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

Yi Xu, *Student Member, IEEE*, and Wenye Wang, *Member, IEEE*

***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 mobile environments is still unknown. In this paper, we define three metrics for measuring network stability: 1) the cluster lifetime; 2) the intercluster link lifetime; and 3) the end-to-end path lifetime. We model and analyze the maximum of these lifetimes under the constraint of random node mobility. 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.**

***Index Terms*—Clustering, hierarchical architecture, mobility, network topology, wireless ad hoc networks.**

I. Introduction

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

However, nodes in ad hoc networks are movable, so their mobility poses a big challenge to the stability of network topology, which, in turn, incurs a negative impact on the network performance. When a node moves in a hierarchical network, it may be attached to different clusters at different times, which results in frequent path rediscovery each time it changes the point of attachment. Intercluster connectivity also affects path stability. When an intercluster link fails, all the communication paths that traverse the broken link have to be replaced. 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 to optimize the network performance.

In this paper, we define three metrics for quantifying network stability in a hierarchical architecture: 1) the cluster lifetime; 2) the intercluster link lifetime; and 3) the end-to-end path lifetime. These three correlated metrics measure different stability aspects of the hierarchical structure. The cluster lifetime indicates how often nodes change their cluster memberships, the intercluster 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 a long lifetime implies a 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 several work in the literature, which constructs a hierarchical network architecture, some of which have considered node mobility to establish stable clusters [1]–[6], whereas some have proposed strategies to minimize cluster changes [2], [7], [8]. All these efforts are effective for improving cluster stability. However, these schemes are all heuristic based, so they do not reveal the longest possible cluster lifetime in mobile environments. In addition, although the link and path lifetime has been investigated in flat networks [9]–[12], little has been done toward understanding the intercluster link lifetime and the cluster-based path lifetime in hierarchical networks. Understanding the maximum lifetimes of the clusters, the intercluster links, and the end-to-end paths will greatly advance our knowledge of the performance bounds of hierarchical architectures and, in addition, provide valuable guidance for the network architecture design.

In this paper, we study the impact of random node mobility on the architectural stability of hierarchical networks. Specifically, we investigate the *maximum* of the cluster lifetime, the intercluster 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 and a new cluster-based routing protocol to achieve maximum stability. The proposed algorithm and protocol can be used in large-scale mobile ad hoc networks (e.g., on pedestrians and vehicles that move in various speeds in a large area) to optimize the communication quality and network-resource utilization. The rest of this paper is organized as follows. Section II discusses related work on the

Manuscript received August 11, 2007; revised February 14, 2008 and June 2, 2008. First published June 27, 2008; current version published March 17, 2009. The review of this paper was coordinated by Dr. X. Zhang.

The authors are with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27606 (e-mail: yxu2@ncsu.edu; wwang@ncsu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2008.928006

clustering algorithms, the hierarchical routing protocols, and the link and path properties in flat networks. In Section III, we present the lifetime analysis of the clusters, the intercluster links, and the cluster-based paths using a Random-Walk-like mobility model. Sections 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. Section VII concludes this paper.

II. Related Work

A. Clustering Algorithm

Clustering algorithms construct clusters by determining the clusterheads and their affiliated member nodes. The basic method of appointing clusterheads is to use some form of node characteristic values, e.g., 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 nodes with the most neighbors. In the weight-based algorithm [16], preference is used such that nodes that are most willing to serve become clusterheads. The common problem with these algorithms is their frequent cluster changes in mobile environments. Aside from simplicity, other factors like energy conservation [17], cluster size bounding [18], and overhead minimization [19] also come into consideration in cluster construction. However, node mobility is not considered; therefore, these algorithms cannot maintain a stable cluster structure when the nodes are mobile either.

In algorithms that consider mobility, clusters are constructed with the goal of maximizing the cluster lifetime, despite node movement. This method is done by selecting low-mobility nodes to serve as clusterheads, because low-mobility nodes are expected to stay in their clusters longer than 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 consider 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].

Although the aforementioned algorithms attempt to stabilize clusters at cluster construction, other strategies extend the cluster lifetime after cluster construction. Chiang *et al.* propose in [7] the least cluster change concept, in which reclustering takes place only when two clusterheads move into contact, or a clustermember has lost contact to its clusterhead. The reduction of cluster significantly changes compared with previous periodical reclustering schemes. Basu *et al.* define a timer called the cluster contention interval to avoid transient cluster changes that are caused by temporary clusterhead contact [2]. Ghosh and Basagni relax the reclustering criterion in the generalized distributed mobility-adaptive clustering (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 attainable *maximum* stability in mobile networks.

B. Hierarchical Routing Protocols

Hierarchical routing protocols establish communication paths and transport packets in clustered networks. Unlike flat networks, path selection decisions are made at the clusterheads in hierarchical networks, where clustermembers participate only in a local region. Pei *et al.* design [20] the LANMAR routing protocol for networks in which the nodes assume group mobility. The clusterhead gateway switch-routing protocol [7] that was proposed by Chiang *et al.* assumes the existence of a global cluster membership table that helps locate the destination node to deliver a packet. A similar membership table is also used in [21]. In the cluster-based routing protocol [22], as proposed by Jiang *et al.*, and core extraction distributed ad hoc routing [23], as proposed by Sivakumar *et al.*, clusterheads form a routing backbone that guides path discovery. In [24], Belding uses the local topology information to maximize the neighbor cluster connectivity and 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 that routing protocols have established.

C. Link and Path Properties

The link and path properties have been studied intensely in recent years. Gerharz *et al.* show [9] that the expected residual internode link lifetime varies with the link age in the random waypoint, Gauss–Markov, and Manhattan grid mobility models. Sadagopan *et al.* demonstrate [10] that internode link durations have multimodal distributions in the reference point group mobility and freeway mobility models when node speeds are low, but paths that are longer than two hops exhibit exponentially distributed lifetimes with moderate to high node speeds. Samar and Wicker present detailed mathematical formulation and analysis of internode link properties [11] using a mobility model that is similar to random direction [13]. Lenders *et al.* carried out an 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 heuristics-based clustering algorithms and routing protocols 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 a hierarchical architecture have more complicated compositions, the previous analytical work in flat networks cannot be applied directly. As such, in this section, we model and analyze the topology stability of hierarchical mobile networks in terms of the cluster, link, and path lifetimes. There are two main factors that affect the network topology stability: 1) node mobility and 2) energy. We investigate the mobility factor in the lifetime modeling and analysis in this paper and leave the energy factor to our future study, because mobility-incurred network topology changes happen more often (on the order of minutes) than energy-incurred changes (on the order of hours). In addition, the independence of the two factors allows us to study them separately.

A. Mobility Model

Different types of mobility have different degrees of impact on the network topology; therefore, we specify the following flexible node mobility model that can easily be tuned to simulate various mobility scenarios. It is similar to 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, 2π) and a random speed from $[v_{\min}, v_{\max}]$ at the beginning of a moving phase, which is the same as in random walk. However, unlike the constant movement duration or distance, we allow a node to choose its travel distance, which is a random variable that is uniformly distributed in [0, $d_{\max}$]. If the node hits the network boundary before finishing its planned travel distance, it bounces back into the network to finish the travel distance. Upon arriving 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 can study the network stability with various node mobility.

B. Network Model

We assume the network model as follows. The network consists of N nodes, which are initially uniformly distributed over an area of $l \times l$ m^2. Every node independently moves and obeys the aforementioned mobility model. The mobility model maintains uniform node spatial distribution over time. The radio transmission radius is r m for every node. When two nodes are within an r-m distance from each other, they can directly communicate; otherwise, direct communication is not possible. We focus on node mobility; therefore, we ignore link disruptions due to wireless signal interferences and obstructions. As such, link existence is solely determined by the internode distance. A *cluster* is defined as 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 that is composed of the clusterheads. Fig. 1 illustrates an example of hierarchical networks.

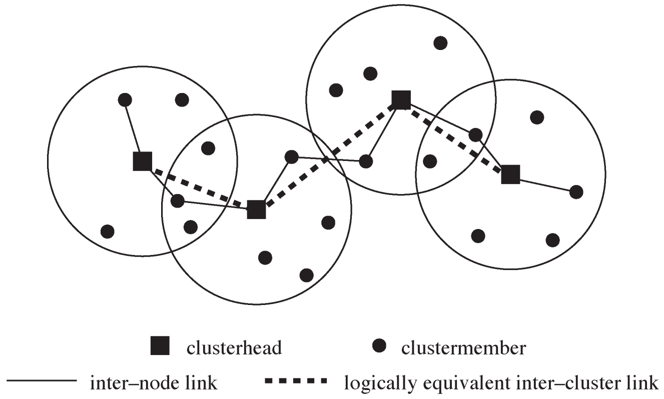


Fig. 1. Clustered structure and the communication path.

C. Cluster Lifetime

We measure a cluster's stability by defining the cluster lifetime, i.e., 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, various clustering algorithms reconstruct clusters per different requirements, resulting in different ending times of a clusterhead. For example, the least amount of cluster change requires reclustering when two clusterheads come into contact [7], the cluster contention interval delays reclustering for a short time [2], and the (K, H) thresholds limit the occurrence of reclustering [8]. We are interested here in the *longest* cluster lifetime that is 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. Once a clusterhead does not have any affiliated clustermembers, the clusterhead initiates reclustering and tries to join another cluster as a clustermember. Therefore, the cluster lifetime in this case equates to the time for the clusterhead to become alone. Although it is possible that a clusterhead insists on keeping an empty cluster, the time that it becomes alone still provides a good approximation of the real cluster lifetime, because it demonstrates that, by carefully designing the clustering algorithm, we can, at least, achieve a cluster lifetime that is as good as this approximation.

1) Cluster Membership Time: The cluster lifetime is determined by the time when the last clustermember leaves the clusterhead; therefore, it is easier to first look at a clustermember's membership duration. The *longest* membership duration takes place when the clustermember is affiliated with its clusterhead the entire time until they move r m 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 that correspond to the initial mobility phases of the clusterhead and the clustermember at cluster formation.

1) Both nodes are stationary.
2) One node is stationary, and the other node 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. Then, the mobility status of the two nodes may change over time.

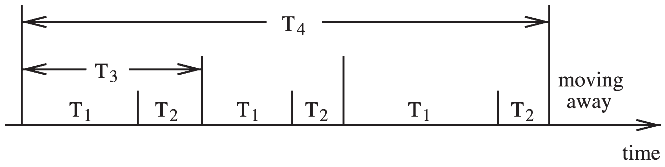
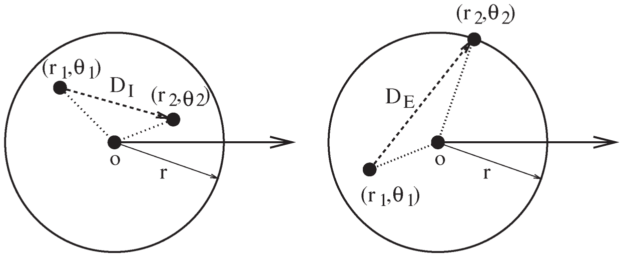


Fig. 2. Intervals of pauses and movements.



(a) (b)

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

In the first case, for ease of analysis, we first assume that one node is fixed to its location such that it never moves. With this node being fixed, the other node pauses and moves into 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. In this case, T_m consists of alternating intervals of pauses and movements until the movable node finally chooses a destination that is out of the fixed node's reach. Fig. 2 describes the interval sequence, in which T_1 and T_2 represent the random durations in a pausing phase and 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 $E(T_2) = \tau_{1,I} = E(D_I/V) = E(D_I)E(1/V)$, where $\tau_{1,I}$ denotes the mean time of *one* node that moves *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

$$\begin{aligned} E(D_I) &= \frac{1}{(\pi r^2)^2} \int_0^{2\pi}\int_0^{r}\int_0^{2\pi}\int_0^{r} d(r_1, \theta_1, r_2, \theta_2) r_1 dr_1 d\theta_1 r_2 dr_2 d\theta_2 \\ &\approx \frac{1}{(\pi r^2)^2} \sum_{\{r_1,\theta_1,r_2,\theta_2\}} d(r_1, \theta_1, r_2, \theta_2) \\ &\quad \times r_1 \left(\frac{r}{k}\right)\left(\frac{2\pi}{k}\right) r_2 \left(\frac{r}{k}\right)\left(\frac{2\pi}{k}\right) \\ &= \frac{4}{k^4 r^2} \sum_{\{r_1,\theta_1,r_2,\theta_2\}} d(r_1, \theta_1, r_2, \theta_2) r_1 r_2 \end{aligned} \tag{1}$$

where

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

There is no closed-form solution for the integral in (1); therefore, we 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 such that the approximation error is negligibly small. We use the following criterion for choosing k: If the difference between two approximations—one approximation uses k fragments, and the other approximation uses $2k$ fragments—is less than 10^{-3} times the approximation that uses 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

$$E\left(\frac{1}{V}\right) = \int_{v_{\min}}^{v_{\max}} \frac{1}{v} \cdot \frac{1}{v_{max} - v_{min}} dv = \frac{\ln(v_{\max}) - \ln(v_{\min})}{v_{\max} - v_{\min}}. \tag{3}$$

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, because 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 $d_{\max}$ that is centered at the movable node. Inside this disk, there is a smaller disk of radius r that is centered at the fixed node, which is the reachable area of the fixed node. By assuming that $d_{\max} > 2r$ and ignoring the boundary effect such that the two nodes are not very close to the network boundary, we have the probability that the movable node travels out of the reach of the fixed node as $P_O = 1 - (\pi r^2/\pi d_{\max}^2) = ((d_{\max}^2 - r^2)/d_{\max}^2)$ in each moving phase. The total neighboring time T_4 is then an exponentially distributed random variable with the mean $E(T_4) = (E(T_3)/P_O)$. Then, we remove the assumption of one node being 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 where both nodes are initially stationary.

In the second case, one node is stationary, and the other node is initially moving. With probability $P_I = 1 - P_O = (r^2/d_{\max}^2)$, 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 of the neighboring time is exactly the same as what we have discussed in 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 m apart as T_5. Its mean is determined by $E(T_5) = \tau_{1,E} = E(D_E/V) = E(D_E)E(1/V)$, where $\tau_{1,E}$ denotes the mean time that *one* node moves to the *edge* of the transmission area of the stationary node, and D_E and V are assumed to be

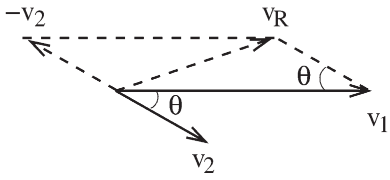


Fig. 4. Relative speed V_R of two nodes with speed v_1, v_2 and direction difference θ.

independent. The random travel distance to the *edge*, which denoted as D_E, is shown in Fig. 3(b). Its mean is determined by

$$\begin{aligned} E(D_E) &= \frac{1}{2\pi \cdot \pi r^2} \int_0^{2\pi}\int_0^{2\pi}\int_0^{r} d(r_1,\theta_1,\theta_2) r_1 dr_1 d\theta_1 d\theta_2, \\ &\approx \frac{1}{2\pi \cdot \pi r^2} \sum_{\{r_1,\theta_1,r_2\}} d(r_1,\theta_1,\theta_2) r_1 \left(\frac{r}{k}\right)\left(\frac{2\pi}{k}\right)\left(\frac{2\pi}{k}\right) \\ &= \frac{2}{k^3 r} \sum_{\{r_1,\theta_1,\theta_2\}} d(r_1,\theta_1,\theta_2) r_1 \end{aligned} \tag{5}$$

where

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

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

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

In the third case, both nodes are initially moving. With probability P_I, they stop within each other's transmission area. Denoting the time that one node stops as T_6, we have $E(T_6) = \tau_{2,I} = E(D_I/V_R) = E(D_I)E(1/V_R)$, where $\tau_{2,I}$ denotes the mean time that *two* nodes move *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

$$\begin{aligned} E\left(\frac{1}{V_R}\right) &= \frac{1}{\pi(v_{\max} - v_{\min})^2} \int_{v_{\min}}^{v_{\max}}\int_{v_{\min}}^{v_{\max}}\int_0^{\pi} \frac{1}{V_R} d\theta dv_1 dv_2 \\ &\approx \frac{1}{\pi(v_{\max} - v_{\min})^2} \sum_{\{\theta,v_1,v_2\}} \frac{1}{V_R}\left(\frac{\pi}{k}\right)\left(\frac{v_{\max} - v_{\min}}{k}\right)^2 \\ &= \frac{1}{k^3} \sum_{\{\theta,v_1,v_2\}} \frac{1}{V_R} \end{aligned} \tag{8}$$

where

$$V_R = \sqrt{v_1^2 + v_2^2 - 2v_1 v_2 \cos\theta}. \tag{9}$$

In (8), we used the same numerical computation for the integral. The mean of the random travel distance $E(D_I)$ is obtained based on (1). After one node has come to stop, the time that the other node stops is described by T_2, and we have $E(T_2) = \tau_{1,I}$. After both nodes have become stationary, the rest of their neighboring time is similar to Case 1, with the mean duration being $E_{s,s}(T_m)$. With probability P_O, the two initially moving nodes will move apart. There are two possibilities for their mobility phases at the time they are apart: 1) Both are moving, and 2) 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 $E(T_7) = \tau_{2,E} = E(D_E/V_R) = E(D_E)E(1/V_R)$, where $\tau_{2,E}$ denotes the mean time that *two* nodes move to the *edge* of each other's transmission area, and D_E and V_R are assumed to be independent. $E(D_E)$ and $E(1/V_R)$ are obtained based on (5) and (8), respectively. In the latter case, one node stops inside the transmission area of the other node, and then, the other node continues to move away. The time it takes for one node stops is described by the random variable T_6 with the mean $\tau_{2,I}$, and the time it takes for the other node to move away is described by T_5 with the mean $\tau_{1,E}$. γ is determined as follows. We first determine the probability that nodes will stop before they are apart as

$$\begin{aligned} Q &= P\left\{\frac{W \cdot X}{V} < \tau_{2,E}\right\} \\ &= \int \int_{\frac{w \cdot x}{v} < \tau_{2,E}} \int \frac{1}{d_{\max}(v_{\max} - v_{\min})} dw dx dv \\ &\approx \frac{1}{k^3} \sum_{\{w,x,v\}} 1_{\left\{\frac{w \cdot x}{v} < \tau_{2,E}\right\}}(w,x,v) \end{aligned} \tag{10}$$

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

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

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

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

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

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

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

where $P_{s,s}$, $P_{s,m}$, and $P_{m,m}$ 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 $\tau_m = E(D/V) = E(D)E(1/V) = (d_{\max}/2)E(1/V)$ in the moving phase.

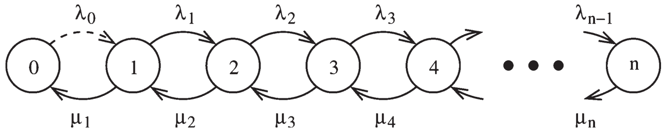


Fig. 5. Markov transition diagram. The initial state S_{init} is determined by the number of initially affiliated clustermembers and, on the average, $init = (N_m/N_h)$. The final state is always S_0.

Therefore, the probabilities of a node's mobility phases are obtained as

$$P_s = \frac{\tau_s}{\tau_s + \tau_m} \quad P_m = \frac{\tau_m}{\tau_s + \tau_m} \tag{14}$$

from which we further have

$$P_{s,s} = P_s^2 \quad P_{s,m} = 2P_sP_m \quad P_{m,m} = P_m^2\cdot \tag{15}$$

2) Cluster Lifetime: 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 that are affiliated to the clusterhead. We assume that 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, and each time a clustermember leaves the cluster, the state transits one step to the left. At time 0, the transition starts at an initial state S_{init}, which denotes the number of clustermembers at the time that the cluster is constructed. The cluster lifetime is the transition time from state S_{init} to S_0.

On state S_0, the clusterhead will recluster to merge into another cluster; therefore, the clustermember joining rate is $\lambda_0 = 0$. The joining rate λ_j on the other states S_j is state dependent on a 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 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; therefore, 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

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

where $(N_m/E(T_m)) + (N_h/E(T_h))$ accounts for the network-wide arrival rate of nodes seeking a cluster to join, $(\pi r^2/l^2)$ 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 can join the clusterhead only when they have relative movement. A relative stationary node inside the clusterhead's neighborhood is likely to be an already-affiliated clustermember, and a relative stationary node out of the clusterhead's reach 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

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

The clustermember departure rate is also state dependent and is expressed as

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

The initial state S_{init} has (N_m/N_h) number of clustermembers on average.

The difficulty in finding $E(T_h)$ rises from the presence of an infinite number of possible paths from S_{init} to S_0. We solve this problem as follows. By 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 the time that a joining node arrives or an affiliated clustermember leaves. Both instances are Poisson processes, so the combination is still a Poisson process with parameter $\lambda_j + \mu_j$. Therefore, the mean sojourn time on S_j is $(1/(\lambda_j + \mu_j))$. After this sojourn time, a left transition occurs with probability $(\mu_j/(\lambda_j + \mu_j))$, or a right transition occurs with probability $(\lambda_j/(\lambda_j + \mu_j))$. At the boundary state S_n, only the left transition is possible, because there are no arrivals of joining nodes.

We reformat (19) into a matrix form, shown at the bottom of the next page, considering (16) and (18).

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

D. Intercluster Link Lifetime

Intercluster connectivity is determined by the availability of the links that connect two nodes from the two clusters, respectively. Such links include nodes between 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 intercluster *Logical Link* to be the set of all these

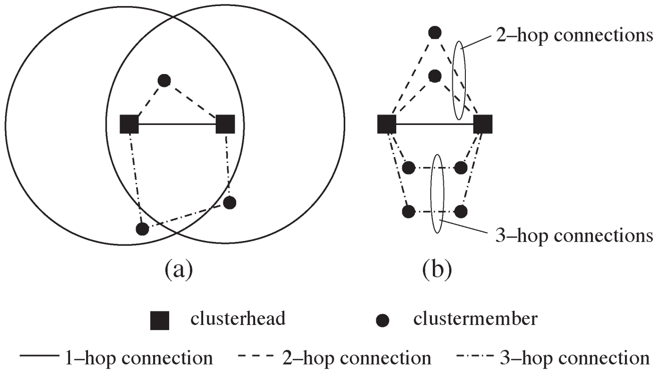


Fig. 6. Connection types and the logical link.

connections, as shown in Fig. 6(b). The logical link exists as long as *any* of the connection types exists. The *longest* intercluster 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 link to transmit packets. Therefore, if a clustermember can be used in both a two-hop connection and a three-hop connection, the two-hop connection will always be used. This observation allows us to assume that a clustermember appears either in a two-hop or a three-hop connection but not in both. The mean cluster membership time $E(T_m)$ that we have earlier derived 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 internode link lifetimes have identical and exponential distributions with mean $(1/\mu) = E(T_m)$.

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. Determining the new connection arrival process is unmanageably difficult; therefore, we approximate the logical link lifetime by the time that all the initial connections fail, 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, $\mathcal{F} = \{F_x^{(j)}\} (j = 1, 2, \ldots, \sum_{i=1}^{3} n_i)$ 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 jth 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 jth failure in the sequence $\mathcal{F}$, and $1_i(F_x^{(j)})$ denote the indicator functions such that

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

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 $\binom{n_1+n_2+n_3}{n_1}\binom{n_2+n_3}{n_2}\binom{n_3}{n_3} = ((n_1 + n_2 + n_3)!/n_1!n_2!n_3!)$. $P(\mathcal{F})$ and $T(\mathcal{F})$ are determined by

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

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

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

E. Path Lifetime

A path in a hierarchical network architecture consists of a sequence of intercluster links. Considering the clustering roles of two end nodes, there are also possible intracluster 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. One 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 intercluster and the intracluster links all have exponentially distributed lifetimes with mean $E(T_l)$ and $E(T_m)$, respectively, the *longest* path

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

$$\mathbf{T} = (\, t_1 \quad t_2 \quad t_3 \quad \cdots \quad t_{n-1} \quad t_n \,)^T$$

$$\mathbf{E} = (\, 1 \quad 1 \quad 1 \quad \cdots \quad 1 \quad 1 \,)^T$$

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

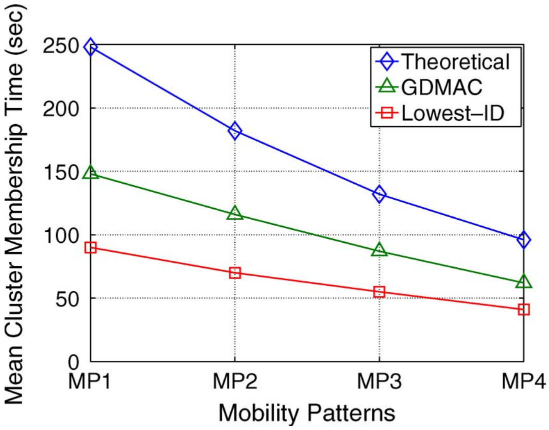


Fig. 7. Mean cluster membership time $E(T_m)$.

lifetime is then also exponentially distributed, with the mean being determined by

$$\begin{aligned} E(T_p) &= P_{h\text{-}h}E_{h\text{-}h}(T_p) + P_{h\text{-}m}E_{h\text{-}m}(T_p) \\ &\quad + P_{m\text{-}m}E_{m\text{-}m}(T_p) \\ P_{h\text{-}h} &= \left(\frac{N_h}{N}\right)^2 \qquad P_{h\text{-}m} = \frac{2N_hN_m}{N^2} \\ P_{m\text{-}m} &= \left(\frac{N_m}{N}\right)^2 \qquad E_{h\text{-}h}(T_p) = \frac{1}{c\mu_l} \\ E_{h\text{-}m}(T_p) &= \frac{1}{c\mu_l + \mu} \qquad E_{m\text{-}m}(T_p) = \frac{1}{c\mu_l + 2\mu} \end{aligned} \tag{25}$$

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

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 $N = 240$, $l = 2000$ m, and $r = 250$ m and specify four mobility patterns by tuning the node moving speed and pause time, as shown in Table I, to represent different mobility scenarios (e.g., low-speed pedestrians and high-speed vehicles). Aside from the theoretical analysis, we have implemented the lowest ID and GDMAC (with $K = 3$ and $H = 32$ as in [8]) clustering algorithms as a 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 lowest ID has a quite-short cluster membership time compared with the theoretical bound. GDMAC improves over lowest ID, but there is still a noticeable gap from the upper bound that was obtained from the analysis. One 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 recluster only after it has lost the contact to its clusterhead, and a clusterhead tries to recluster only after it has lost the contact to all of its clustermembers. The average values that were 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 $t_{\text{init}} = t_3$, because S_3 is the state closest to the mean cluster size $(184/56) = 3.28$. We see in Fig. 8 that the cluster lifetime of lowest ID is significantly lower than the analytical bound, and GDMAC has not yet reached this bound.

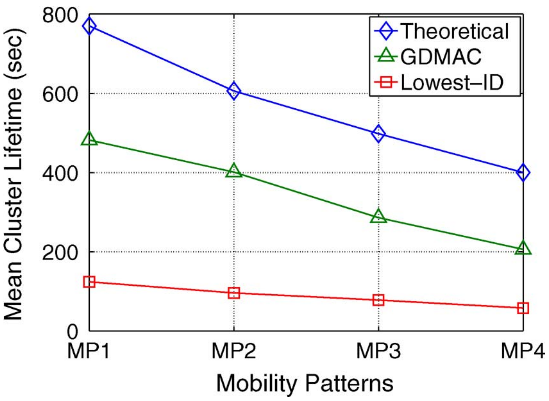


Fig. 8. Mean cluster lifetime $E(T_h)$.

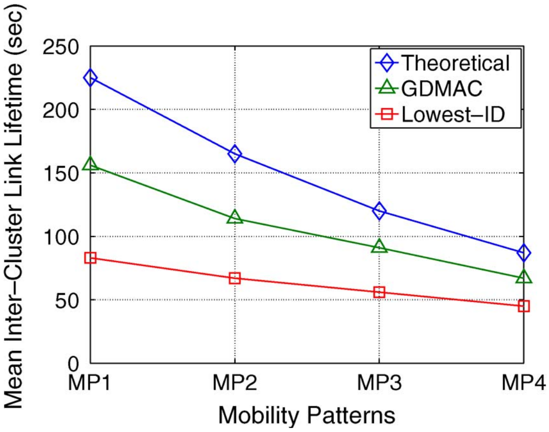


Fig. 9. Mean intercluster link lifetime $E(T_l)$.

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

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

The mean intercluster link lifetime is then determined based on (22). Fig. 9 plots the results from the analysis and

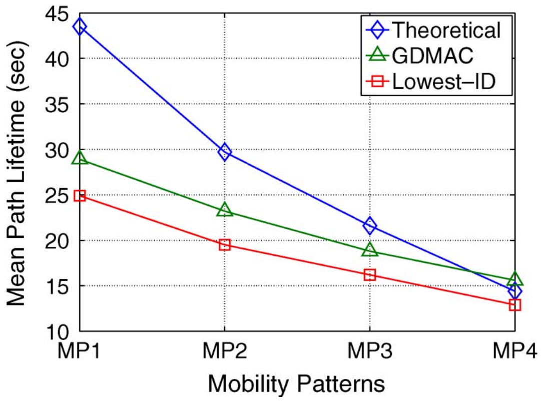


Fig. 10. Mean path lifetime $E(T_p)$.

simulations. Again, we observe the similar instability of lowest ID in the intercluster connectivity and the stability gap between the GDMAC and the analytical bound.

Finally, we choose ten random source–destination pairs and measure their average path length in the number of intercluster 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 ten node pairs are also used in the lowest ID and GDMAC simulations. The analytical mean path lifetime is determined by (25) and is plotted in Fig. 10, together with the simulation results. Fig. 10 shows that the path lifetime could be higher than those achieved by lowest ID and GDMAC. We notice that, in mobility pattern 4, the path lifetimes from the analysis, GDMAC, and lowest ID are very close, and GDMAC even exceeds the analytical result, because our analysis derives a lower estimate of the intercluster link lifetime, as mentioned in Section III-D, which leads us to underestimate the path lifetime. However, this observation, indeed, indicates that there is limited room for improving the path stability in a high-mobility environment.

IV. Mobility-and Energy-Aware Clustering Algorithm

The stability analysis reveals the longest lifetimes of the clusters, the intercluster links, and the paths that are allowed by the node mobility. It also shows that these longest lifetimes are achieved by extending a node's clustering status to its natural ending. Premature reclustering 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: 1) It preserves the existing clusters to the maximum extent, and 2) it also tries to maximize the cluster lifetime at cluster construction. The first feature 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 a long time. We interpret stability mainly in the mobility sense but also with energy considerations. As such, we name the new algorithm the mobility- and energy-aware clustering algorithm (MEACA).

The MEACA Algorithm

```
initialize $u.timer$
when $u.timer$ expires
  reschedule $u.timer$
  **if**    $u.status$ = HEAD and $u.member \neq$ NULL
       or $u.status$ = MEMBER and $u.head \neq$ NULL
     **then return**
  $C \leftarrow N(u) \cup \{u\}$
  $C \leftarrow \{v \,|\, v.status \neq \text{MEMBER}, v \in C\}$
  $A_m^* \leftarrow \alpha \cdot \max\{A_m(v) \,|\, v \in C\}$ $(0.9 < \alpha < 1)$
  $S \leftarrow \{v \,|\, A_m(v) \geq A_m^*, v \in C\}$
  $w \leftarrow \arg_v \max\{A_e(v) \,|\, v \in S\}$
  **if** $w = u$
     **then** $u.status \leftarrow$ HEAD
           $u.head \leftarrow u$
  **else if** $w.status$ = HEAD
     **then** $u.status \leftarrow$ MEMBER
           $u.head \leftarrow w$
  **return**
```

Fig. 11. MEACA.

A. MEACA

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, and τ_v is the neighboring time between a neighbor node v and the node u. The neighboring time is acquired using *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: a larger value means higher stability. The energy attribute A_e of u is defined to be the estimated lasting time of its 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 reclustering, it determines its clustering status by comparing its own attributes with its neighbors' attributes. Suppose 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 that will join the selected node. Fig. 11 presents the clusterhead selection algorithm in pseudocode. Node u periodically checks if it needs reclustering. If its status is a clusterhead, but there is no affiliated clustermember left (or its status is a clustermember, but it has lost contact with 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 stability 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 stability. 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 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. Because w may also be in the clustering process, there are three possibilities: 1) If w was determined to be a clusterhead, w accepts u's registration; 2) if w was determined to be a clustermember, w rejects u's registration, and u repeats the clustering algorithm (in the second round, w is excluded from the set C; therefore, u will end up with another selection of the clusterhead); 3) if w has not yet made its decision, u waits until w has made its decision.

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

B. MEACA Properties

Property 1: The Algorithm Converges: Algorithm convergence is the property that a node can determine its status in finite time. We prove this property in different cases.

Case 1) If u is determined to be a clusterhead, its status is finalized in one round of executing this algorithm.

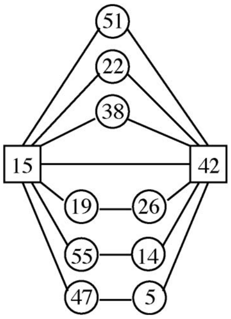
Case 2) If u selects $w (w \neq u)$, and w is a clusterhead, u's status as a clustermember is also finalized in one round.

Case 3) If w is a clustermember, u repeats the clustering process one more time, with w being 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 yet made its decision but is not waiting for another node to decide, then w will determine its status in a finite number of rounds according to our discussion on u. Then, u will also finalize in a finite number of rounds.

Case 5) If w is waiting for w', the chain of waiting nodes u, $w, w', \ldots$ must be finite, given the finite node population in the network, and they satisfy $A_m(u) < A_m(w) < A_m(w') < \cdots$ with high probability when α is large. The last node in this chain will determine its status first, and then, the rest of the nodes will successively determine their status in the reverse order of the chain, with each node having a finite number of clustering rounds.

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.



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. Example connection table.

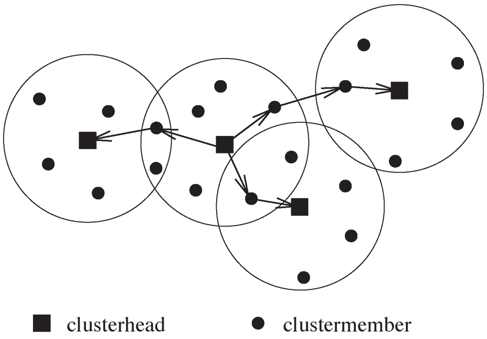


Fig. 13. Route request broadcasting example.

Property 2: The Algorithm Has No Ripple Effect: The ripple effect is the phenomenon that a reclustering node triggers a sequence of cluster changes in a wide area. In MEACA, a reclustering 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, reclustering is limited only to the participating nodes, and the rest of the network is not affected.

Property 3: The Algorithm Balances Node Energy: Based on 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 stability. Former clusterheads are expected to have lower energy than the former clustermembers and, therefore, will likely hand over the clusterhead role to other nodes at reclustering. The mean cluster lifetime is relatively short (in minutes) compared with the usual node energy lasting time (in hours); therefore, a clusterhead has many chances to shift its role before depleting its energy.

C. Comparison With Related Work

The main differences of MEACA from the existing work on clustering algorithms are given in two aspects. First, the existing algorithms [2], [7], [8] are *extending*, rather than *maximizing*, the cluster stability, whereas MEACA attempts to reach the maximum cluster stability under the constraint of node mobility. 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 premature reclustering. Second, when node mobility is considered, the existing schemes [1], [4], [5] measure

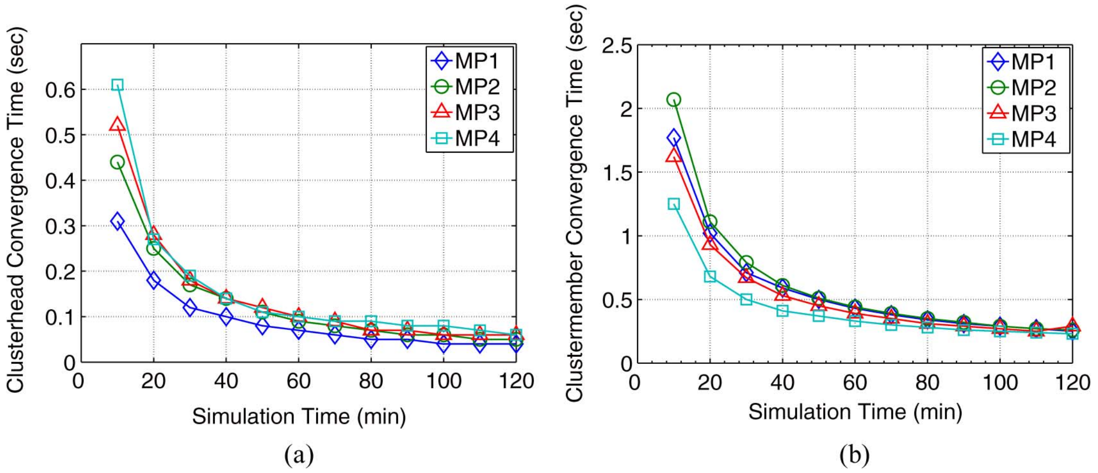


Fig. 14. Node status determination time (MEACA). (a) Clusterhead. (b) Clustermember.

the node speed and then infer the network topology stability, whereas MEACA directly measures the topology stability by defining the mobility attribute as the node neighboring times, which is simpler but more accurate.

V. Clustered Network Overlay Routing Protocol

Based on the analysis, we know that, to achieve the longest path lifetime, it is equally important to maximize the intercluster connectivity aside from stabilizing individual clusters. In this section, we propose the clustered network overlay routing protocol (CNORP), which aims to achieve two goals: 1) establishing the intercluster links and maintaining them to the maximum extent and 2) implementing the path discovery and packet forwarding mechanisms in the overlay network that is composed of 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 intercluster logical link. CNORP 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.

Clusterheads gather connection information through *hello* messages. These hello messages are different from those used in the clustering attribute exchanges, although they may be combined. Every node periodically broadcasts hello messages that 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 one- and two-hop connections to its neighbor clusterheads: Distance 0 in a received hello message indicates a one-hop connection to the broadcasting clusterhead, whereas distance 1 indicates a two-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 one- or two-hop connection to a neighbor clusterhead, then its clusterhead may utilize it to reach that neighbor clusterhead in a two- or three-hop connection via the reporting clustermember. A clusterhead organizes the connections in a tabular format, as shown in Fig. 12. The connection table is kept up to date, because any topology changes will be reflected in the periodical hello messages and clustermember reports. An intercluster logical link is broken 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.

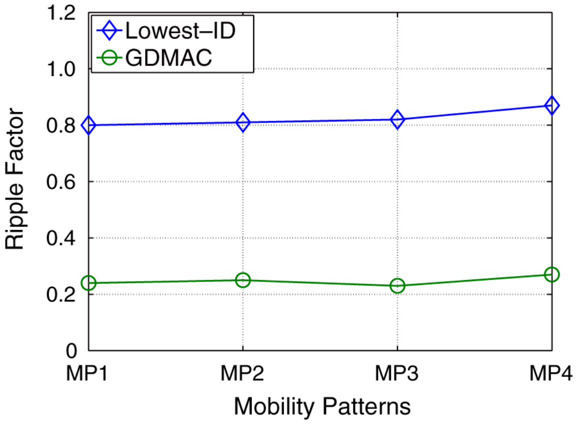
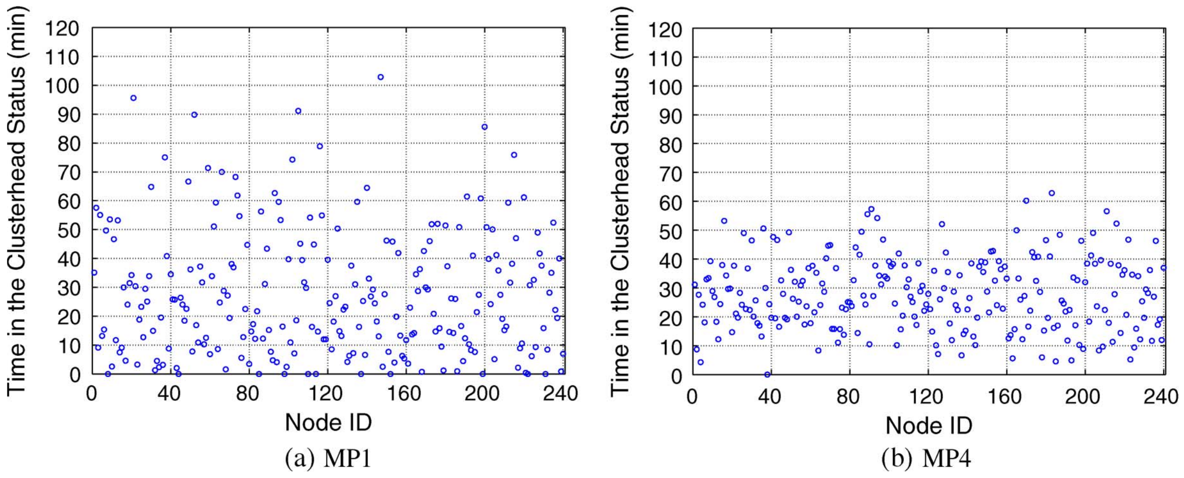


Fig. 15. Ripple effect of lowest ID and GDMAC.

A clustermember organizes its connections to the neighbor clusterheads in a similar table. A clustermember does not receive reports from other clustermembers; therefore, it does not have three-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

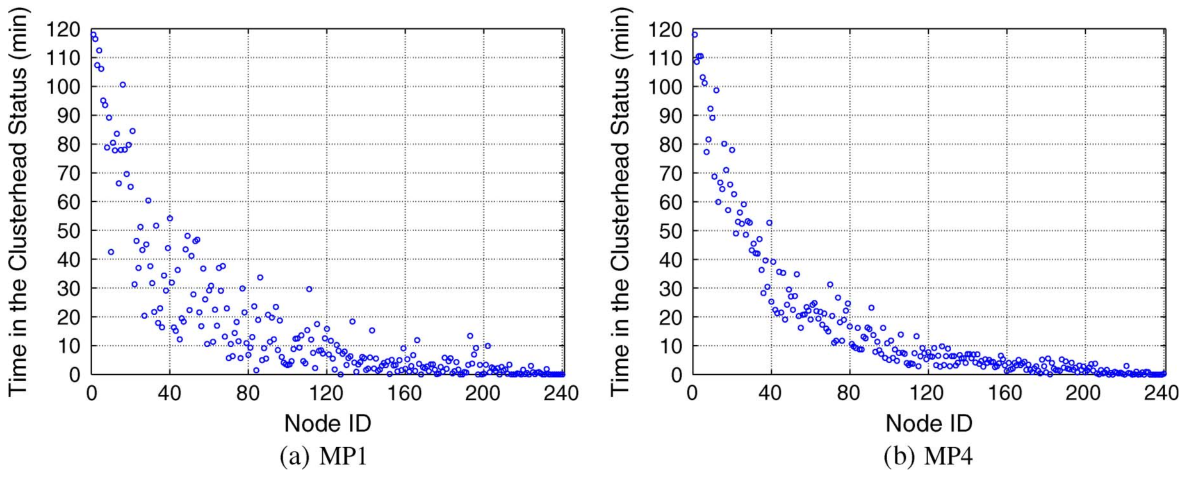
In the overlay network of the connected clusterheads, any reactive routing protocol (e.g., ad hoc on-demand distance vector and dynamic source routing) can be used for path discovery and maintenance. Proactive routing protocols are not suitable because of their excessive overhead when the clusters change.



(a) MP1

(b) MP4

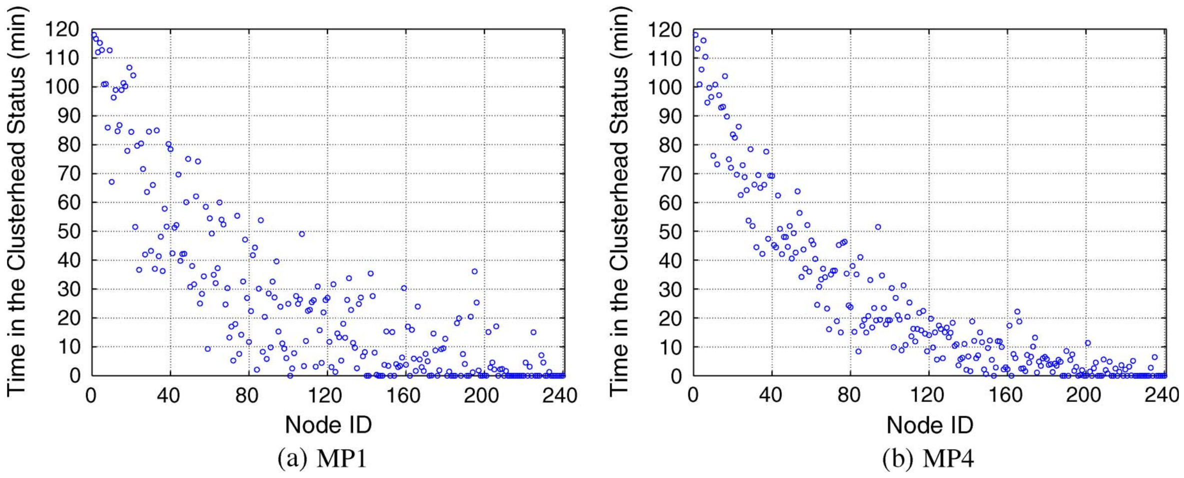
Fig. 16. Time distribution of the clusterhead status (MEACA). (a) MP1. (b) MP4.



(a) MP1

(b) MP4

Fig. 17. Time distribution of the clusterhead status (lowest ID). (a) MP1. (b) MP4.



(a) MP1

(b) MP4

Fig. 18. Time distribution of the clusterhead status (GDMAC). (a) MP1. (b) MP4.

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. One example of the route-request broadcasting is shown in Fig. 13. A path that was 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, routed to the destination node's clusterhead, and 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 when forwarding the packet, but using the shortest available connection is always enforced.

C. Comparison With Related Work

CNORP differs from the existing work, because it does not require a global cluster membership table compared with [7] and [21]. It also simplifies the path management and maximizes the path lifetime compared with [22] and [23] by utilizing the intercluster logical links. Compared with [24], a clustermember that uses CNORP acquires sufficient local routing information by itself such that it can independently forward packets to the neighbor clusterheads without obtaining the routing information from its clusterhead. In addition, using unicast to implement the route-request broadcasting suppresses the redundant broadcast messages in a simpler way.

VI. Performance Evaluation

We have implemented MEACA and CNORP 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 is set to be every 2 s.

Fig. 14(a) plots the mean time that a node spent in determining its status as a clusterhead using MEACA. Initially, it takes 0.3–0.6 s 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 less than 0.1 s. The similar trend is observed in the mean time to determine the clustermember role, which decreases from 1.3–2 s to 0.3 s, as shown in Fig. 14(b). These results demonstrate that MEACA converges.

We have proven that MEACA does not have the ripple effect. As a comparison, we plot the ripple effect of lowest ID and GDMAC in Fig. 15, where the *ripple factor* is defined as the ratio of the number of premature reclustering to the number of mature reclustering. Lowest ID has a ripple factor of 0.8, which indicates that each node status update causes, on the average, extra 0.8 update among its neighbors. GDMAC still has the ripple effect problem, although it reduces the ripple factor to 0.2.

Figs. 16–18 depict the cumulative time that each node stayed in the clusterhead status during a 2-h simulation. We observe the obvious dependence of this cumulative time on node ID numbers in the lowest ID and GDMAC algorithms, where the nodes with low IDs have the clusterhead status significantly longer than the others. This phenomenon poses big fairness and energy-balancing problems. MEACA, on the contrary, does not have the ID dependence problem. We see in Fig. 16 that both the low- and high-ID nodes have equal chances to become the clusterheads.

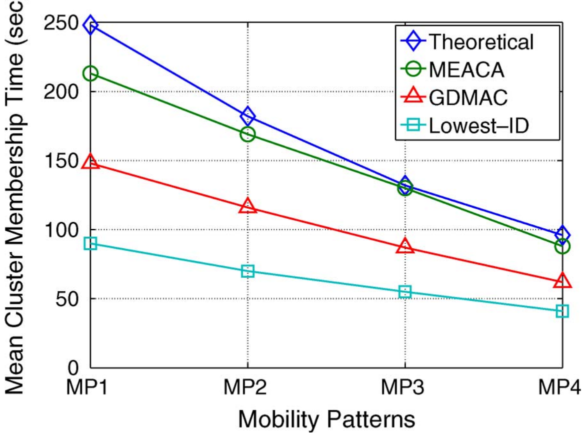


Fig. 19. Mean cluster membership time $E(T_m)$.

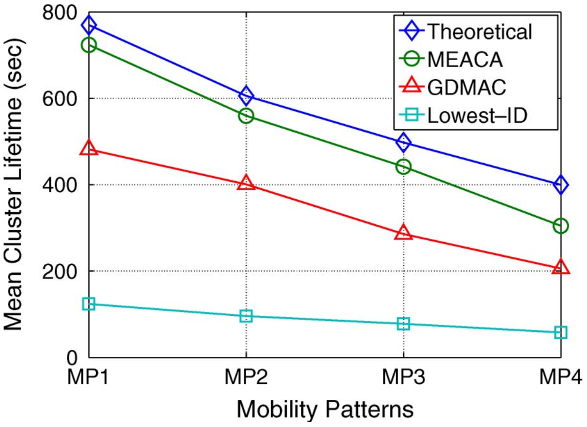


Fig. 20. Mean cluster lifetime $E(T_h)$.

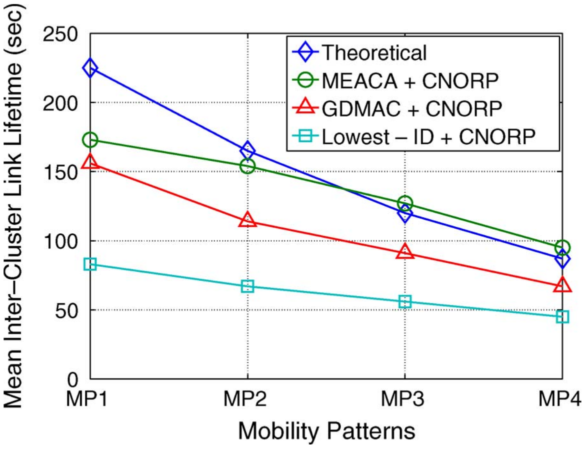


Fig. 21. Mean intercluster link lifetime $E(T_l)$.

Figs. 19 and 20 compare the cluster membership time and the cluster lifetime achieved by MEACA to the previous results obtained from the analysis and the lowest ID and GDMAC simulations. We observe that MEACA is an improvement over lowest ID and GDMAC, and its performance approaches the theoretical bounds. Similar improvements are observed in Figs. 21 and 22, which compare the intercluster link lifetime and the path lifetime. We see that the performance of MEACA

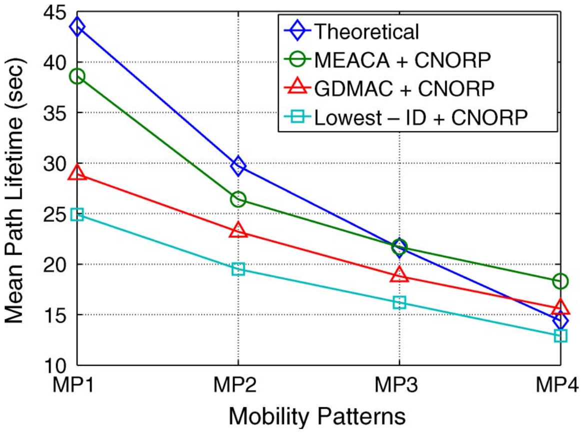


Fig. 22. Mean path lifetime $E(T_p)$.

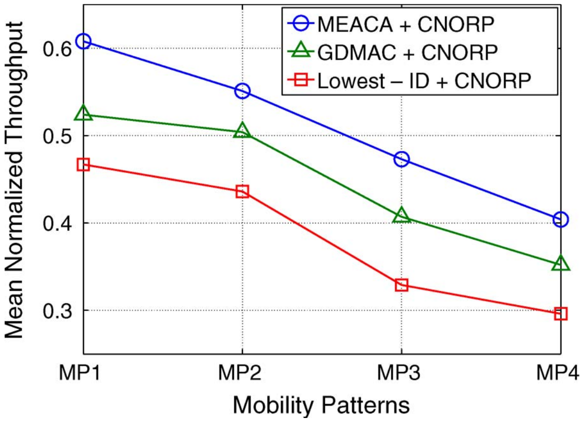


Fig. 23. Throughput comparison.

and CNORP exceeds the theoretical results in the high-mobility patterns (i.e., MP3 and MP4), because we have underestimated the intercluster link lifetime and the path lifetime in the analysis. Fig. 23 shows the normalized throughput of the ten random connections using different clustering algorithms. Due to a longer path lifetime, MEACA achieves higher end-to-end throughput than the lowest ID and GDMAC algorithms.

VII. Conclusion

In hierarchical ad hoc networks that consist of heterogeneous nodes (e.g., stationary devices, pedestrians, and vehicles), random node mobility significantly impacts the stability of network architecture. The existing efforts to stabilize the hierarchical architecture do not address the maximum stability problem, which is the foundation to understand the optimal performance of mobile ad hoc networks. In this paper, we have presented the mathematical modeling and analysis of the maximum lifetimes of the clusters, the intercluster links, and the end-to-end paths in a mobile environment, as described by a random-walk-like mobility model. We have found that the maximum stability of the hierarchical architecture is achieved when premature reclustering is avoided, and intercluster connectivity is maximized. Then, we have designed MEACA and CNORP, which work together to maximize the stability of hierarchical networks. Simulation has shown that their performance approaches the theoretical bounds that were 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," *Int. J. Netw. Manag.*, vol. 11, no. 6, pp. 387–395, Nov./Dec. 2001.

[2] P. Basu, N. Khan, and T. D. C. Little, "A mobility based metric for clustering in mobile ad hoc networks," in *Proc. IEEE ICDCSW*, Apr. 2001, pp. 413–418.

[3] A. B. McDonald and T. F. Znati, "A mobility-based framework for adaptive clustering in wireless ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1466–1487, Aug. 1999.

[4] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: A weighted clustering algorithm for mobile ad hoc networks," *Cluster Comput.*, vol. 5, no. 2, pp. 193–204, Apr. 2002.

[5] L. Bao and J. J. Garcia-Luna-Aceves, "Topology management in ad hoc networks," in *Proc. ACM MobiHoc*, Jun. 2003, pp. 129–140.

[6] S. Sivavakeesar and G. Pavlou, "Scalable location services for hierarchically organized mobile ad hoc networks," in *Proc. ACM MobiHoc*, 2005, pp. 217–228.

[7] C.-C. Chiang, H.-K. Wu, W. Liu, and M. Gerla, "Routing in clustered multihop, mobile wireless networks with fading channel," in *Proc. IEEE SICON*, Apr. 1997, pp. 197–211.

[8] R. Ghosh and S. Basagni, "Limiting the impact of mobility on ad hoc clustering," in *Proc. ACM PE-WASUN*, Oct. 2005, pp. 197–204.

[9] M. Gerharz, C. de Waal, M. Frank, and P. Martini, "Link stability in mobile wireless ad hoc networks," in *Proc. IEEE LCN*, 2002, pp. 30–39.

[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. ACM MobiHoc*, Jun. 2003, pp. 245–256.

[11] P. Samar and S. B. Wicker, "On the behavior of communication links of a node in a multi-hop mobile environment," in *Proc. ACM MobiHoc*, May 2004, pp. 145–156.

[12] V. Lenders, J. Wagner, and M. May, "Analyzing the impact of mobility in ad hoc networks," in *Proc. ACM REALMAN*, May 2006, pp. 39–46.

[13] E. Royer, P. M. Melliar-Smith, and L. Moser, "An analysis of the optimum node density for ad hoc mobile networks," in *Proc. IEEE ICC*, 2001, pp. 857–861.

[14] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proc. IEEE*, vol. 75, no. 1, pp. 56–73, Jan. 1987.

[15] M. Gerla and J. T.-C. Tsai, "Multicluster, mobile, multimedia radio network," *Wirel. Netw.*, vol. 1, no. 3, pp. 255–265, Oct. 1995.

[16] S. Basagni, "Distributed clustering for ad hoc networks," in *Proc. ISPAN*, Jun. 1999, pp. 310–315.

[17] V. Kawadia and P. R. Kumar, "Power control and clustering in ad hoc networks," in *Proc. IEEE INFOCOM*, 2003, pp. 459–469.

[18] R. Ramanathan and M. Steenstrup, "Hierarchically organized, multihop mobile wireless networks for quality-of-service support," *Mobile Netw. Appl.*, vol. 3, no. 1, pp. 101–119, Jun. 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," in *Proc. IEEE ICC*, 2002, pp. 3179–3185.

[20] G. Pei, M. Gerla, and X. Hong, "LANMAR: Landmark routing for large scale wireless ad hoc networks with group mobility," in *Proc. ACM MobiHoc*, 2000, pp. 11–18.

[21] N. Zhou and A. A. Abouzeid, "Routing in ad hoc networks: A theoretical framework with practical implications," in *Proc. IEEE INFOCOM*, 2005, pp. 1240–1251.

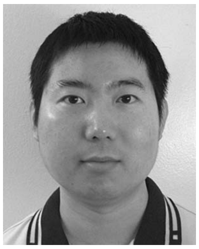
[22] M. Jiang, J. Li, and Y. C. Tay, "Cluster-Based Routing Protocol (CBRP)," *IETF MANET Working Group, Internet Draft*, Jul. 1999.

[23] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: A core-extraction distributed ad hoc routing algorithm," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1454–1465, Aug. 1999.

[24] E. Belding-Royer, "Hierarchical routing in ad hoc mobile networks," *Wirel. Commun. Mob. Comput.*, vol. 2, no. 5, pp. 515–532, Aug. 2002.

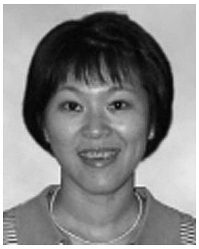
[25] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wirel. Commun. Mob. Comput.*, vol. 2, no. 5, pp. 483–502, Aug. 2002.

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



Yi Xu (S'00–A'02–M'04–S'06) received the B.E. degree in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1998 and the M.E. degree in electrical engineering from the National University of Singapore, Singapore, in 2002. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh.

His research interests include wireless networks, in particular fundamental understanding, performance analysis, protocol design, and security issues.



Wenye Wang (M'98) received the B.S. and M.S. degrees from the Beijing University of Posts and Telecommunications, Beijing, China, in 1986 and 1991, respectively, and the M.S.E.E. and Ph.D. degrees from the Georgia Institute of Technology, Atlanta, in 1999 and 2002, respectively.

She is currently an Associate Professor with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh. Her research interests include mobile and security computing and quality-of-service (QoS) sensitive networking protocols in single-hop and multihop networks.

Dr. Wang is a member of the Association for Computing Machinery. Since 2004, she has been serving on the Program Committees for the IEEE INFOCOM, the IEEE International Conference on Communications, and the IEEE International Conference on Computer Communications and Networks. She received the National Science Foundation Faculty Early Career Development Award in 2006.