Sucec and I. Marsic, "Clustering Overhead for Hierarchical Routing in Mobile Ad Hoc Networks", in *Proc. of INFOCOM,* New York, NY, June 2002, pp. 1698–1706.

[4] J. Sucec and I. Marsic, "Hierarchical Routing Overhead in Mobile Ad Hoc Networks", *IEEE Transactions on Mobile Computing,* Vol. 3, No. 1, Jan-Mar 2004, pp. 46–56.

[5] L. Kleinrock and F. Kamoun, "Hierarchical Routing for Large Networks: Performance Evaluation and Optimization," *Computer Networks (Elsevier)*, Vol. 1, No. 1, 1997, pp. 155–174.

[6] M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad-hoc Wireless Networks", in *Proc. of IEEE INFOCOM*, 2001, pp. 1360–1369.

[7] P. Y.-Z. Chen, A. L. Liestman and J. Liu, "Clustering Algorithms for Ad Hoc Wireless Networks", *Ad Hoc and Sensor Networks*, edited by Y. Xiao and Y. Pan, Nova Science Publisher, 2004.

[8] C. Perkins and P. Bhagwat. "Highly Dynamic Destination Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", in *Proc. of ACM SIGCOMM*, October 1994.

[9] C. E. Perkins, and E. M. Royer, "Ad-Hoc On-Demand Distance Vector Routing", in *Proc. of IEEE WMCSA 1999*, New Orleans, LA, Feb. 1999.

[10] S. Cho and J.P. Hayes, "Impact of mobility on connection stability in ad hoc networks", in *Proc. of Wireless Communication and Networking Conference (WCNC),* New Orleans, LA, USA, March 2005.

[11] T. Camp, J. Boleng, and V. Davies, "Survey of mobility models for ad hoc network research", *Wireless Communication and Mobile Computing (WCMC)*, vol. 2, 2002, pp. 483-502.

[12] C. Bettstetter, G. Resta, and P. Santi, "The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks", *IEEE Transactions on Mobile Computing*, Vol. 2, No. 3, pp. 257-269, Jul-Sep 2003.

[13] Lenard E. Miller, "Distribution of Link Distances in a Wireless Network", *Journal of Research of the National Institute of Standards and Technology*, Vol. 106, No.2, pp. 401-412, March-April 2001.

[14] C.-C. Chiang, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel", in *Proc. of IEEE SICON*, Apr 1997, pp. 197-211.

[15] M. Gerla and J.T.-C.Tsai, "Multicluster, mobile, multimedia radio network", *ACM/Baltzer Journal of Wireless Networks*, Vol. 1, No. 3, 1995, pp. 244-265.

[16] C.R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 7, Sep 1997, pp. 1265-1275.

[17] X. Zeng, R. Bagrodia and M. Gerla, "GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks", in *Proc. of 12th Workshop on Parallel and Distributed Simulations* (PADS '98), Banff, Alberta, Canada, 1998. Available from: http://pcl.cs.ucla.edu/projects/glomosim/.

[18] D. J. Baker and A. Ephremides, "The architectural organization of a mobile radio network via a distributed algorithm", *IEEE Trans. on Communications.*, Vol. 29, No. 11, Nov 1981, pp. 1694-1701.

[19] C. Bettstetter and S. König, "On the Message and Time Complexity of a Distributed Mobility-Adaptive Clustering Algorithm in Wireless Ad Hoc Networks", in *Proc. of 4th European Wireless*, Florence, Italy, Feb 25-28, 2002, pp. 128-134.
[20] I. I. Er and Winston K. G. Seah, 'Clustering Overhead and Convergence Time Analysis of the Mobility-based Multi-Hop Clustering Algorithm for Mobile Ad Hoc Networks', *Proceedings of the 1st International Workshop on Performance Modeling in Wired, Wireless, Mobile Networking and Computing* (PMW2MNC05), in conjunction with 11th International Conference on Parallel and Distributed Systems (ICPADS-2005), Jul 20-22, 2005.
[21] S. Basagni, "Distributed Clustering for Ad Hoc Networks", in *Proc. of the International Symposium on Parallel Architectures, Algorithms and Networks* (ISPAN'99), Jun 23–25, 1999, Washington, DC, USA.
[22] I. I. Er, and Winston K. G. Seah, "Mobility-based d-Hop Clustering Algorithm for Mobile Ad Hoc Networks", in *Proceedings of IEEE Wireless Communications and Networking Conference*, Mar 21-25, 2004, Atlanta, Georgia, USA.
[23] A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh, "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks", in *Proc. of IEEE INFOCOM*, Mar 1999.
[24] T. J. Kwon, and M. Gerla, "Efficient flooding with Passive Clustering (PC) in ad hoc networks", *SIGCOMM Computer Communications Review*, Vol. 32, No. 1, Jan 2002, pp. 44-56.

<u>Appendix</u> – Derivation of f_{routing}

Firstly, the total number links between nodes in the same cluster is derived as follows:

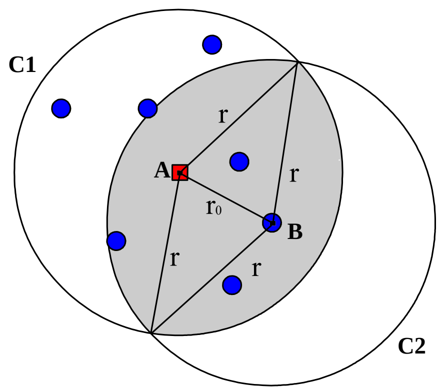


Figure 11: Links between an arbitrary member node, B, with other member nodes in the same cluster.

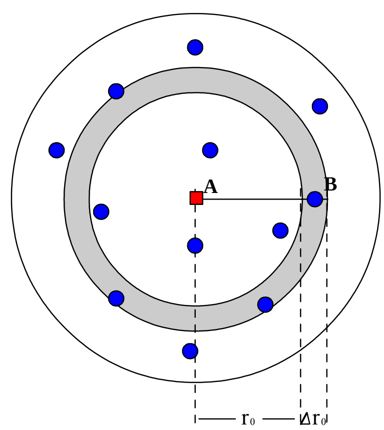


Figure 12: Derivation of the total number of links between members of a cluster

In Figure 11, node A is the *cluster-head* of cluster A and other nodes are the *cluster-members* of cluster A, with an arbitrary *cluster-member* B that is distance r_0 from node A. There are m-1 members in cluster A excluding the *cluster-head* itself, and they are uniformly distributed within the transmission range of A denoted by the circular region C1; similarly, C2 is the transmission range of B. The shaded region represents the intersection of C1 and C2, and B is connected to all the *cluster-members* of A in the shaded region. The number of *cluster-members* of cluster A in the shaded region is proportional to the area of the shaded region, which is a function of r_0 and given by:

$$f(r_0) = 4r^2 \cdot \arccos\frac{r_0}{2r} - r \cdot \sqrt{r^2 - \frac{1}{4}{r_0}^2}$$

Thus, the average number of *cluster-members* connected with a *cluster-member* at distance r_0 from A is:

$$\omega = \frac{4r^2 \cdot \arccos\frac{r_0}{2r} - r \cdot \sqrt{r^2 - \frac{1}{4}{r_0}^2}}{\pi r^2}(m-1) \qquad (19)$$

In Figure 12, a belt region that is distance r_0 from *cluster-head* with width Δr is shaded. When $\Delta r \to 0$, all the *cluster-members* in the belt region are distance r_0 from the *cluster-head*; thus each have ω connected *cluster-members*. Due to the uniform distribution, the number of *cluster-members* residing in the belt region is proportional to the area of the belt region, which is $2\pi r_0 \Delta r$ as $\Delta r \to 0$, and the number of *cluster-members* of cluster A in the belt region is:

$$\mu = \frac{2\pi r_0 \Delta r}{\pi r^2}(m-1) \qquad (20)$$

Observe that the circular region C1 covered by the transmission range of A can be divided into infinitely many belt regions of width Δr and their distance from A ranges from 0 to r. Each belt region contributes $\omega \cdot \mu$ amount of links with other *cluster-members* in A. Thus, the number of links in cluster A connecting *cluster-members* in A can be derived as follows:

$$\begin{aligned}\frac{1}{2}\int_0^r \frac{2\pi r_0}{\pi r^2}(m-1)\omega dr_0 &= \frac{1}{2}\int_0^r \frac{2\pi r_0}{\pi r^2}(m-1)\cdot\frac{4r^2 \cdot \arccos\frac{r_0}{2r} - r \cdot \sqrt{r^2 - \frac{1}{4}{r_0}^2}}{\pi r^2}(m-1)dr_0 \\ &= \frac{(m-1)^2}{\pi r^3}\int_0^r r_0(4r \cdot \arccos\frac{r_0}{2r} - \sqrt{r^2 - \frac{1}{4}{r_0}^2})dr_0 \\ &= (m-1)^2(-\frac{3\sqrt{3}}{8\pi} + \frac{1}{2})\end{aligned} \qquad (21)$$

We do not include the links between *cluster-members* and *cluster-head* in the computation, since the *cluster-head* always connected to all the *cluster-members* in its cluster and the total number of links within a cluster is computed by adding *m*-1 to the result of Eqn (21), which is $(m-1)^2(-\frac{3\sqrt{3}}{8\pi} + \frac{1}{2}) + (m-1)$. Since there are n clusters in the network, and the total links in the entire network is $\frac{Nd}{2}$ as explained in subsection 3.5, the link change rate among nodes within the same cluster at each node is:

$$\frac{2n((m-1)^2(-\frac{3\sqrt{3}}{8\pi}+\frac{1}{2})+(m-1))}{Nd}\lambda$$

By replacing n with $N\mathrm{P}_{\mathrm{HEAD}}$, m with $\frac{1}{\mathrm{P}_{\mathrm{HEAD}}}$ and λ with $\frac{16dv}{\pi^2 r}$, we get:

$$\frac{32v(1-\mathrm{P}_{\mathrm{HEAD}})((1-\mathrm{P}_{\mathrm{HEAD}})(-\frac{3\sqrt{3}}{8\pi}+\frac{1}{2})+\mathrm{P}_{\mathrm{HEAD}})}{\pi^2 r\mathrm{P}_{\mathrm{HEAD}}}$$

Because each link change within the cluster causes a round of cluster-wide routing update, i.e., each link change within a cluster causes each node within the cluster to broadcast routing information. Thus, the frequency of incurring packet transmission overheads p_{route} at each node should be at least the rate of link change in the cluster, which is m times that of the rate of link change within the cluster at each node. Thus,

$$f_{routing}=\frac{32v(1-\mathrm{P}_{\mathrm{HEAD}})((1-\mathrm{P}_{\mathrm{HEAD}})(-\frac{3\sqrt{3}}{8\pi}+\frac{1}{2})+\mathrm{P}_{\mathrm{HEAD}})}{\pi^2 r\mathrm{P}_{\mathrm{HEAD}}}\cdot m$$

$$\Rightarrow f_{routing}=\frac{32v(1-\mathrm{P}_{\mathrm{HEAD}})((1-\mathrm{P}_{\mathrm{HEAD}})(-\frac{3\sqrt{3}}{8\pi}+\frac{1}{2})+\mathrm{P}_{\mathrm{HEAD}})}{\pi^2 r{\mathrm{P}_{\mathrm{HEAD}}}^2}$$

In addition, p_{route} represents a route entry to a node in the cluster and the routing information broadcasted at each node is proportional to the size of the cluster in steady state and the actual size of routing information broadcasted at each node is mp_{route}. Therefore, the control overhead due to routing information broadcasting:

$$O_{routing}=f_{routing}mp_{route}=p_{route}\frac{32v(1-\mathrm{P}_{\mathrm{HEAD}})((1-\mathrm{P}_{\mathrm{HEAD}})(-\frac{3\sqrt{3}}{8\pi}+\frac{1}{2})+\mathrm{P}_{\mathrm{HEAD}})}{\pi^2 r{\mathrm{P}_{\mathrm{HEAD}}}^3}$$