Topology Stability Analysis and its Application in Hierarchical Mobile Ad Hoc Networks

Yi Xu and Wenye Wang
Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27606
Email: {yxu2, wwang}@ncsu.edu

***Abstract*— The hierarchical architecture has been proven effective for solving the scalability problems in large-scale ad hoc networks. The stability of the hierarchical architecture is a key factor in determining the network performance. Although many solutions have been proposed to construct stable clusters, the maximum stability achievable in the mobile environments is still unknown. In this paper, we define three metrics to measure the network stability: the cluster lifetime, the inter-cluster link lifetime, and the end-to-end path lifetime. We model and analyze the maximum of these lifetimes under the constraint of random node mobility. The analytical results provide the fundamental understanding of the bounds on network stability. Inspired by this understanding, we propose a clustering algorithm and a hierarchical routing protocol that work together to achieve the maximum network stability. The analytical results are verified by simulations.**

I. Introduction

In large-scale ad hoc networks, such as the campus network where a mixture of large number of heterogeneous nodes (e.g., stationary devices, pedestrians, vehicles, etc) are present, the flat architecture faces scalability problems in which the traffic throughput, packet delay and network overhead deteriorate very fast as the network size increases. As an effective solution to the scalability problems, the hierarchical architecture is constructed which groups geographically close nodes into *clusters* and connects the clusters into a higher-tier overlay network. A representative node selected from each cluster, named as clusterhead, coordinates the communications inside and between clusters. By organizing the large-scale network into different logical tiers, the network resources can be allocated and utilized efficiently, thus improving the network performance.

However, since the nodes in ad hoc networks are movable, their mobility poses a big challenge to the stability of network topology, which in turn incurs negative impact on the network performance. When a node moves in a hierarchical network, it may be attached to different clusters at different times, resulting in frequent path rediscovery each time it changes the point of attachment. The inter-cluster connectivity affects the path stability too. When an inter-cluster link fails, all the communication paths traversing the broken link have to be replaced. The frequent communication interruptions significantly jeopardize the performance of the network, causing low throughput, long delay, and high overhead. Ideally, the stability of the clusters and their connections should be maximized in order to optimize the network performance.

In this paper, we define three metrics to quantify the network stability in the hierarchical architecture: the cluster lifetime, the inter-cluster link lifetime, and the end-to-end path lifetime. These three correlated metrics measure different stability aspects of the hierarchical structure. The cluster lifetime indicates how often the nodes change their cluster memberships, the inter-cluster link lifetime assesses how long neighbor clusters remain connected, and the path lifetime evaluates how stable an end-to-end communication path can be. It is obvious that long lifetime implies stable architecture and good communication performance. Hence, the objective of optimizing the network performance translates into maximizing these three stability-indicating lifetimes.

There has been abundance of work done in the literature that constructs hierarchical network architecture, among which some have considered node mobility in order to establish stable clusters [1]–[6] and some have proposed strategies to minimize the cluster changes [2], [7], [8]. All these efforts are effective to improve the cluster stability. However, since these schemes are all heuristic based, they do not reveal the longest possible cluster lifetime in mobile environments. Besides, though the link and path lifetime has been investigated in flat networks [9]–[12], little has been done towards understanding the inter-cluster link lifetime and the cluster-based path lifetime in hierarchical networks. Understanding the maximum lifetimes of the clusters, the inter-cluster links and the end-to-end paths will greatly advance our knowledge on the performance bounds of hierarchical architectures and, in addition, provide valuable guidance for the network architecture design.

We study in this paper the impact of random node mobility on the architectural stability of hierarchical networks. Specifically, we investigate the *maximum* of the cluster lifetime, the inter-cluster link lifetime and the path lifetime in a given mobility environment, through both analysis and simulations. We then apply the stability results to the network architecture design and propose a new clustering algorithm as well as a new cluster-based routing protocol to achieve the maximum stability. The proposed algorithm and protocol can be used in large-scale mobile ad hoc networks, such as those established on pedestrians and vehicles moving in a variety of speeds in a large area, to optimize the communication quality and the network resource utilization.

The rest of this paper is organized as follows. Section II discusses the related work on the clustering algorithms, the hierarchical routing protocols, as well as the link and path properties in flat networks. In Section III we present the lifetime analysis of the clusters, the inter-cluster links and the cluster-based paths using a Random-Walk-like mobility model. Section IV and V propose a new clustering algorithm and a new routing protocol respectively, which work in conjunction to stabilize the hierarchical architecture to the largest extent. Their performance is evaluated by simulation in Section VI. Finally, Section VII concludes this paper.

II. Related Work

A. Clustering Algorithms

The clustering algorithms construct clusters by determining the clusterheads and their affiliated member nodes. The basic method to appoint clusterheads is to use some form of node characteristic values, like the identification number, node degree, and node preference. The Lowest-ID algorithm [14] selects the nodes with the lowest IDs in their respective neighborhood to become the clusterheads. The Highest Connectivity algorithm [15] assigns the clusterhead role to those nodes with the most neighbors. In the Weight-Based algorithm [16], preference is used such that the most willing-to-serve nodes become clusterheads. The common problem with these algorithms is their frequent cluster changes in mobile environments. Besides simplicity, other factors like energy conservation [17], cluster size bounding [18] and overhead minimization [19] also come into consideration in the cluster construction process. However, as node mobility is not considered, these algorithms cannot maintain stable cluster structure either when the nodes are mobile.

In the mobility considered algorithms, clusters are constructed with the goal of maximizing the cluster lifetime in spite of node movement. This is done through selecting the low mobility nodes to serve as clusterheads, because the low mobility nodes are expected to stay in their clusters longer than the high mobility nodes. Among these algorithms, An and Papavassiliou measure the relative node speed [1], Chatterjee *et al.* combine various factors including node speed, cluster size and power consumption [4], Bao and Garcia-Luna-Aceves considere node speed and energy [5], Sivavakeesar and Pavlou predict node mobility pattern [6], and McDonald and Znati estimate the node mutual reachability within a cluster by using a probability measure [3].

While the abovementioned algorithms attempt to stabilize clusters at the time of cluster construction, other strategies extend the cluster lifetime after cluster construction. Chiang *et al.* propose in [7] the Least Cluster Change concept in which re-clustering takes place only when two clusterheads move into contact or a clustermember has lost contact to its clusterhead. It reduces cluster changes significantly compared to the previous periodical re-clustering schemes. Basu *et al.* define a timer named Cluster Contention Interval to avoid the transient cluster changes caused by temporary clusterhead contact [2]. Ghosh and Basagni relax the re-clustering criterion in the GDMAC algorithm [8] such that up to K clusterheads are allowed to be neighbors and a clustermember switches to a newly arrived clusterhead only if the weight of the new clusterhead exceeds its current clusterhead by a quantity H.

Although the consideration of mobility and the avoidance of excessive cluster changes have enhanced the cluster stability, all the discussed work has followed heuristic approaches that do not reveal the *maximum* stability attainable in mobile networks.

B. Hierarchical Routing Protocols

The hierarchical routing protocols establish communication paths and transport packets in the clustered networks. Different from flat networks, the path selection decisions are made at the clusterheads in hierarchical networks, where the clustermembers participate only in a local region. Pei *et al.* design in [20] the LANMAR routing protocol for the networks in which the nodes assume group mobility. The CGSR protocol [7] proposed by Chiang *et al.* assumes the existence of a global cluster membership table that helps locate the destination node in order to deliver a packet. Similar membership table is also used in [21]. In the CBRP [22] proposed by Jiang *et al.* and the CEDAR [23] proposed by Sivakumar *et al.* the clusterheads form a routing backbone which guides the path discovery process. In [24] Belding uses the local topology information to maximize the neighbor cluster connectivity and to stabilize the paths. All these protocols can successfully set up data communication paths in hierarchical networks, but none of them has provided an in-depth stability analysis of the links and paths established by the routing protocols.

C. Link and Path Properties

The link and path properties have been studied intensely in recent years. Gerharz *et al.* show in [9] that the expected residual inter-node link lifetime varies with the link age in the Random Waypoint, Gauss-Markov and Manhattan Grid mobility models. Sadagopan *et al.* demonstrate in [10] that the inter-node link durations have multi-modal distributions in the Reference Point Group Mobility and the Freeway mobility models when node speeds are low, but the paths longer than two hops exhibit exponentially distributed lifetimes with moderate to high node speeds. Samar and Wicker present detailed mathematical formulation and analysis of the inter-node link properties in [11] using a mobility model similar to the Random Direction [13]. Lenders *et al.* carry out experimental study on the link and path lifetimes [12]. These existing efforts have provided valuable insights into the mobility impact on network topology and performance, but they all focus on flat networks. There is still insufficient knowledge on the stability of the links and paths in hierarchical networks, in which the composition of links and paths is complicated.

III. Lifetime Analysis of the Hierarchical Architecture

The heuristic based clustering algorithms and routing protocols seen in the literature do not provide theoretical modeling and analysis of the architectural stability which, if available, will enable a better understanding of the communication performance in hierarchical networks. Because the links and paths in hierarchical architecture have more complicated compositions, the previous analytical work in flat networks cannot be applied directly. As such, we model and analyze in this section the topology stability of the hierarchical mobile networks, in terms of the cluster, link and path lifetimes. There are two main factors that affect the network topology stability: node mobility and energy. We investigate the mobility factor in the lifetime modeling and analysis in this paper while leave the energy factor to our future study, because the mobility incurred network topology changes are more often (on the order of minutes) than the energy incurred changes (on the order of hours). Besides, the independence of the two factors also allows us to study them separately.

A. Mobility Model

Since different mobilities have different degrees of impact on the network topology, we specify the following flexible node mobility model that can be tuned easily to simulate various mobility scenarios. It is similar to the Random Walk [25], but with some modifications to represent more realistic moving behaviors. In this model, a node alternates between the moving and the pausing phases. It selects a random direction from $[0, 2\pi)$ and a random speed from $[v_{min}, v_{max}]$ at the beginning of a moving phase, which is the same as in the Random Walk. But unlike the constant movement duration or distance, we allow a node to choose its travel distance, which is a random variable uniformly distributed in $[0, d_{max}]$. If the node hits the network boundary before having finished its planned travel distance, it bounces back into the network to finish the travel distance. Upon arrival at the destination, the node pauses for a random time before starting another movement. The pause (stationary) time is exponentially distributed with mean τ_s. By tuning the moving speed and the pause time, we are able to study the network stability with various node mobilities.

B. Network Model

We assume the network model as follows. The network consists of N nodes, which are uniformly distributed over an area of $l \times l$ square meters initially. Every node moves independently and obeys the mobility model defined above. The mobility model maintains uniform node spacial distribution over time. The radio transmission radius is r meters for every node. When two nodes are within r meters distance from each other, they are able to communicate directly; otherwise direct communication is not possible. Because we focus on node mobility, we ignore the link disruptions due to wireless signal interferences and obstructions. As such, link existence is solely determined by the inter-node distance. The *cluster* is defined to be a group of geographically close nodes in which one node acts as the clusterhead and the others are clustermembers. Every clustermember is a neighbor of its clusterhead. Communication paths are established and packets are forwarded through the overlay network composed by the clusterheads. Fig. 1 illustrates an example of the hierarchical networks.

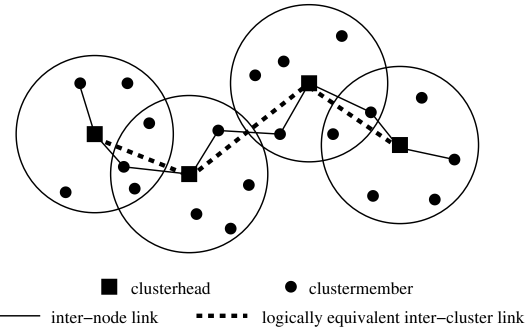


Fig. 1. The clustered structure and the communication path.

C. Cluster Lifetime

We measure a cluster's stability by defining the cluster lifetime which is the duration of its clusterhead staying in the head status without interruption. The cluster lifetime starts when its clusterhead is appointed and ends when the clusterhead switches to a clustermember. Clearly, the various clustering algorithms reconstruct clusters per different requirements, resulting in different ending times of a clusterhead. For example, the Least Cluster Change requires re-clustering when two clusterheads come into contact [7], the Cluster Contention Interval delays re-clustering for a short time [2], and the (K, H) thresholds limit the occurance of re-clustering [8]. We are interested here in the *longest* cluster lifetime allowed in a given mobility environment, which takes place when a clusterhead undertakes its role until all of its affiliated clustermembers have moved away. In this case, a coming clusterhead does not force an existing valid cluster to reconstruct. At the time that a clusterhead does not have any affiliated clustermembers, the clusterhead initiates re-clustering and tries to join another cluster as a clustermember. Therefore, the cluster lifetime in this case equates the time for the clusterhead to become alone. Though it is possible that a clusterhead insists in keeping an empty cluster, the time for it to become alone still provides a good approximation of the real cluster lifetime, because it demonstrates that by carefully designing the clustering algorithm we will at least be able to achieve the cluster lifetime as good as this approximation.

1) Cluster Membership Time: Since the cluster lifetime is determined by the time when the last clustermember leaves the clusterhead, it is easier to look at a clustermember's membership duration first. The *longest* membership duration takes place when the clustermember is affiliated to its clusterhead all the time until they move to r meters apart. It is determined by their neighboring time. We denote the longest membership time as the random variable T_m and discuss it in three cases corresponding to the initial mobility phases of the clusterhead and the clustermember at the time of cluster formation:

1) both nodes are stationary;
2) one node is stationary and the other is moving;
3) both nodes are moving.

Note that these three cases describe only the initial mobility status of the clusterhead and the clustermember when their cluster is just constructed. After that the mobility status of the two nodes may change over time.

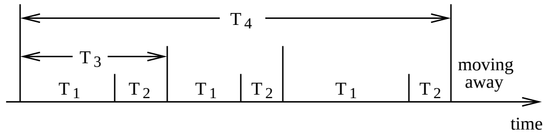


Fig. 2. The intervals of pauses and movements.

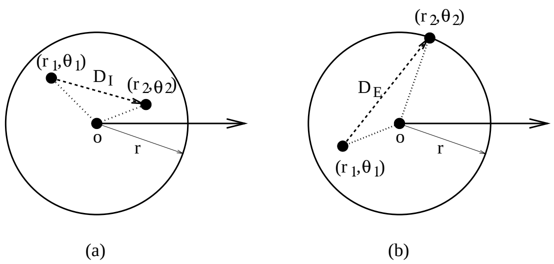


Fig. 3. The movements inside and beyond the coverage area of another node. The moving node moves from the location (r_1, θ_1) to the location (r_2, θ_2) in polar coordinates.

In the first case, for the ease of analysis we first assume that one node is fixed to its location such that it never moves. With this node fixed, the other node pauses and moves in its vicinity. At the end of a pausing phase, if the movable node chooses a destination that is still reachable from the fixed node, the two nodes will stay in contact for an extended duration. T_m in this case consists of a sequence of alternated intervals of pauses and movements until the movable node finally chooses a destination out of the reach of the fixed node. Fig. 2 describes the interval sequence, in which T_1 and T_2 represent the random durations in a pausing phase and in a moving phase respectively. T_1 is exponentially distributed with the mean $E(T_1) = \tau_s$ as specified in the mobility model. The mean of T_2 is determined as $E(T_2) = \tau_{1,I} = E(\frac{D_I}{V}) = E(D_I)E(\frac{1}{V})$, where $\tau_{1,I}$ denotes the mean time of *one* node moving *inside* the transmission area of the fixed node, D_I is the random travel distance *inside* the transmission area, V is the random speed, and D_I and V are assumed to be independent. We illustrate such a movement in Fig. 3(a), in which the fixed node is located at the origin. The mean travel distance $E(D_I)$ is computed as

$$\begin{aligned} E(D_I) &= \frac{1}{(\pi r^2)^2}\int_0^{2\pi}\!\!\int_0^{r}\!\!\int_0^{2\pi}\!\!\int_0^{r} d(r_1,\theta_1,r_2,\theta_2)\, r_1 \mathrm{d}r_1 \mathrm{d}\theta_1\, r_2 \mathrm{d}r_2 \mathrm{d}\theta_2 \\ &\approx \frac{1}{(\pi r^2)^2}\sum_{\{r_1,\theta_1,r_2,\theta_2\}} d(r_1,\theta_1,r_2,\theta_2)\, r_1 (\frac{r}{k})(\frac{2\pi}{k}) r_2 (\frac{r}{k})(\frac{2\pi}{k}) \\ &= \frac{4}{k^4 r^2}\sum_{\{r_1,\theta_1,r_2,\theta_2\}} d(r_1,\theta_1,r_2,\theta_2)\, r_1 r_2, \end{aligned} \tag{1}$$

where

$$d(r_1,\theta_1,r_2,\theta_2) = \sqrt{(r_1\cos\theta_1 - r_2\cos\theta_2)^2 + (r_1\sin\theta_1 - r_2\sin\theta_2)^2}. \tag{2}$$

Because there is no closed-form solution for the integral in (1), we have used the numerical approximation of dividing the integration range of each variable into k fragments and summing the integration results over all the fragments. The value of k is determined in a way such that the approximation error is negligibly small. We use the following criterion to choose k: if the difference between two approximations, one using k fragments and the other using $2k$ fragments, is less than 10^{-3} times the approximation using k fragments, the value of k is good enough. We use $k = 1000$ in this paper. The speed V is uniformly distributed and it is not difficult to compute the mean of its inverse as

$$E(\frac{1}{V}) = \int_{v_{min}}^{v_{max}} \frac{1}{v}\cdot\frac{1}{v_{max} - v_{min}}\,\mathrm{d}v = \frac{\ln(v_{max}) - \ln(v_{min})}{v_{max} - v_{min}}. \tag{3}$$

We define the sum of T_1 and T_2 as another random variable T_3. For mathematical manageability, the distribution of T_3 is approximated by an exponential distribution, since one of the components T_1 is exponentially distributed. Its mean is $E(T_3) = \tau_s + \tau_{1,I}$. When the movable node starts to move to a new location in the moving phase, the new location is randomly selected inside a disk of radius d_{max} centered at the movable node. Inside this disk, there is a smaller disk of radius r centered at the fixed node, which is the reachable area of the fixed node. By assuming $d_{max} > 2r$ and ignoring the boundary effect such that the two nodes are not very close to the network boundary, we have the probability of the movable node to travel out of the reach of the fixed node as $P_O = 1 - \frac{\pi r^2}{\pi d_{max}^2} = \frac{d_{max}^2 - r^2}{d_{max}^2}$ in each moving phase. The total neighboring time T_4 is then an exponentially distributed random variable with the mean $E(T_4) = \frac{E(T_3)}{P_O}$. Then we remove the assumption of one node fixed to consider the independent movements of both nodes. Let us denote the two nodes as A and B, and denote the neighboring time when A is fixed as $T_{4,A}$ and the neighboring time when B is fixed as $T_{4,B}$. We have $T_m = \min(T_{4,A}, T_{4,B})$ and determine its mean as

$$E_{s,s}(T_m) = \frac{E(T_4)}{2} = \frac{\tau_s + \tau_{1,I}}{2P_O}, \tag{4}$$

where $E_{s,s}(T_m)$ denotes the mean in the case of both nodes being initially stationary.

In the second case, one node is stationary and the other is moving initially. With probability $P_I = 1 - P_O = \frac{r^2}{d_{max}^2}$, the moving node stops inside the transmission area of the stationary node. The mean of this duration is $\tau_{1,I}$. After the moving node has stopped, the rest neighboring time is exactly the same as what we have discussed in the case 1. The mean duration is $E_{s,s}(T_m)$. With probability P_O, the moving node has a destination outside the coverage of the stationary node. Let us denote the time before the two nodes become r meters apart as T_5, its mean is determined by $E(T_5) = \tau_{1,E} = E(\frac{D_E}{V}) = E(D_E)E(\frac{1}{V})$, where $\tau_{1,E}$ denotes the mean time of *one* node moving to the *edge* of the transmission area of the stationary node, and D_E and V are assumed to be independent. The random travel distance to the *edge*, denoted as D_E, is shown in Fig. 3(b). Its mean is determined by

$$\begin{aligned} E(D_E) &= \frac{1}{2\pi\cdot\pi r^2}\int_0^{2\pi}\!\!\int_0^{2\pi}\!\!\int_0^{r} d(r_1,\theta_1,\theta_2)\, r_1 \mathrm{d}r_1 \mathrm{d}\theta_1\, \mathrm{d}\theta_2, \\ &\approx \frac{1}{2\pi\cdot\pi r^2}\sum_{\{r_1,\theta_1,r_2\}} d(r_1,\theta_1,\theta_2)\, r_1 (\frac{r}{k})(\frac{2\pi}{k})(\frac{2\pi}{k}) \\ &= \frac{2}{k^3 r}\sum_{\{r_1,\theta_1,\theta_2\}} d(r_1,\theta_1,\theta_2)\, r_1, \end{aligned} \tag{5}$$

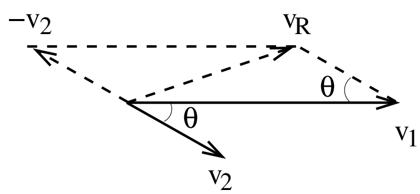


Fig. 4. The relative speed V_R of two nodes with speeds v_1, v_2 and direction difference θ.

where

$$d(r_1,\theta_1,\theta_2) = \sqrt{(r_1\cos\theta_1 - r\cos\theta_2)^2 + (r_1\sin\theta_1 - r\sin\theta_2)^2}. \quad (6)$$

The integral in (5) does not have a closed-form solution, so we have used a numerical approximation similar to (1). Summarizing both possibilities, we have the mean membership time in the case of one node being initially stationary and the other being initially moving as

$$E_{s,m}(T_m) = P_I(\tau_{1,I} + E_{s,s}(T_m)) + P_O\tau_{1,E}. \quad (7)$$

In the third case, both nodes are moving initially. With probability P_I, they stop within each other's transmission area. Denoting the time to one node stopping as T_6, we have $E(T_6) = \tau_{2,I} = E(\frac{D_I}{V_R}) = E(D_I)E(\frac{1}{V_R})$, where $\tau_{2,I}$ denotes the mean time of *two* nodes moving *inside* each other's transmission area, V_R is their relative speed, and D_I and V_R are assumed to be independent. Fig. 4 depicts the formation of V_R, from which we obtain

$$\begin{aligned} E(\frac{1}{V_R}) &= \frac{1}{\pi(v_{max}-v_{min})^2}\int_{v_{min}}^{v_{max}}\int_{v_{min}}^{v_{max}}\int_0^{\pi}\frac{1}{V_R}\,\mathrm{d}\theta\mathrm{d}v_1\mathrm{d}v_2 \\ &\approx \frac{1}{\pi(v_{max}-v_{min})^2}\sum_{\{\theta,v_1,v_2\}}\frac{1}{V_R}\,(\frac{\pi}{k})(\frac{v_{max}-v_{min}}{k})^2 \\ &= \frac{1}{k^3}\sum_{\{\theta,v_1,v_2\}}\frac{1}{V_R}, \end{aligned} \quad (8)$$

where

$$V_R = \sqrt{v_1^2 + v_2^2 - 2v_1v_2\cos\theta}. \quad (9)$$

We have used in (8) the numerical computation for the integral as before. The mean of the random travel distance $E(D_I)$ is obtained from (1). After one node has come to stop, the time to the other node stopping is described by T_2 and we have $E(T_2) = \tau_{1,I}$ as before. After both nodes become stationary, the rest of their neighboring time is again the case 1 that we have discussed earlier, which has the mean duration of $E_{s,s}(T_m)$. With probability P_O, the two initially moving nodes will move apart. There are two possibilities of their mobility phases at the time of being apart: both are moving or one is moving and the other is stationary. Let us denote the respective probabilities as γ and $1-\gamma$. In the former case, we denote the random neighboring time as T_7 and obtain its mean as $E(T_7) = \tau_{2,E} = E(\frac{D_E}{V_R}) = E(D_E)E(\frac{1}{V_R})$, where $\tau_{2,E}$ denotes the mean time of *two* nodes moving to the *edge* of each other's transmission area, and D_E and V_R are assumed to be independent. $E(D_E)$ and $E(\frac{1}{V_R})$ are obtained from (5) and (8) respectively. In the latter case, one node stops inside the transmission area of the other node first and then the other node continues to move away. The time to one node stopping is described by the random variable T_6 with the mean $\tau_{2,I}$ and the time to the other node moving away is described by T_5 with the mean $\tau_{1,E}$ as we have discussed before. γ is determined as follows. We first determine the probability that a node stops before they are apart as

$$\begin{aligned} Q &= P\Big\{\frac{W\cdot X}{V} < \tau_{2,E}\Big\} \\ &= \iiint_{\frac{w\cdot x}{v}<\tau_{2,E}} \frac{1}{d_{max}(v_{max}-v_{min})}\mathrm{d}w\mathrm{d}x\mathrm{d}v \\ &\approx \frac{1}{k^3}\sum_{\{w,x,v\}} 1_{\{\frac{w\cdot x}{v}<\tau_{2,E}\}}(w,x,v) \end{aligned} \quad (10)$$

where W is uniformly distributed in $[0,1]$, X is the random travel distance, $W\cdot X$ is the residual travel distance at the time of cluster formation, and $1_{\{\frac{w\cdot x}{v}<\tau_{2,E}\}}(w,x,v)$ is an indicator function. Numerical approximation is used to compute Q. Then the probability of both nodes moving at the time they are apart is $(1-Q)^2$ and the probability of one moving and one stationary is $2Q(1-Q)$. Normalizing them we obtain

$$\gamma = \frac{(1-Q)^2}{(1-Q)^2 + 2Q(1-Q)},\ 1-\gamma = \frac{2Q(1-Q)}{(1-Q)^2 + 2Q(1-Q)}. \quad (11)$$

Considering both probabilities of P_I and P_O, the mean membership time is determined as

$$\begin{aligned} E_{m,m}(T_m) &= P_I(\tau_{2,I} + \tau_{1,I} + E_{s,s}(T_m)) \\ &\quad + P_O(\gamma\tau_{2,E} + (1-\gamma)(\tau_{2,I} + \tau_{1,E})) \end{aligned} \quad (12)$$

where $E_{m,m}(T_m)$ denotes the mean of T_m in the case that both nodes are moving initially.

Summarizing all the three cases, the mean cluster membership time can be written as

$$E(T_m) = P_{s,s}E_{s,s}(T_m) + P_{s,m}E_{s,m}(T_m) + P_{m,m}E_{m,m}(T_m), \quad (13)$$

where $P_{s,s}$, $P_{s,m}$, $P_{m,m}$ are the probabilities of the respective cases. We know from the mobility model that the mean duration is τ_s in the pausing phase and $\tau_m = E(\frac{D}{V}) = E(D)E(\frac{1}{V}) = \frac{d_{max}}{2}E(\frac{1}{V})$ in the moving phase. Therefore, the probabilities of a node's mobility phases are obtained as

$$P_s = \frac{\tau_s}{\tau_s + \tau_m}, \qquad P_m = \frac{\tau_m}{\tau_s + \tau_m}, \quad (14)$$

from which we further have

$$P_{s,s} = P_s^2, \qquad P_{s,m} = 2P_sP_m, \qquad P_{m,m} = P_m^2. \quad (15)$$

2) Cluster Lifetime: The cluster lifetime is the time duration before all the clustermembers leave the clusterhead. Let T_h denote the random cluster lifetime. We use a Markov model to study its mean value. The Markov transition diagram is depicted in Fig. 5, in which a state is defined to be the number of clustermembers affiliated to the clusterhead. We assume the clustermembers come and leave in Poisson processes. The transitions among the states are then viewed as a birth-death process. Each time a node joins the cluster, the state transits one step to the right; each time a clustermember leaves the cluster, the state transits one step to the left. At

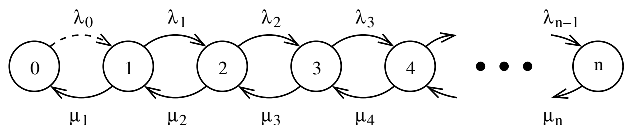


Fig. 5. The Markov transition diagram. The initial state S_{init} is determined by the number of initially affiliated clustermembers and on average $init = \frac{N_m}{N_h}$. The final state is always S_0.

time 0, the transition starts at an initial state S_{init}, which denotes the number of clustermembers at the time the cluster is constructed. The cluster lifetime is the transition time from state S_{init} to S_0.

Since on state S_0 the clusterhead will re-cluster to merge into another cluster, the clustermember joining rate $\lambda_0 = 0$. The joining rate λ_j on the other states S_j is state dependent with finite node population. For any two states S_{j_1} and S_{j_2}, $\lambda_{j_1} > \lambda_{j_2}$ if $j_1 < j_2$. To simplify the model, we truncate the Markov chain at the state S_n, where $n \ll N$. In this truncated model $\lambda_j = \lambda$ $(1 \le j \le n-1)$. We observe from simulations that the cluster size always fluctuates within a few multiples of its initial value, so we set $n = 5 \times init$. λ is determined as follows. Let N_m and N_h denote the total number of clustermembers and clusterheads in the network respectively. We have

$$\lambda_j = \lambda = \Big(\frac{N_m}{E(T_m)} + \frac{N_h}{E(T_h)}\Big) \cdot \frac{\pi r^2}{l^2} \cdot \beta_1 \cdot \beta_2 \quad (1 \le j \le n-1), \tag{16}$$

where $\frac{N_m}{E(T_m)} + \frac{N_h}{E(T_h)}$ accounts for the networkwide arrival rate of nodes seeking a cluster to join, $\frac{\pi r^2}{l^2}$ is the geographical factor considering the percentage that takes place in the clusterhead's transmission area, β_1 is the mobility factor, and β_2 is the clusterhead selection factor. β_1 is concerned with the fact that a node is able to join the clusterhead only when they have relative movement: a relative stationary node inside the clusterhead's neighborhood is likely to be an affiliated clustermember already and a relative stationary node out of the reach of the clusterhead does not have the chance to join this clusterhead. β_2 considers the chance for a node to join a particular clusterhead given the possibilities that it may see several candidate clusterheads in its neighborhood and it may also become a clusterhead itself. We determine them as

$$\beta_1 = P_m + P_s P_m, \qquad \beta_2 = \frac{1}{\frac{\pi r^2}{l^2} N_h + 1}. \tag{17}$$

The clustermember departure rate is also state dependent and expressed as

$$\mu_j = j\mu, \qquad \mu = \frac{1}{E(T_m)} \qquad (1 \le j \le n). \tag{18}$$

The initial state S_{init} has $\frac{N_m}{N_h}$ number of clustermembers on average.

The difficulty in finding $E(T_h)$ rises from the fact that there exists infinite number of possible paths from S_{init} to S_0. We solve this problem as follows. Defining t_j $(0 \le j \le n)$ as the mean transition time from S_j to S_0, we have

$$\begin{cases} t_0 = 0, \\ t_j = \frac{1}{\lambda_j+\mu_j} + \frac{\mu_j}{\lambda_j+\mu_j} t_{j-1} + \frac{\lambda_j}{\lambda_j+\mu_j} t_{j+1} \quad (1 \le j \le n-1), \\ t_n = \frac{1}{\mu_n} + t_{n-1}. \end{cases} \tag{19}$$

The transition time from a state S_j to S_0 is the sum of the sojourn time on state S_j and the transition time from the successive state (S_{j-1} or S_{j+1}) to S_0. The sojourn time is determined by when a joining node arrives or an affiliated clustermember leaves. Since both are Poisson processes, the combination is still a Poisson process with parameter $\lambda_j + \mu_j$. Therefore the mean sojourn time on S_j is $\frac{1}{\lambda_j+\mu_j}$. After this sojourn time, a left transition occurs with probability $\frac{\mu_j}{\lambda_j+\mu_j}$ or a right transition occurs with probability $\frac{\lambda_j}{\lambda_j+\mu_j}$. At the boundary state S_n, only the left transition is possible as there are no arrivals of joining nodes.

We reformat (19) into the matrix form. Considering (16) and (18), we difine

$$\mathbf{A} = \begin{pmatrix} \lambda+\mu & -\lambda & & & & \\ -2\mu & \lambda+2\mu & -\lambda & & & \\ & -3\mu & \lambda+3\mu & -\lambda & & \\ & & & \ddots & & \\ & & & -(n-1)\mu & \lambda+(n-1)\mu & -\lambda \\ & & & & -n\mu & n\mu \end{pmatrix},$$

$$\mathbf{T} = \begin{pmatrix} t_1 & t_2 & t_3 & \cdots & t_{n-1} & t_n \end{pmatrix}^T,$$

$$\mathbf{E} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \end{pmatrix}^T.$$

Then we have the following equation

$$\mathbf{A} \cdot \mathbf{T} = \mathbf{E},$$

which gives the solution of $\mathbf{T}$ as

$$\mathbf{T} = \mathbf{A}^{-1}\mathbf{E}, \tag{20}$$

and $E(T_h) = t_{init}$. Note that (16) requires the knowledge of $E(T_h)$. In the computations, $E(T_h)$ is first assigned an initial value and then computed recursively until it converges.

D. Inter-Cluster Link Lifetime

The inter-cluster connectivity is determined by the availability of the links that connect two nodes from the two clusters respectively. Such links include those between the two clusterheads, one clusterhead and one clustermember, and two clustermembers. They provide direct or indirect connectivity between the two clusterheads. Fig. 6(a) illustrates all the connection types. We define the inter-cluster *Logical Link* to be the set of all these connections, as shown in Fig. 6(b). The logical link exists as long as *any* of the connection types exists. The *longest* inter-cluster link lifetime is then the lifetime of the logical link.

When a logical link consists of multiple connection types, out of the efficiency consideration, a routing protocol preferably chooses the shortest one to transmit packets. Therefore, if a clustermember can be used in both a 2-hop connection and a

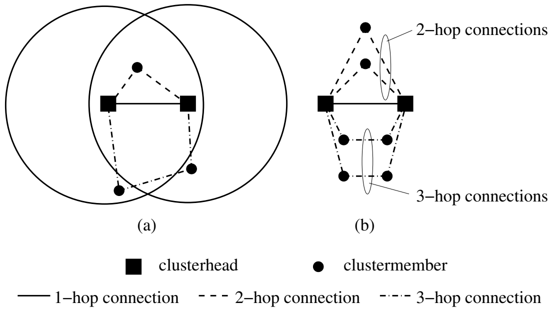


Fig. 6. The connection types and the logical link.

3-hop connection, the 2-hop connection will always be used. This observation allows us to assume that a clustermember appears either in a 2-hop or a 3-hop connection, but not both. The mean cluster membership time $E(T_m)$ derived earlier describes the neighboring time between a clusterhead and a clustermember. It can be generalized as the mean neighboring time between any two neighbor nodes. Thus, following our assumption in Section III-C, the inter-node link lifetimes have identical and exponential distributions with mean $\frac{1}{\mu} = E(T_m)$.

The lifetime of the logical link is determined by the initial composition of the link, the arrivals of new connections and the failures of existing connections. Due to the unmanageable difficulty in determining the new connection arrival process, we approximate the logical link lifetime by the time to the failures of all the initial connections, which serves as a lower estimate of the real lifetime. Let n_i $(i = 1, 2, 3)$ denote the initial number of i-hop connections, $\mathcal{F} = \{F_x^{(j)}\}$ $(j = 1, 2, \cdots, \sum_{i=1}^{3} n_i)$ denote a permutation of the failure sequence of the $\sum_{i=1}^{3} n_i$ connections where $F_x^{(j)} \in \{F_1, F_2, F_3\}$ denotes that the j-th failure happens on an x-hop connection, $n_i^{(j)}$ denote the number of remaining i-hop connections after the $(j-1)$-th but before the j-th failure in the sequence $\mathcal{F}$, and $1_i(F_x^{(j)})$ denote the indicator functions such that

$$1_i(F_x^{(j)}) = \begin{cases} 1 & x = i \\ 0 & x \neq i \end{cases} \quad (i = 1, 2, 3). \tag{21}$$

The mean lifetime of the logical link is written as

$$E(T_l) = \sum_{\{\mathcal{F}\}} P(\mathcal{F}) T(\mathcal{F}), \tag{22}$$

where $P(\mathcal{F})$ is the probability of $\mathcal{F}$, $T(\mathcal{F})$ is the mean lifetime given $\mathcal{F}$, and $\{\mathcal{F}\}$ has cardinality $\binom{n_1+n_2+n_3}{n_1}\binom{n_2+n_3}{n_2}\binom{n_3}{n_3} = \frac{(n_1+n_2+n_3)!}{n_1! n_2! n_3!}$. $P(\mathcal{F})$ and $T(\mathcal{F})$ are determined by

$$P(\mathcal{F}) = \prod_{j=1}^{n_1+n_2+n_3} P(F_x^{(j)}) = \prod_{j=1}^{n_1+n_2+n_3} \frac{\sum_{i=1}^{3} i \cdot n_i^{(j)} \cdot 1_i(F_x^{(j)})}{\sum_{i=1}^{3} i \cdot n_i^{(j)}}, \tag{23}$$

$$T(\mathcal{F}) = \sum_{j=1}^{n_1+n_2+n_3} T(F_x^{(j)}) = \sum_{j=1}^{n_1+n_2+n_3} \frac{1}{\sum_{i=1}^{3} i \cdot n_i^{(j)} \cdot \mu}. \tag{24}$$

In (23), $\sum_{i=1}^{3} i \cdot n_i^{(j)}$ is the total number of inter-node links before the j-th connection failure, $\sum_{i=1}^{3} i \cdot n_i^{(j)} \cdot 1_i(F_x^{(j)})$ is the number of inter-node links of which any break will result in $F_x^{(j)}$, their ratio determines $P(F_x^{(j)})$, and $P(\mathcal{F})$ is the multiplication of all the $P(F_x^{(j)})$'s. In (24), $\frac{1}{\sum_{i=1}^{3} i \cdot n_i^{(j)} \cdot \mu}$ is the mean time between the $(j-1)$-th and the j-th connection failures, and $T(\mathcal{F})$ is the sum of all these intervals.

TABLE I

MOBILITY PATTERNS

Pattern	τ_s (min)	$[v_{min}, v_{max}]$ (m/s)	d_{max} (m)
MP1	8	$[1, 3]$	1000
MP2	6	$[1, 5]$	1000
MP3	4	$[1, 7]$	1000
MP4	2	$[1, 9]$	1000

E. Path Lifetime

A path in the hierarchical network architecture consists of a sequence of inter-cluster links. Considering the clustering roles of the two end nodes, there are also possible intra-cluster links at the two ends of the path. If the node at one end of the path is a clustermember, there must be a clustermember–clusterhead link at that end of the path. An example path is illustrated in Fig. 1. The lifetime of the path is determined by the shortest lifetime among all these links. Assuming that the inter-cluster and the intra-cluster links all have exponentially distributed lifetimes with mean $E(T_l)$ and $E(T_m)$ respectively, the *longest* path lifetime is then also exponentially distributed with the mean determined by

$$E(T_p) = P_{h\text{-}h} E_{h\text{-}h}(T_p) + P_{h\text{-}m} E_{h\text{-}m}(T_p) + P_{m\text{-}m} E_{m\text{-}m}(T_p),$$
$$P_{h\text{-}h} = (\frac{N_h}{N})^2, \; P_{h\text{-}m} = \frac{2N_h N_m}{N^2}, \; P_{m\text{-}m} = (\frac{N_m}{N})^2,$$
$$E_{h\text{-}h}(T_p) = \frac{1}{c\mu_l}, \; E_{h\text{-}m}(T_p) = \frac{1}{c\mu_l + \mu}, \; E_{m\text{-}m}(T_p) = \frac{1}{c\mu_l + 2\mu}, \tag{25}$$

where $P_{h\text{-}h}$, $P_{h\text{-}m}$, $P_{m\text{-}m}$ are the probabilities of the end node roles, $E_{h\text{-}h}(T_p)$, $E_{h\text{-}m}(T_p)$, $E_{m\text{-}m}(T_p)$ are the mean path lifetimes in the respective cases, c is the number of inter-cluster links in the path, $\mu_l = \frac{1}{E(T_l)}$, and $\mu = \frac{1}{E(T_m)}$.

F. Numerical Results

Due to problem complexity, accurate analytical determination of the parameters N_h, N_m, n_i $(i = 1, 2, 3)$ and c is difficult. We use simulations to obtain their average values and then apply them in the analysis. We configure an example network as $N = 240$, $l = 2000$ meters, $r = 250$ meters, and specify four mobility patterns by tuning the node moving speed and pause time as shown in Table I to represent different mobility scenarios, such as low speed pedestrians and high speed vehicles. Besides the theoretical analysis, we have also implemented the Lowest-ID and the GDMAC ($K = 3$, $H = 32$ as in [8]) clustering algorithms as comparison in the NS-2 simulator [26].

The mean cluster membership times from the analysis and the simulations are shown in Fig. 7. We observe that the Lowest-ID has a quite short cluster membership time as compared to the theoretical bound. The GDMAC improves over the Lowest-ID, but there is still a noticeable gap from

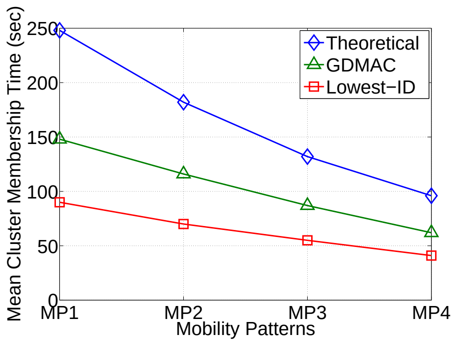


Fig. 7. The mean cluster membership time $E(T_m)$.

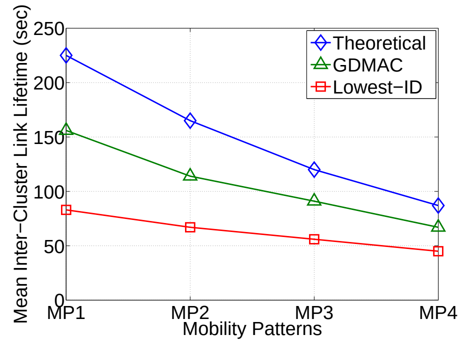


Fig. 9. The mean inter-cluster link lifetime $E(T_l)$.

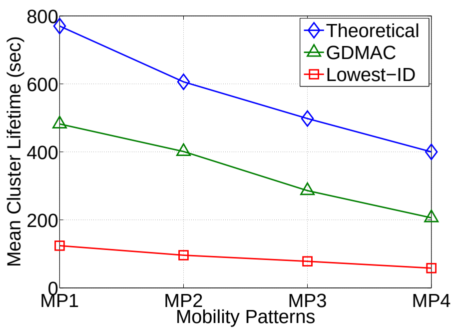


Fig. 8. The mean cluster lifetime $E(T_h)$.

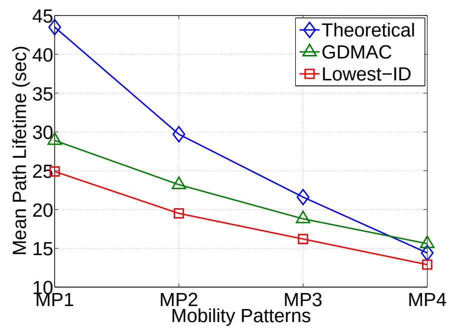


Fig. 10. The mean path lifetime $E(T_p)$.

the upper bound obtained from analysis. Similar observation on the mean cluster lifetime is shown in Fig. 8. We determine N_h and N_m by letting each node stay in its role (clusterhead/clustermember) as long as its cluster is still valid. That is, a clustermember tries to re-cluster only after it has lost the contact to its clusterhead and a clusterhead tries to re-cluster only after it has lost the contact to all of its clustermembers. The average values measured are $N_h = 56$ and $N_m = 184$ across all the four mobility patterns with very slight variations (± 2). The transition time vector $\mathbf{T}$ is then determined by (20) and $t_{init} = t_3$ because S_3 is the state closest to the mean cluster size $\frac{184}{56} = 3.28$. We see in Fig. 8 that the cluster lifetime of the Lowest-ID is significantly lower than the analytical bound and the GDMAC does not reach this bound yet.

With $N_h = 56$ and $N_m = 184$, we measure the average number of connections between neighbor clusterheads and obtain $n_1 = 0.23$ (± 0.03), $n_2 = 1.85$ (± 0.1), $n_3 = 1.05$ (± 0.15) across the four mobility patterns. As they are not integers, we approximate them as:

$$n_1 = \begin{cases} 1 \ w.p.\ 0.23 \\ 0 \ w.p.\ 0.77 \end{cases}, \ n_2 = \begin{cases} 2 \ w.p.\ 0.85 \\ 1 \ w.p.\ 0.15 \end{cases}, \ n_3 = \begin{cases} 2 \ w.p.\ 0.05 \\ 1 \ w.p.\ 0.95 \end{cases}.$$

The mean inter-cluster link lifetime is then determined from (22). Fig. 9 plots the results from analysis and simulations. Again we observe the similar instability of the Lowest-ID in the inter-cluster connectivity and the stability gap between the GDMAC and the analytical bound.

Finally we choose 10 random source-destination pairs and measure their average path length in the number of inter-cluster links with $N_h = 56$ and $N_m = 184$. The measured average path lengths are $c = 3.95, 4.21, 4.21, 4.71$ respectively in the four mobility patterns. The same 10 node pairs are also used in the Lowest-ID and the GDMAC simulations. The analytical mean path lifetime is determined by (25) and plotted in Fig. 10 together with the simulation results. Fig. 10 presents that the path lifetime could be higher than those achieved by the Lowest-ID and the GDMAC. We notice that in the mobility pattern 4 the path lifetimes from the analysis, GDMAC, and Lowest-ID are very close, and GDMAC even exceeds the analytical result. This is due to the fact that our analysis derives a lower estimate of the inter-cluster link lifetime, as mentioned in Section III-D, which leads to the underestimation of the path lifetime. However, this observation indeed indicates that there is limited room for improving the path stability in high mobility environment.

IV. The Mobility and Energy Aware Clustering Algorithm

The stability analysis reveals the longest lifetimes of the clusters, the inter-cluster links and the paths allowed by the node mobilities. It also shows that these longest lifetimes are achieved by extending a node's clustering status to its natural ending. Premature re-clustering terminates the valid clusters and therefore undermines the network topology stability. In this section we design a new clustering algorithm that aims to achieve the longest cluster lifetime. This new algorithm has two features: first, it preserves the existing clusters to the maximum extent; second, it also tries to maximize the cluster lifetime at the time of cluster construction. The first feature

The MEACA Algorithm

```
initialize u.timer
when u.timer expires
   reschedule u.timer
   if      u.status = HEAD and u.member ≠ NULL
      or u.status = MEMBER and u.head ≠ NULL
      then return
   C ← N(u) ∪ {u}
   C ← {v | v.status ≠ MEMBER, v ∈ C}
   A*_m ← α · max{A_m(v) | v ∈ C} (0.9 < α < 1)
   S ← {v | A_m(v) ≥ A*_m, v ∈ C}
   w ← arg_v max{A_e(v) | v ∈ S}
   if w = u
      then u.status ← HEAD
           u.head ← u
   else if w.status = HEAD
      then u.status ← MEMBER
           u.head ← w
   return
```

Fig. 11. The MEACA clustering algorithm.

is straightforward in the design. Once a cluster is constructed, node joining and leaving are allowed, but as long as the cluster is still valid a coming node cannot force the clusterhead to revoke its role or a clustermember to switch to a different cluster. The second feature is implemented as choosing the most stable nodes to become the clusterheads. A node is stable if it is expected to stay in the clusterhead status for long time. We interpret the stableness mainly in the mobility sense but also with the energy consideration. As such, we name the new algorithm the Mobility and Energy Aware Clustering Algorithm (MEACA).

A. The MEACA Algorithm

We define two node attributes for cluster construction. The mobility attribute of a node u is defined to be $A_m = \sum_{v \in N(u)} \tau_v$, where $N(u)$ is the neighbor set of u, τ_v is the neighboring time between a neighbor node v and the node u. The neighboring time is acquired through the use of *hello* messages. Each node periodically broadcasts hello messages to inform the neighbors of its presence. Node u determines the duration that it has been in contact with v by summing up all the continuous time ticks on which it has heard from v. The attribute A_m indicates a node's relative mobility to its neighbors: larger value means higher stableness. The energy attribute A_e of u is defined to be its estimated lasting time of the remaining energy. The nodes exchange their attributes through the hello messages such that a node knows all its neighbors' attributes.

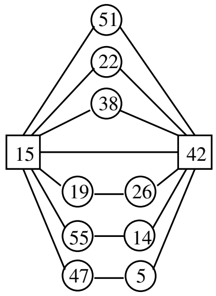
When a node u first joins the network or needs re-clustering, it determines its clustering status by comparing its own attributes with its neighbors' attributes. It finds the node that has high A_m and A_e values to be its clusterhead. If this node is itself, it becomes a clusterhead; otherwise, it becomes a clustermember to join the selected node. Fig. 11 presents the clusterhead selection algorithm in pseudocode. The node u checks periodically if it needs re-clustering. If its status is a clusterhead but there is no affiliated clustermember any more or its status is a clustermember but it has lost contact to its clusterhead, the clustering algorithm is executed. In the first step, u adds itself to the list of nodes from which the clusterhead will be selected. Then it excludes any node that has become a clustermember, because it cannot take the clusterhead status according to our existing cluster preservation requirement. Next, it finds the maximum mobility attribute of the nodes in the list and determines a mobility stableness threshold A_m^*, where α is a constant parameter. The purpose of α is to prevent the selection of a clusterhead that has low energy even though it may have the highest mobility stableness. The value of α adjusts the tradeoff between selecting a high-stability node and selecting a high-energy node to be the clusterhead. If $\alpha = 0$, the algorithm will always choose the highest-energy node in the list as the clusterhead. If $\alpha = 1$, the algorithm will on the contrary always choose the highest-stability node in the list as the clusterhead. However, large α reduces the node status determination delay. As such, we limit α to the range $(0.9, 1)$ for the algorithm convergence considerations. After A_m^* is determined, it shortlists the nodes whose mobility attributes exceed A_m^*. Finally, it finds the node w that has the highest A_e in the shortlist to be its clusterhead. If $w = u$, u becomes a clusterhead; if $w \neq u$, u registers its membership with w. Since w may also be in the clustering process, there are three possibilities. If w has determined to be a clusterhead, w accepts u's registration. If w has determined to be a clustermember, w rejects u's registration and u repeats the clustering algorithm. In the second round as w is excluded from the set C, u will end up with another selection of the clusterhead. If w has not made its decision yet, u waits until w has made its decision.

After u has determined its clustering status, it stays in this status until either of the two following situations takes place: 1) if u is a clusterhead, all of its clustermembers have left the cluster; 2) if u is a clustermember, it has left its selected cluster. In these cases, u determines its new status by running the clustering algorithm again.

B. MEACA Properties

Property 1: The algorithm converges.

The algorithm convergence is the property that a node is able to determine its status in finite time. We prove this property in different cases. *Case 1:* if u determines to be a clusterhead, its status is finalized in one round of executing this algorithm. *Case 2:* if u selects w ($w \neq u$) and w is a clusterhead, u's status as a clustermember is finalized in one round too. *Case 3:* if w is a clustermember, u repeats the clustering process one more time with w excluded from its candidate list. In the worst case u excludes every node in $N(u)$ and ends up with itself. This takes $|N(u) \cup \{u\}|$ rounds. *Case 4:* if w has not made its decision yet but it is not waiting for another node to make decision, then w will determine its status in finite number of rounds according to our discussion on u and afterwards u will finalize in finite number of rounds too. *Case 5:* if w is waiting for w', the



Node 15's Connection Table		
ID	Type	Next Hop
42	1–hop	42
	2–hop	22 38 51
	3–hop	19 47 55

Fig. 12. An example connection table.

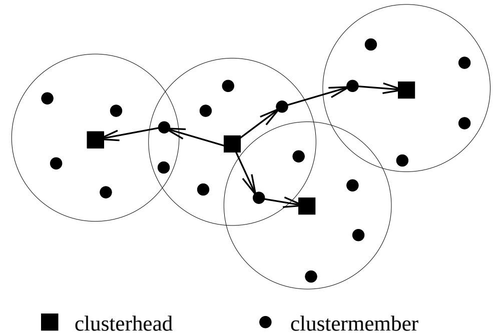


Fig. 13. A route request broadcasting example.

chain of waiting nodes $u, w, w', \cdots$ must be finite given the finite node population in the network and they satisfy the inequality $A_m(u) < A_m(w) < A_m(w') < \cdots$ with high probability when α is large. The last node in this chain will determine its status first and afterwards the rest nodes will determine their status successively in the reverse order of the chain with finite number of clustering rounds each. *Case 6:* if $A_m(u) > A_m(w) > A_m(w') > \cdots$ happens (due to $\alpha < 1$) and the waiting nodes form a loop, node movements will break the loop.

Property 2: The algorithm has no ripple effect.

The ripple effect is the phenomenon that a re-clustering node triggers a sequence of cluster changes in a wide area. In MEACA, a re-clustering node may join an existing clusterhead or construct a new cluster, but it does not force the existing clusterhead to revoke its clusterhead status or an existing clustermember to switch to it. Thus re-clustering is limited to the participating nodes only and the rest of the network is not affected.

Property 3: The algorithm balances node energy.

From a fairness point of view, it is ideal to rotate the clusterhead roles among the nodes because the clusterheads consume energy faster than the clustermembers. In MEACA, the node with the highest energy becomes the clusterhead when several candidate nodes have comparable mobility stableness. A former clusterhead is expected to have lower energy than the former clustermembers and therefore likely to hand over the clusterhead role to others at the time of re-clustering. As the mean cluster lifetime is relatively short (minutes) as compared to the usual node energy lasting time (hours), a clusterhead has plenty of chances to shift its role before depleting its energy.

C. Comparison to the Related Work

The main differences of MEACA from the existing work on clustering algorithms are in two aspects. First, the existing algorithms [2], [7], [8] are *extending* rather than *maximizing* the cluster stability, while MEACA attempts to reach the maximum cluster stability under the constraint of node mobilities. As the analysis suggests, the maximum cluster lifetime is achieved in mobile networks when all the unnecessary cluster changes are avoided. MEACA maximizes the cluster stability by eliminating the premature re-clusterings. Second, when node mobility is considered, the existing schemes [1], [4], [5] measure the node speed first and then infer the network topology stableness, while MEACA directly measures the topology stableness by defining the mobility attribute as the node neighboring times, which is simpler but more accurate.

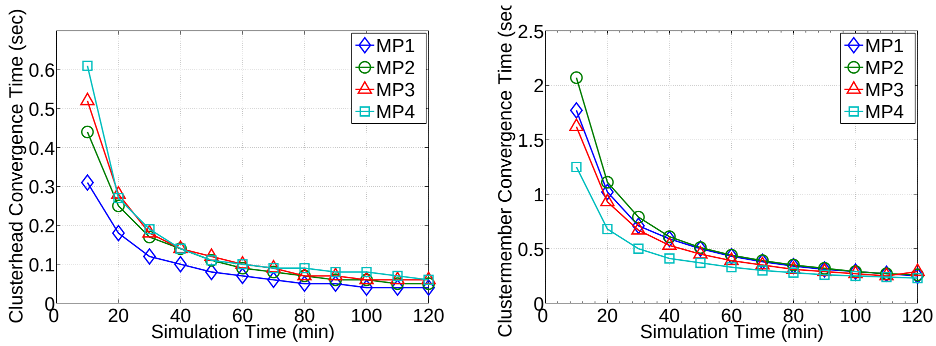
V. The Clustered Network Overlay Routing Protocol

From the analysis we know that in order to achieve the longest path lifetime it is equally important to maximize the inter-cluster connectivity besides stabilizing individual clusters. In this section we propose the Clustered Network Overlay Routing Protocol (CNORP) which aims to achieve two goals: 1) establishing the inter-cluster links and maintaining them to the maximum extent; 2) implementing the path discovery and packet forwarding mechanisms in the overlay network composed of the connected clusterheads.

A. Link Establishment and Maintenance

We have shown in Fig. 6 the three types of connections between two neighbor clusterheads and defined the inter-cluster logical link. The CNORP protocol implements the logical link and keeps it up-to-date. It avoids unnecessary path rediscovery when a connection in the logical link breaks as long as alternative connections are available.

A clusterhead gathers the connection information through *hello* messages. These hello messages are different from those used in the clustering attribute exchanges, though they may be combined. Every node periodically broadcasts hello messages which carry the ID of its clusterhead and the distance from its clusterhead. If the node is a clusterhead, the distance is 0; if it is a clustermember, the distance is 1. A clusterhead that receives the broadcast from its neighbors then knows the 1-hop and 2-hop connections to its neighbor clusterheads: distance 0 in a received hello message indicates a 1-hop connection to the broadcasting clusterhead; distance 1 indicates a 2-hop connection via the broadcasting clustermember. In addition to the hello messages, a clustermember also periodically reports its received hello messages to its clusterhead. These reports provide supplementary connection information: if the clustermember has a 1-hop or a 2-hop connection to a neighbor clusterhead, then its clusterhead may utilize it to reach that neighbor clusterhead in a 2-hop or a 3-hop connection via the reporting clustermember. A clusterhead organizes the connections in a table format as shown in Fig. 12. The connection



(a) Clusterhead

(b) Clustermember

Fig. 14. The node status determination time (MEACA).

table is kept up-to-date as any topology changes will be reflected in the periodical hello messages and clustermember reports. An inter-cluster logical link is broken at the time when all the three connection type entries become empty. When a clusterhead forwards packets to a neighbor clusterhead, it always chooses the shortest available connection in the table. If there are multiple shortest connections, a random selection is made.

A clustermember organizes its connections to the neighbor clusterheads in a similar table. Because a clustermember does not receive reports from other clustermembers, it does not have 3-hop connections. The table is kept up-to-date by the periodically received hello messages. When a clustermember forwards packets for its clusterhead, it uses the shortest available connection to the neighbor clusterhead.

B. Path Discovery and Maintenance

In the overlay network of the connected clusterheads any reactive routing protocol, such as AODV and DSR, can be used for the path discovery and maintenance. Proactive routing protocols are not suitable because of their excessive overhead when the clusters change. During path discovery, route request broadcasting is implemented in a unicast way. If a clusterhead needs to broadcast, it unicasts a copy of the message to each neighbor clusterhead via the shortest connection in the respective logical link. An example of the route request broadcasting is shown in Fig. 13. A path discovered is a sequence of clusterheads from the source node to the destination node. If the source and the destination nodes are clustermembers, the source node's packets are sent to its clusterhead first, then routed to the destination node's clusterhead, and finally delivered to the destination node. An intermediate clusterhead is flexible in choosing a connection to forward a packet to the next clusterhead. The selection is determined by the availability of the connections at the time of forwarding the packet, but the use of the shortest available connection is always enforced.

C. Comparison to the Related Work

The CNORP protocol differs from the existing work in that it does not require a global cluster membership table as

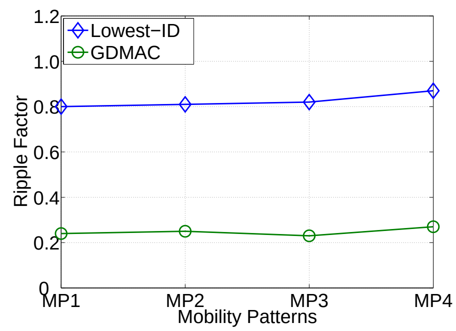


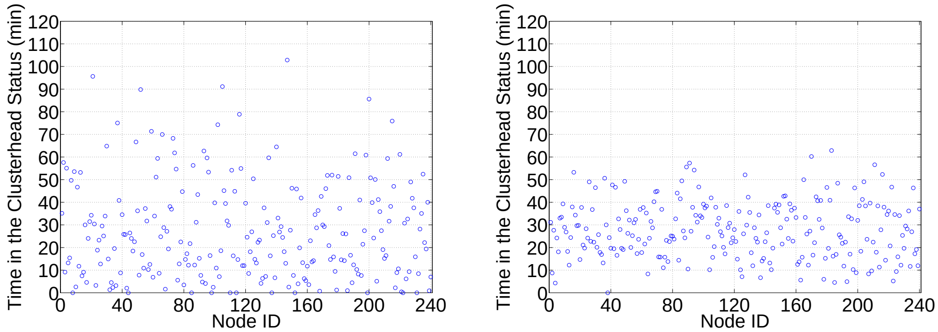
Fig. 15. The ripple effect of Lowest-ID and GDMAC.

compared to [7], [21] and it simplifies the path management and maximizes the path lifetime as compared to [22], [23] by utilizing the inter-cluster logical links. In comparison to [24], a clustermember using the CNORP protocol acquires sufficient local routing information by itself such that it is able to forward packets independently to the neighbor clusterheads without obtaining the routing information from its clusterhead. In addition, the use of unicast to implement the route request broadcasting suppresses the redundant broadcast messages in a simpler way.

VI. Performance Evaluation

We have implemented the MEACA clustering algorithm and the CNORP routing protocol in NS-2 for their performance evaluation. The same simulation settings as in Section III-F are used. We set $\alpha = 0.9$ in the MEACA simulations and the frequency of neighbor node information exchange to be every 2 seconds.

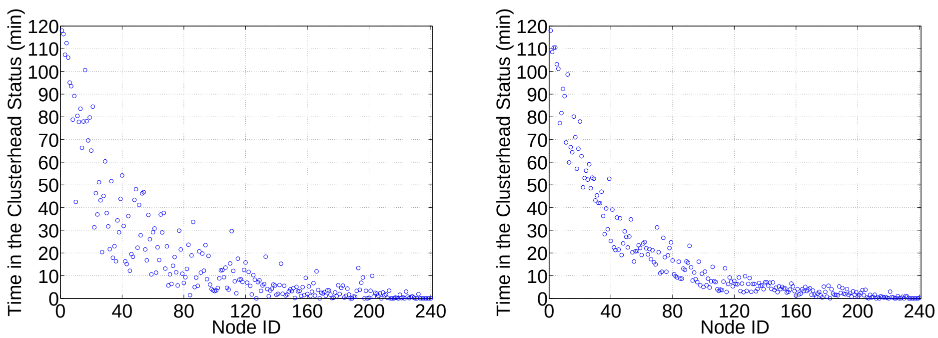
Fig. 14(a) plots the mean time spent by a node in determining its status as a clusterhead using the MEACA algorithm. Initially it takes 0.3–0.6 second for a node to become a clusterhead, because at the beginning of the simulation all the nodes are trying to determine their roles and their decisions are interdependent. Some nodes must wait to determine until their neighbors have finalized. As the simulation proceeds, the mean time decreases to be less than 0.1 second. The



(a) MP1

(b) MP4

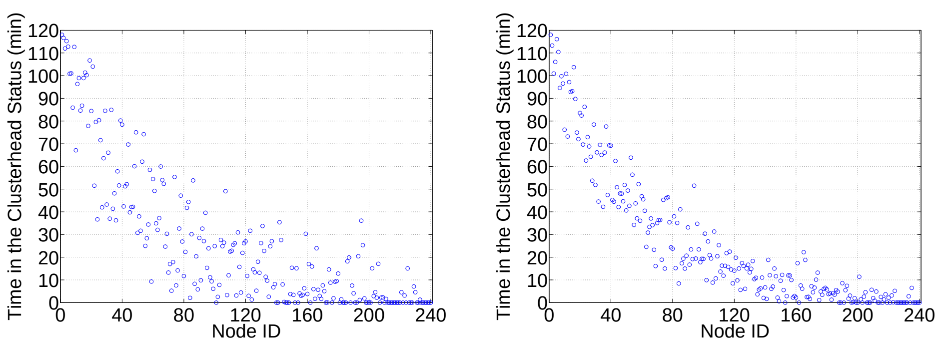
Fig. 16. The time distribution of the clusterhead status (MEACA).



(a) MP1

(b) MP4

Fig. 17. The time distribution of the clusterhead status (Lowest-ID).



(a) MP1

(b) MP4

Fig. 18. The time distribution of the clusterhead status (GDMAC).

similar trend is observed in the mean time to determine the clustermember role, which decreases from 1.3–2 seconds to 0.3 second as shown in Fig. 14(b). These results demonstrate that the MEACA algorithm converges.

We have proven that the MEACA algorithm does not have the ripple effect. As comparison, we plot the ripple effect of the Lowest-ID and the GDMAC in Fig. 15, where the *ripple factor* is defined to be the ratio of the number of prematural re-clustering to the number of matural re-clustering. The Lowest-ID has a ripple factor of 0.8, indicating that each node status update causes on average extra 0.8 update among its neighbors. The GDMAC still has the ripple effect problem though it reduces the ripple factor to 0.2.

Fig. 16, 17, 18 depict the cumulative time of staying

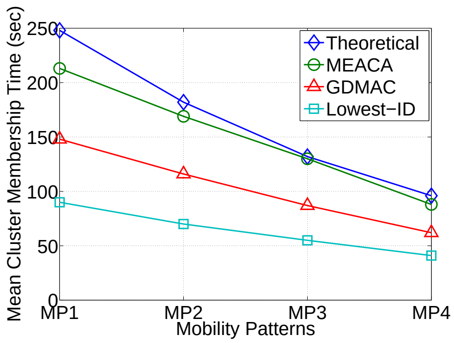


Fig. 19. The mean cluster membership time $E(T_m)$.

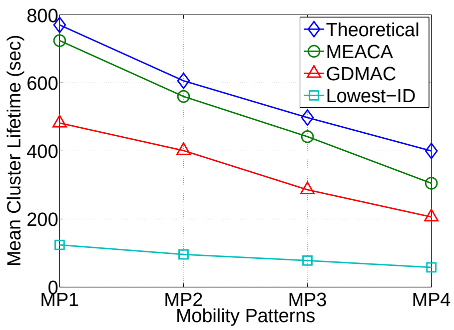


Fig. 20. The mean cluster lifetime $E(T_h)$.

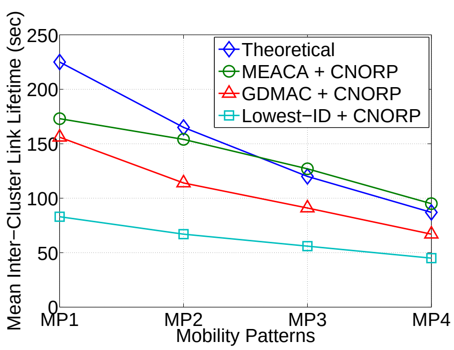


Fig. 21. The mean inter-cluster link lifetime $E(T_l)$.

in the clusterhead status for each node during a 2-hour simulation. We observe obvious dependence of this cumulative time on node ID numbers in the Lowest-ID and the GDMAC algorithms, where the nodes with low IDs take the clusterhead status significantly longer than the others. This phenomenon poses big fairness and energy balancing problems. The MEACA algorithm, on the contrary, does not have the ID-dependence problem. We see in Fig. 16 that both the low-ID and the high-ID nodes have the equal chance to become the clusterheads.

Fig. 19 and 20 compare the cluster membership time and the cluster lifetime achieved by MEACA to the previous results obtained from analysis and the Lowest-ID and GDMAC simulations. We observe that MEACA improves over the Lowest-ID and the GDMAC, and its performance approaches the theoretical bounds. Similar improvements are observed in Fig. 21 and 22 that compare the inter-cluster link lifetime and

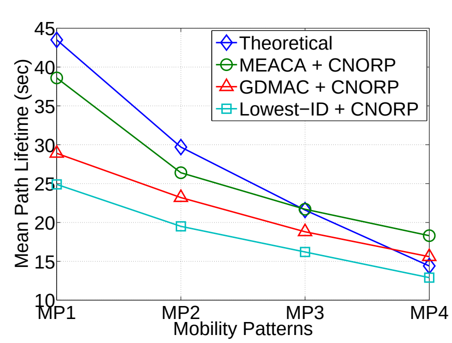


Fig. 22. The mean path lifetime $E(T_p)$.

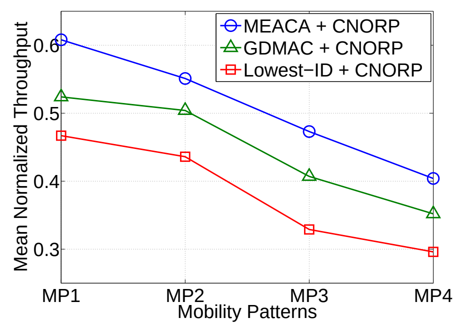


Fig. 23. Throughput comparison.

the path lifetime. We see that the performance of MEACA and CNORP exceeds the theoretical results in the high mobility patterns (MP3 and MP4), which is due to the underestimation of the inter-cluster link lifetime and the path lifetime in the analysis, as mentioned earlier. Finally, Fig. 23 shows the normalized throughout of the 10 random connections using different clustering algorithms. As a result from longer path lifetime, MEACA achieves higher end-to-end throughput than the Lowest-ID and the GDMAC algorithms.

VII. Conclusions

In the hierarchical ad hoc networks consisting of heterogeneous nodes, such as stationary devices, pedestrians and vehicles, the random node mobility impacts the stability of network architecture significantly. The existing efforts in stabilizing the hierarchical architecture do not address the maximum stability problem, which is the foundation to understand the optimal performance of mobile ad hoc networks. We have presented in this paper the mathematical modeling and analysis of the maximum lifetimes of the clusters, the inter-cluster links and the end-to-end paths in a mobile environment described by a Random-Walk-like mobility model. We have found that the maximum stability of the hierarchical architecture is achieved when premature re-clustering is avoided and inter-cluster connectivity is maximized. Enlightened by the analysis, we have designed the MEACA clustering algorithm and the CNORP hierarchical routing protocol that work together to maximize the stability of hierarchical networks. Simulation shows that their performance approaches the theoretical bounds obtained from analysis.

REFERENCES

[1] B. An, and S. Papavassiliou. "A Mobility-Based Clustering Approach to Support Mobility Management and Multicast Routing in Mobile Ad-Hoc Wireless Networks". *International Journal of Network Management*, pages 387–395, 2001.

[2] P. Basu, N. Khan, and T.D.C. Little. "A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks". In *Proc. of IEEE ICDCSW*, pages 413–418, April 2001.

[3] A.B. McDonald, and T.F. Znati. "A Mobility-Based Framework for Adaptive Clustering in Wireless Ad Hoc Networks". *IEEE Journal on Selected Areas in Communications*, 17(8):1466–1487, August 1999.

[4] M. Chatterjee, S.K. Das, and D. Turgut. "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks". *Cluster Computing*, 5, 193–204, 2002.

[5] L. Bao, and J.J. Garcia-Luna-Aceves. "Topology Management in Ad Hoc Networks". In *Proc. of ACM MobiHoc*, pages 129–140, June 2003.

[6] S. Sivavakeesar, and G. Pavlou. "Scalable Location Services for Hierarchically Organized Mobile Ad Hoc Networks". In *Proc. of ACM MobiHoc*, pages 217–228, 2005.

[7] C.-C. Chiang, H.-K. Wu, W. Liu, and M. Gerla. "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel". In *Proc. of IEEE SICON*, pages 197–211, April 1997.

[8] R. Ghosh, and S. Basagni. "Limiting the Impact of Mobility on Ad Hoc Clustering". In *Proc. of ACM PE-WASUN*, pages 197–204, October 2005.

[9] M. Gerharz, C. Waal, M. Frank, and P. Martini. "Link Stability in Mobile Wireless Ad Hoc Networks". In *Proc. of IEEE LCN*, 2002.

[10] N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy. "PATHS: Analysis of PATH Duration Statistics and their Impact on Reactive MANET Routing Protocols". In *Proc. of ACM MobiHoc*, June 2003.

[11] P. Samar, and S.B. Wicker. "On the Behavior of Communication Links of a Node in a Multi-Hop Mobile Environment". In *Proc. of ACM MobiHoc*, May 2004.

[12] V. Lenders, J. Wagner, and M. May. "Analyzing the Impact of Mobility in Ad Hoc Networks". In *Proc. of ACM REALMAN*, May 2006.

[13] E. Royer, P.M. Melliar-Smith, and L. Moser. "An Analysis of the Optimum Node Density for Ad Hoc Mobile Networks". In *Proc. of IEEE ICC*, 2001.

[14] A. Ephremides, J.E. Wieselthier, and D.J. Baker. "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling". In *Proc. of IEEE*, vol. 75, pages 56–73, 1987.

[15] M. Gerla, and J.T-C. Tsai. "Multicluster, Mobile, Multimedia Radio Network". *Wireless Networks*, vol. 1, pages 255–265, October 1995.

[16] S. Basagni. "Distributed Clustering for Ad Hoc Networks". In *Proc. of ISPAN, International Symposium on Parallel Architectures, Algorithms, and Networks*, pages 310–315, June 1999.

[17] V. Kawadia, and P.R. Kumar. "Power Control and Clustering in Ad Hoc Networks". In *Proc. of IEEE Infocom*, pages 459–469, 2003.

[18] R. Ramanathan, and M. Steenstrup. "Hierarchically-Organized Multihop Mobile Wireless Networks for Quality-of-Service Support". *Mobile Networks and Applications*, pages 101–119, 1998.

[19] X. Hong, M. Gerla, Y. Yi, K. Xu, and T.J. Kwon. "Scalable Ad Hoc Routing in Large, Dense Wireless Networks Using Clustering and Landmarks". *Proc. of IEEE ICC*, pages 3179–3185, 2002.

[20] G. Pei, M. Gerla, and X. Hong. "LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility". In *Proc. of ACM MobiHoc*, pages 11–18, 2000.

[21] N. Zhou, and A.A. Abouzeid. "Routing in Ad Hoc Networks: A Theoretical Framework with Practical Implications". In *Proc. of IEEE Infocom*, pages 1240–1251, 2005.

[22] M. Jiang, J. Li, and Y.C. Tay. "Cluster Based Routing Protocol (CBRP)". *IETF MANET Working Group, Internet Draft*, July 1999.

[23] R. Sivakumar, P. Sinha, and V. Bharghavan. "CEDAR: A Core-Extraction Distributed Ad Hoc Routing Algorithm". *IEEE Journal on Selected Areas in Communications*, 17(8):1454–1465, August 1999.

[24] E. Belding-Royer. "Hierarchical Routing in Ad Hoc Mobile Networks". *Wireless Communications and Mobile Computing*, vol. 2, issue 5, pages 515–532, 2002.

[25] T. Camp, J. Boleng, and V. Davies. "A Survey of Mobility Models for Ad Hoc Network Research". *Wireless Communications and Mobile Computing*, vol. 2, issue 5, pages 483–502, 2002.

[26] The Network Simulator NS-2. http://www.isi.edu/nsnam/ns/.