

Research Article

Reliable Node Clustering for Mobile Ad Hoc Networks

Tao Wang¹ and William N. N. Hung²

¹ Department of Mechanical Engineering, Institute of Manufacturing Engineering, Tsinghua University, Beijing 100084, China

² Synopsys Inc., 700 East Middlefield Road, Mountain View, CA 94043, USA

Correspondence should be addressed to William N. N. Hung; william_hung@alumni.utexas.net

Received 8 February 2013; Accepted 18 March 2013

Academic Editor: Xiaoyu Song

Copyright © 2013 T. Wang and W. N. N. Hung. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile ad hoc networks (MANETs) are gaining popularity in recent years due to their flexibility, the proliferation of smart computing devices, and developments in wireless communications. Due to the mobile nature of the network, creating reliable clusters that can sustain for long period of time is a challenging problem. In this paper, we use probabilistic analysis to guide our clustering algorithm towards more reliable clusters. We also use scatter search to perform clustering while considering various performance metrics. Experiment results show that our clustering approach produces more reliable clusters than prior approaches.

1. Introduction

In recent years, wireless technologies have gained a lot of popularity. The worldwide proliferation of wireless devices, such as laptop computers and smart phones, has underscored the importance of mobile computing in our daily lives. Some applications of mobile computing do not depend on a preexisting infrastructure, such as routers in wired networks or access points in wireless networks; their communication can utilize a mobile ad hoc network (MANET). A MANET is a self-configuring, infrastructureless, wireless network of mobile devices. Many standardized technologies support MANET, such as Bluetooth [1], IEEE 802.11 (WiFi) [2], IEEE 802.15.3 (Wireless PAN) [3], and Ultra-Wideband (UWB). A mobile ad hoc network enables us to setup a temporary mobile network for instant communication without any fixed infrastructure. It has great application potential in a variety of areas such as emergency rescue, disaster relief, mobile conferencing, law enforcement, battle field communications. It has been shown that a hierarchical network architecture will outperform a flat structure for large MANET regardless of routing schemes [4–6]. A typical implementation of a hierarchical architecture is through a clustered structure. Choosing clusterheads optimally is an NP-hard problem [7].

1.1. Reliability Characterization. Reliability for mobile ad hoc networks can be divided into three categories: mobile

coverage reliability, clusterhead reliability, and communication reliability.

(i) *Mobile Coverage Reliability.* Mobile nodes in the cluster can move out of the transmission range of the clusterhead. If a mobile node cannot find any other clusterhead within its transmission range, it will trigger a reclustering of the entire mobile ad hoc network. Reclustering is costly because it completely destroys the previous cluster architecture and triggers a lot of message exchanges in order to build new clusters. Frequent reclustering can severely affect the network performance, bandwidth allocation, and scheduling protocols. Hence, there is a practical need to minimize the number of reclustering for MANETs.

(ii) *Clusterhead Reliability.* Mobile nodes are powered by batteries. If they run out of energy, they will not be able to participate in the mobile ad hoc network. In particular, clusterheads tend to consume a lot more energy than clustermembers. If a clusterhead runs out of battery, it will also trigger a reclustering of the entire mobile ad hoc network. Hence, it is important to reduce the power consumption of clusterheads and keep their energy high enough to last for longer time.

(iii) *Communication Reliability.* Communication links between some network nodes can also be unreliable. During network communication, packets/messages can get lost

when routing through these unreliable links. This problem also happens in classic communication networks, and there is a lot of related research [8].

In this paper, we focus on the first two categories (mobile coverage reliability and clusterhead reliability) because they pertain specifically to the mobile ad hoc nature of the network.

1.2. Reliability-Based Clustering. Prudent clustering for mobile ad hoc networks can address the concern for mobile coverage reliability and clusterhead reliability.

- (i) If we can choose the clusterheads such that every mobile node can always find a clusterhead within its transmission range along its mobile path, there will be no reclustering due to mobile coverage.
- (ii) The power used to support a wireless link is essentially proportional to the distance between mobile nodes [9]. Since clusterheads are responsible for a lot of the network communication infrastructure, the power usage of clusterheads is proportional to the distance between clusterheads and their clustermembers. If we can form clusters where clusterheads and clustermembers have close geographical proximity, we can reduce the power consumption of clusterheads.
- (iii) When electing mobile nodes for clusterheads, we can examine the energy (or battery storage) level for every node and elect nodes with more energy level. This will improve clusterhead reliability.

1.3. Prior Work. In [10, 11], the authors present a weighted clustering heuristic that combines various metrics for clustering, such as the number of nodes connected to a clusterhead, transmission power, mobility, and battery power of the nodes. This approach has been improved with genetic algorithm [12] and simulated annealing [13]. The weighted clustering algorithm was relatively simplistic and may not be able to capture complex scenarios such as group mobility [14]. Hence, a modified approach considering stability with multiobjective evolutionary algorithm was presented in [15]. Another stability-based clustering approach was presented in [16] but it did not consider other factors such as transmission power and battery life. Further investigation on battery power and clustering can be found in [17]. Reliability aware routing was investigated in [18]. Surveys of various clustering schemes can be found in [19–21].

Prior works on stable clustering focus on maintaining a stable link between the clusterhead and its members. They measure the velocity of the clusterhead [10, 11], and also the relative distance between the clusterhead and its members [15], forming a stable neighborhood in various ways [15, 16] such that the nodes being picked as clusterheads will have a stable set of clustermembers.

1.4. Our Contribution. There is a better way to improve the mobile coverage reliability of clusters. When a mobile node moves out of the transmission range of its original clusterhead, it will look for other clusterheads nearby within

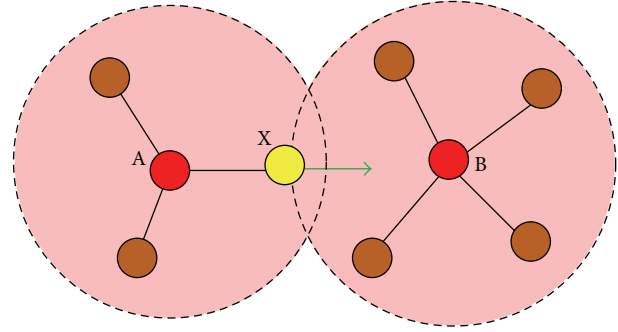


FIGURE 1: Mobile node X moving from cluster A to cluster B.

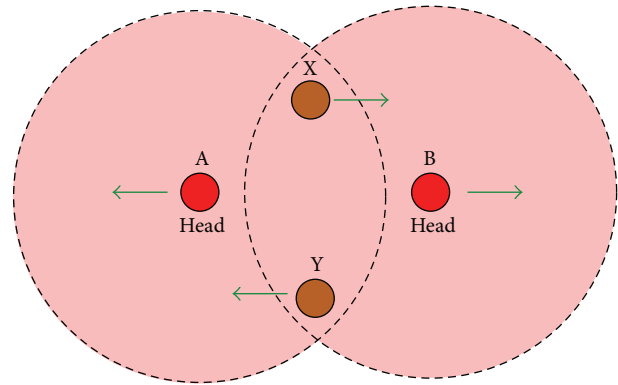


FIGURE 2: Nodes and clusterheads moving in different directions.

the transmission range and with sufficient capacity to support additional clustermembers. If found, the node will join that cluster, thus avoiding reelection of the clusterheads. Reclustering is only needed if current set of clusterheads cannot cover all mobile nodes [10, 11]. An example is shown in Figure 1, when node X moves out of the range of cluster A and joins cluster B, there is no reclustering. In this paper, we exploit this opportunity to improve the chances of mobile nodes finding alternative clusters when they moves out of the range of their original clusterhead. Experiment results show that our method reduces the number of reclusterings.

Prior clustering algorithms only pick certain nodes as clusterheads, while the rest of the nodes are left to freely join any (neighboring) clusterheads as clustermembers by themselves. However, consider the scenario in Figure 2. Nodes A and B are clusterheads. Nodes X and Y can choose to join either cluster A or cluster B. If node X joins cluster A, they will end up moving in opposite directions and node X will need to choose a new clusterhead. Even if there is another clusterhead (such as B) nearby, it may not have the extra capacity to handle node X. However, if node X joins cluster B in the original clustering, they will both move in the same direction and node X can stay with the same cluster longer. Thus the MANET clustering would be more stable. Similarly, node Y should better join cluster A instead of cluster B because they will be more stable due to their movements.

In this paper, we propose a new clustering scheme. While deciding the clusterheads, we also decide the clustermembers

of each cluster. The cluster membership will be broadcasted/communicated as an advice to each node. We use scatter search to optimize the clustering solution. Experiment results show that our clustering approach produces more reliable clusters over a diverse set of random scenarios.

2. Preliminaries

In this section, we introduce basic concepts and terminologies. Given a mobile ad hoc network, we use M to denote the set of mobile nodes in the network. Given a node $v \in M$, we use $N(v)$ to represent the set of nodes in the neighborhood of node v :

$$N(v) = \{v' \mid v' \in M \wedge \text{dist}(v, v') < r\}, \quad (1)$$

where r is the transmission range of node v and $\text{dist}(v, v')$ is the distance between nodes v and v' . The mobile ad hoc network is organized by clusters. Each cluster has a clusterhead, and the rest of its nodes are clustermembers. We use H to denote the set of clusterheads in the mobile ad hoc network. The degree of a clusterhead v , d_v , is the number of members in its cluster. The capacity (threshold) δ_v represents the number of neighbors that node v can optimally handle as a clusterhead. The degree difference $\Delta_v = |d_v - \delta_v|$ is the absolute difference between the degree and the capacity of node v .

According to [9], the power used to support a wireless link is essentially proportional to the distance in MANETs. Hence, the power consumption of a clusterhead node v is proportional to the sum of the distances between this node v and its clustermember nodes. Hence we have

$$\text{dist}_v = \sum_{v' \in C(v)} \text{dist}(v, v'). \quad (2)$$

Suppose each node v can calculate its residual battery energy E_v , then the remaining battery lifetime life_v is

$$\text{life}_v = \frac{E_v}{\text{Dist}_v}. \quad (3)$$

Given a minimization problem with m objectives, we use $x < y$ to denote that solution x dominates solution y :

$$x < y = f_i(x) \leq f_i(y) \wedge \exists j : f_j(x) < f_j(y), \quad (4)$$

where f_i is the objective function of i th objective.

The clustering algorithm is invoked at the beginning to determine clusterheads and their members. This clustering organization will sustain as long as the clusterheads and their member are within each other's transmission range. If a clustermember moves out of the transmission range from its original clusterhead, it will find another clusterhead nearby within the transmission range and with sufficient capacity to handle this node. If found, this node will join the nearby clusterhead. Otherwise, the reclustering will be needed.

3. Our Clustering Approach

The MANET clustering problem is to partition M into a set of clusters, $C = \{c_1, c_2, \dots, c_N\}$, such that

$$c_1 \cup c_2 \cup \dots \cup c_N = M. \quad (5)$$

For each cluster $c \in C$, the head node (clusterhead) is denoted by $h(c)$. The relative distance between each node and its clusterhead must lie within a transmission range r , that is,

$$\forall c \in C, \forall n \in c, \quad \text{dist}(n, h(c)) < r. \quad (6)$$

Prior clustering algorithms only focus on deciding clusterheads, while the rest of the nodes are left to freely join any (neighboring) clusterheads as clustermembers by themselves. We propose a slight modification to the clustering scheme. While deciding the clusterheads, we also decide the clustermembers of each cluster. The cluster membership will be broadcasted/communicated as an advice to every node.

One of the reliability objectives is to avoid reclustering. There are two factors that prevent the clustering algorithm from being reinvoked:

- (1) creating stable clusters in which clusterheads and their clustermembers are always within transmission range.
- (2) strategically pick clusterheads so that even if a clustermember moves out of the transmission range from its original clusterhead, it can become the member of another clusterhead nearby.

Prior clustering schemes focus only on the first idea. In this paper, we take advantage of both ideas.

To create stable clusters, we consider the average relative distance as well as the current relative distance between two nodes when deciding whether they should be put into the same cluster.

We define the *average neighborhood* $\bar{N}(v)$ of a node v to be the set of nodes with average relative distance within the transmission range r :

$$\bar{N}(v) = \{v' \mid v' \in M \wedge \overline{\text{dist}}(v, v') < r\}, \quad (7)$$

where $\overline{\text{dist}}(v, v')$ is the average relative distance between nodes v and v' . If any node v becomes a clusterhead, we only pick its clustermembers from the intersection of its neighborhood and its average neighborhood $R(v) = N(v) \cap \bar{N}(v)$.

To improve the chances of a node finding another cluster when moving out of range from its original clusterhead, we perform a probabilistic analysis for the clustering solutions.

Given a node v which is a clustermember, we can find all the clusterheads $\bar{h}(v)$ within its average neighborhood:

$$\bar{h}(v) = \{v' \mid v' \in H \wedge v' \in \bar{N}(v)\}. \quad (8)$$

The probability of node v finding another clusterhead v' is dominated by the values of r , $\overline{\text{dist}}(v, v')$, and $\sigma_{v,v'}$, where $\sigma_{v,v'}$ is the standard deviation of the relative distance between

v and v' . The probability is higher if the difference (slack) between the transmission range and their relative distance, $r - \overline{\text{dist}}(v, v')$, is larger. The probability is smaller if the standard deviation of the relative distance is larger. Hence, we can model the chances of node v finding another clusterhead with the probability measure $P(v)$:

$$P(v) = \sum_{v' \in \tilde{h}(v)} K_v^{v'} \frac{r - \overline{\text{dist}}(v, v')}{\sigma_{v,v'}}, \quad (9)$$

where $K_v^{v'}$ is a constant. The mobile coverage reliability is determined by the *minimum probability measure* of all clustermembers in the network:

$$P_{\min} = \min \{P(v) \mid v \in M \setminus H\}. \quad (10)$$

Given a mobile ad hoc network with a set of mobile node M , we want to partition the nodes into clusters based on the following objectives:

- (1) to maximize the minimum probability measure \tilde{P}_{\min}
- (2) to minimize the total power consumption of all clusterheads, dist ,

$$\text{dist} = \sum_{c \in C} \text{Dist}_{h(c)}, \quad (11)$$

- (3) to maximize the minimum lifetime of all clusterheads, Life

$$\text{Life} = \text{Min} \{ \text{Life}_{h(c)} \mid c \in C \}, \quad (12)$$

- (4) to minimize the total degree-difference of all clusterheads, Δ

$$\Delta = \sum_{c \in C} \Delta_{h(c)}. \quad (13)$$

Objective (1) is related to mobile coverage reliability. Objectives (2) and (3) are related to clusterhead reliability.

4. Scatter Search

The main idea of scatter search is to operate on a set of solutions, called the reference set, which strives to maintain both diversity and optimality. Scatter search combine solutions from the reference set to create new ones. Unlike a “population” in genetic algorithms, the size of the reference set in scatter search tends to be small. In genetic algorithms, we can randomly choose two solutions and apply a “crossover” or other combination operator to generate one or more offspring. A typical population in a genetic algorithm has 100 solutions, which are randomly sampled to generate combinations. In contrast, scatter search chooses two or more solutions from the reference set in a systematic way to create new solutions. Hence, the size of the reference set is kept small to allow all possible combination pairs.

The original idea of scatter search was introduced by Glover for integer programming [22] in 1977. It was later combined with Tabu search [23]. Glover presented a generalized template for scatter search [24] in 1998. More details can be found in [25–27].

```

(1) for  $i = 0$  to  $MaxIter$  do
(2) pool of solutions  $P := \text{Diversification}()$ 
(3) for  $p \in P$  do
(4)  $\hat{p} := \{p\} \cup \text{Improvement}(p)$ 
(5)  $RefSet := \text{Reference Set Update}(\hat{p})$ 
(6) end for
(7) repeat
(8) pool of subsets  $S := \text{Subset Generation}(RefSet)$ 
(9) for  $(s_1, s_2) \in S$  do
(10)  $s' := \text{Solution Combination}(s_1, s_2)$ 
(11)  $\hat{s} := \{s'\} \cup \text{Improvement}(s')$ 
(12)  $RefSet := \text{Reference Set Update}(\hat{s})$ 
(13) end for
(14) until  $RefSet$  remains the same
(15) end for

```

ALGORITHM 1: Scatter search.

4.1. *Basic Framework.* Scatter search is built using the following basic operations.

- (i) **Diversification:** it generates a collection of diversified trial solutions.
- (ii) **Improvement:** it enhances a trial solution with potentially better quality or diversity.
- (iii) **Reference Set Update:** it maintains the reference set using the trial solution. The set membership is determined by solution quality or diversity.
- (iv) **Subset Generation:** it systematically selects subsets from the reference set to combine them into new solutions.
- (v) **Solution Combination:** it transforms a given subset of solutions from Subset Generation into new solutions.

The high-level view of our scatter search implementation is shown in Algorithm 1.

4.2. *Diversification.* In diversification, we randomly generate a pool of trial solutions. To generate each solution, we start with a set of unassigned nodes $U = M$. We randomly pick a node $v \in U$, where the probability of picking each node v is biased by $|R(v)|$. This will ensure nodes with larger neighborhood and average neighborhood to be picked earlier. For each picked node v , we first assign it to be a new clusterhead, and then randomly shuffle its neighbor nodes in $R(v)$. For all the unassigned nodes in $R(v)$, we will assign them one by one to be the clustermember of v , until the cluster v reaches its capacity limit δ_v . We repeat this process to pick other nodes and other clusters, until all nodes have been assigned. This will ensure that all the clustermembers are within the transmission range of their clusterhead.

As we generate more and more trial solutions in the diversification pool, we adjust the bias of each node v with increasing offset to diminish the effect of $|R(v)|$, such that nodes with smaller neighborhoods will have a fairer chance of getting picked.

4.3. Improvement. To improve the quality or diversity of the trial solution, we introduce three tactics: (1) randomly reelect the head of each cluster, (2) heuristically reelect the head of each cluster, and (3) perturb the clustermembers of the solution. Each of these tactics will produce a new solution, and we pass all of them together with the original trial solution to update the reference set.

4.3.1. Random Clusterhead Reelection. For each cluster, we randomly pick a node. If the node can reach all clustermember nodes within the transmission range, it will become the head of its cluster.

4.3.2. Heuristic Clusterhead Reelection. Given a cluster of nodes, we first identify the potential candidates for clusterhead. These candidates must be able to reach all clustermember nodes within their transmission ranges. For each candidate, we compute the maximum of the average relative distance between the candidate and any clustermember node. The candidate with the lowest maximum will be picked as clusterhead.

4.3.3. Perturb Solution. To perturb a solution, we identify nodes with multiple clusterheads within their transmission ranges, and the average relative distances from these nodes to those multiple clusterheads are also within their transmission ranges. Hence, these nodes have alternative clusters to join. We randomly pick some of these nodes to move to their respective alternative clusters.

4.4. Reference Set Update. The purpose of reference set is to retain a few elite solutions with the best quality or diversity. Since we have multiple optimization objectives, we use nondominated sorting to pick solutions with the best quality. We split our reference set into two subsets, a high quality subset and a high diversity subset, which is similar to the ideas in [28–30] where the reference set is also split into two subsets.

To measure diversity, we introduce a concept of clustering difference. Suppose we are given two clustering solutions C^A and C^B ; we compute the difference score using the following scheme.

- (1) For every clusterhead in C^A , if the node is not a clusterhead in C^B , increment the score by the cardinality of the cluster in C^A .
- (2) For every clusterhead in C^B , if the node is not a clusterhead in C^A , increment the score by the cardinality of the cluster in C^B .
- (3) For every node that is a clusterhead in both C^A and C^B , increment the score by the absolute difference in the cardinality of the cluster in C^A and C^B .

4.5. Subset Generation and Solution Combination. In Subset Generation, we systematically generate all pairwise combinations from the reference set. For each pair of solutions, we combine them to create new solutions.

TABLE 1: Random experiments.

	Alg	Calls	Δ	Power	Lifetime
Scenario 1	WCA	120.0	35.0	1276.72	2628621.20
	SA	239.0	38.1	2252.21	924750.45
	GA	190.0	35.0	1276.09	3114806.84
	MOEA	25.0	35.2	1435.13	2104599.68
	Ours	22.2	30.1	1318.61	2783563.66
Scenario 2	WCA	970.0	25.0	3207.82	1018304.98
	SA	970.9	32.0	3409.93	730274.92
	GA	975.0	37.0	3596.76	612649.47
	MOEA	375.2	28.6	2776.77	1498541.71
	Ours	363.0	32.2	2673.35	1460839.50
Scenario 3	WCA	999.0	24.0	3565.14	896697.38
	SA	995.2	34.5	3783.96	594590.42
	GA	996.0	33.0	3562.35	751293.98
	MOEA	120.6	57.5	2658.63	1981888.65
	Ours	115.2	65.8	2503.66	2250301.86
Scenario 4	WCA	282.0	24.0	3546.54	770192.05
	SA	401.7	31.1	3641.09	658183.64
	GA	453.6	27.0	3877.86	599849.35
	MOEA	100.9	28.4	3279.22	1057580.01
	Ours	77.0	38.2	2855.70	1460495.10
Scenario 5	WCA	385.0	29.0	3635.96	780614.45
	SA	520.5	39.8	3684.19	528506.76
	GA	539.3	38.0	3842.48	382091.35
	MOEA	106.5	33.0	3095.48	934557.61
	Ours	64.7	43.8	2303.02	2024360.09
Scenario 6	WCA	244.0	29.0	1872.88	1760353.32
	SA	471.8	32.6	2366.21	1217731.34
	GA	477.0	32.0	2240.39	850329.92
	MOEA	57.2	26.5	2380.38	1252280.17
	Ours	23.6	33.7	1674.88	2850557.03
Scenario 7	WCA	565.0	30.0	2787.32	806865.69
	SA	731.7	31.4	3426.67	636318.24
	GA	770.0	36.0	3338.92	480890.51
	MOEA	99.8	29.6	2707.97	1183808.40
	Ours	64.6	33.0	2244.37	1808681.33
Scenario 8	WCA	955.0	14.0	3759.78	1175884.00
	SA	964.6	32.1	3960.76	554502.83
	GA	973.0	37.0	3929.95	616480.55
	MOEA	82.9	81.0	2286.75	2169811.24
	Ours	71.1	85.4	2197.82	2752915.84
Scenario 9	WCA	110.0	25.0	1276.72	2628623.69
	SA	228.2	28.5	1947.18	1376451.71
	GA	176.0	25.0	1276.09	3114827.01
	MOEA	13.7	28.2	1372.77	2711002.96
	Ours	14.4	33.1	1322.53	3127514.25
Scenario 10	WCA	974.0	30.0	4412.99	530579.23
	SA	985.4	38.0	4472.37	469315.73
	GA	988.0	30.0	4706.47	499732.87
	MOEA	283.3	30.6	4049.11	1225402.29
	Ours	243.4	41.8	3375.71	1639581.17

TABLE 1: Continued.

	Alg	Calls	Δ	Power	Lifetime
Scenario 11	WCA	511.0	30.0	3831.71	517272.03
	SA	663.9	38.3	4882.75	304942.92
	GA	699.0	37.0	4783.21	352880.24
	MOEA	141.9	27.2	3366.05	818898.17
	Ours	51.6	22.7	2457.29	1782671.16
Scenario 12	WCA	870.0	25.0	5400.82	610737.06
	SA	915.3	37.2	5193.26	303617.98
	GA	928.0	40.0	5048.88	358995.29
	MOEA	155.7	37.4	4085.91	1413935.07
	Ours	130.8	43.8	3743.62	1269461.80

Given a solution pair C^A and C^B , we use $|C^A|$ and $|C^B|$ to denote the number of clusters in each solution. We first randomly assign clusterheads from C^A and C^B to become clusterheads in the combined solution. For each clusterhead $c_a \in C^A$, its probability of being assigned as a clusterhead is

$$\frac{|c_a|}{|M|} + \frac{|C^B|}{|C^A| + |C^B|}. \quad (14)$$

Similarly, the probability for each clusterhead $c_b \in C^B$ is

$$\frac{|c_b|}{|M|} + \frac{|C^A|}{|C^A| + |C^B|}. \quad (15)$$

We then assign all nodes that are clusterheads in both C^A and C^B to become clusterheads in the combined solution. After picking all the above clusterheads, we try to assign the rest of nodes to be clustermembers, if a node x has a clusterhead in C^A that is also a clusterhead in the combined solution and its clusterhead in C^B is not assigned, we will assign x to the cluster headed by its original clusterhead in C^A . Similarly for clusterhead in C^B . If a node x has both its clusterheads in C^A and C^B assigned as clusterheads in the combined solution, we randomly assign x to one of these clusters in the combined solution, biased by the current and average relative distances between x and each clusterhead. Finally, for all remaining nodes that are unassigned (because they are out of range from the above clusterheads), we use the weighted random assignment scheme from the diversification method described in Section 4.2.

5. Experiments

We simulate a mobile ad hoc network with N nodes to evaluate the performance of various clustering algorithms. The simulation is carried out in a 500 km \times 500 km square-shaped region. We create 12 random scenarios shown in Table 1. In each scenario, the mobile nodes randomly form several groups. Some of the groups may have many nodes, and some groups may have only 1 or 2 nodes. All nodes are moving randomly throughout every scenario. Nodes within each group have their average velocity following the Reference Velocity Group Mobility Model [14, 31], whereas

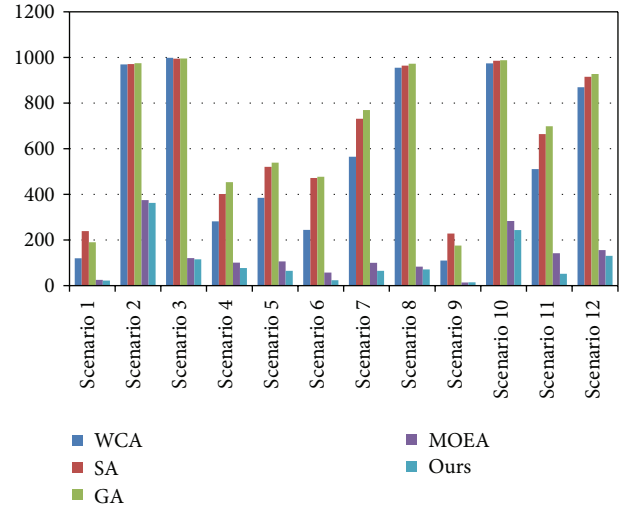


FIGURE 3: Number of reclusterings.

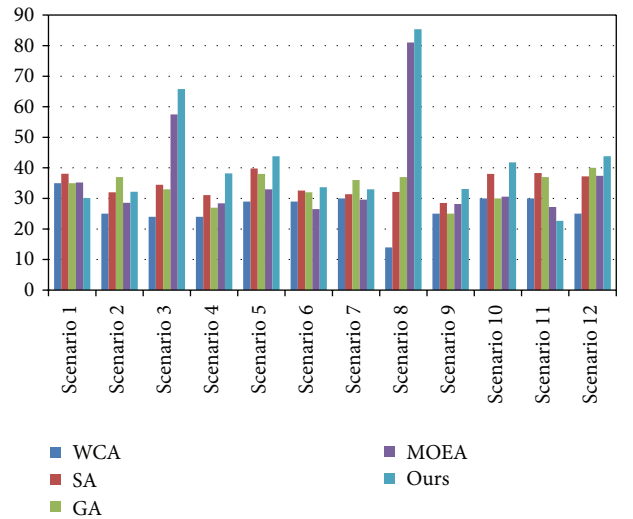


FIGURE 4: Degree difference.

the velocity of each node can still randomly deviate from the average velocity of the group. For each randomly generated scenario, we cluster the MANET using WCA [10, 11], SA [13], GA [12], MOEA [15], and our proposed algorithm, respectively. Each clustering algorithm is applied 10 times using different random seeds, which explore various random behavior of these algorithms such that experiment results can be easily reproducible. We take the average numeric result from each algorithm (over 10 different random seeds) and record them for each of the 12 random scenarios. The “calls” column shows the number of reclusterings. The Δ , “power,” and “lifetime” show the degree difference, power usage, and minimum (remaining) lifetime of the MANET after simulating the scenario for 1000 time units.

To better analyze the experiment results, we have plotted the number of reclusterings in Figure 3, the degree difference in Figure 4, the power usage in Figure 5, and the remaining lifetime in Figure 6.

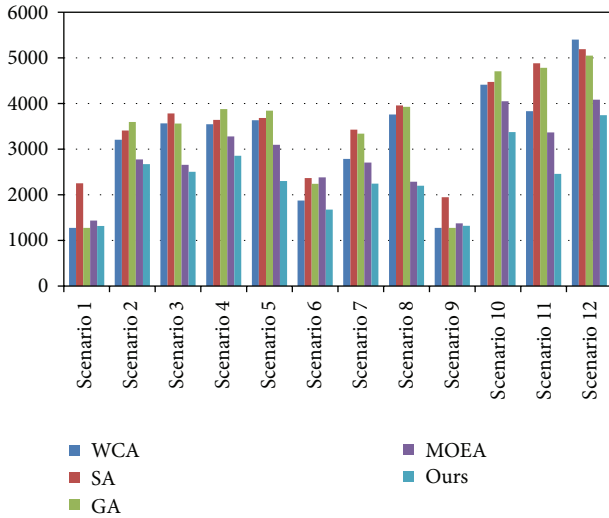


FIGURE 5: Power usage.

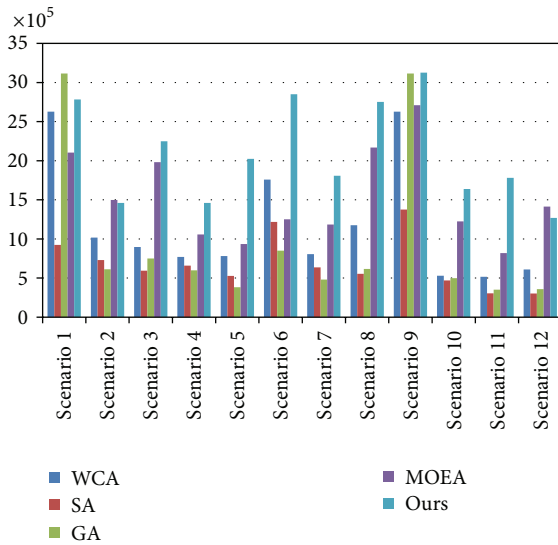


FIGURE 6: Remaining lifetime.

From experiment results, we can see that our clustering approach has the smallest number of reclusters (mobile coverage reliability) except scenario 9. The degree difference is higher using our clustering approach in some scenarios, which is a tradeoff for reliable clusters. However, notice that our degree difference is the best among all approaches for scenarios 1 and 11. We have the smallest power usage in 10 out of 12 random scenarios. We also have the longest remaining lifetime in 9 out of 12 random scenarios. Both power usage and lifetime are related to clusterhead reliability.

6. Conclusion

We have explored reliability metrics in node clustering for mobile ad hoc networks. We harnessed probabilistic analysis to control our clustering process to achieve more reliable clusters. We used scatter search to perform clustering while

considering robustness and performance metrics. Experiment results showed that our clustering results produce more reliable clusters than prior approaches.

References

- [1] "Bluetooth official website," 2011, <http://www.bluetooth.com>.
- [2] IEEE 802. 11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Computer Society, June 2007.
- [3] "IEEE 802. 15 Working Group for WPAN," 2011, <http://www.ieee802.org/15/>.
- [4] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.
- [5] X. Hong, K. Xu, and M. Gerla, "Scalable routing protocols for mobile ad hoc networks," *IEEE Network*, vol. 16, no. 4, pp. 11–21, 2002.
- [6] K. Xu, X. Hong, and M. Gerla, "An ad hoc network with mobile backbones," in *Proceedings of the IEEE International Conference on Communications (ICC)*, vol. 5, pp. 3138–3143, 2002.
- [7] S. Basagni, I. Chlamtac, and A. Farago, "A generalized clustering algorithm for peer-to-peer networks," in *Proceedings of the Workshop on Algorithmic Aspects of Communication*, July 1997.
- [8] 8th International Workshop on the Design of Reliable Communication Networks, October 2011.
- [9] M. Gerla and J. T. C. Tsai, "Multicluster, mobile, multimedia radio network," *Wireless Networks*, vol. 1, no. 3, pp. 255–265, 1995.
- [10] M. Chatterjee, S. K. Das, and D. Turgut, "An on-demand weighted clustering algorithm (WCA) for ad hoc networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '00)*, vol. 3, pp. 1697–1701, San Francisco, Calif, USA, 2000.
- [11] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: a weighted clustering algorithm for mobile ad hoc networks," *Cluster Computing*, vol. 5, no. 2, pp. 193–204, 2001.
- [12] D. Turgut, S. K. Das, R. Elmasri, and B. Turgut, "Optimizing clustering algorithm in mobile ad hoc networks using genetic algorithmic approach," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '02)*, vol. 1, pp. 62–66, November 2002.
- [13] D. Turgut, B. Turgut, R. Elmasri, and T. V. Le, "Optimizing clustering algorithm in mobile ad hoc networks using simulated annealing," in *Proceedings of the IEEE Wireless Communications and Networking (WCNC '03)*, vol. 3, pp. 1492–1497, New Orleans, La, USA, March 2003.
- [14] B. C. Li, "On increasing service accessibility and efficiency in wireless ad-hoc networks with group mobility," *Wireless Personal Communications*, vol. 21, no. 1, pp. 105–123, 2002.
- [15] H. Cheng, J. Cao, X. Wang, and S. K. Das, "Stability-based multi-objective clustering in mobile ad hoc networks," in *Proceedings of the 3rd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine '06)*, p. 27, 2006.
- [16] C. Konstantopoulos, D. Gavalas, and G. Pantziou, "Clustering in mobile ad hoc networks through neighborhood stability-based mobility prediction," *Computer Networks*, vol. 52, no. 9, pp. 1797–1824, 2008.
- [17] M. K. Watfa, O. Mirza, and J. Kawtharani, "BARC: a battery aware reliable clustering algorithm for sensor networks," *Journal*

- of Network and Computer Applications*, vol. 32, no. 6, pp. 1183–1193, 2009.
- [18] J. Wang, Y. Liu, and Y. Jiao, “Building a trusted route in a mobile ad hoc network considering communication reliability and path length,” *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1138–1149, 2011.
- [19] J. Y. Yu and P. H. J. Chong, “A survey of clustering schemes for mobile ad hoc networks,” *IEEE Communications Surveys and Tutorials*, vol. 7, no. 1, pp. 32–48, 2005.
- [20] S. Chinara and S. K. Rath, “A survey on one-hop clustering algorithms in mobile ad hoc networks,” *Journal of Network and Systems Management*, vol. 17, no. 1-2, pp. 183–207, 2009.
- [21] K. A. Darabkh, S. S. Ismail, M. Al-Shurman, I. F. Jafar, E. Alkhader, and M. F. Al-Mistarihi, “Performance evaluation of selective and adaptive heads clustering algorithms over wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 2068–2080, 2012.
- [22] F. Glover, “Heuristics for integer programming using surrogate constraints,” *Decision Sciences*, vol. 8, no. 1, pp. 156–166, 1977.
- [23] F. Glover, “Tabu search for nonlinear and parametric optimization (with links to genetic algorithms),” *Discrete Applied Mathematics*, vol. 49, no. 1–3, pp. 231–255, 1994.
- [24] F. Glover, “A template for scatter search and path relinking,” in *Artificial Evolution*, J. K. Hao, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, Eds., vol. 1363 of *Lecture Notes in Computer Science*, pp. 1–51, Springer, Berlin, Germany, 1998.
- [25] F. Glover, M. Laguna, and R. Marti, “Fundamentals of scatter search and path relinking,” *Control and Cybernetics*, vol. 29, no. 3, pp. 653–684, 2000.
- [26] F. Glover, M. Laguna, and R. Marti, “Scatter search and path relinking: advances and applications,” in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, Eds., vol. 57 of *International Series in Operations Research and Management Science*, pp. 1–35, Springer, New York, NY, USA, 2003.
- [27] R. Marti, M. Laguna, and F. Glover, “Principles of scatter search,” *European Journal of Operational Research*, vol. 169, no. 2, pp. 359–372, 2006.
- [28] W. N. N. Hung and X. Song, “BDD variable ordering by scatter search,” in *Proceedings of the IEEE International Conference on Computer Design (ICCD’ 01)*, pp. 368–373, Austin, Tex, USA, September 2001.
- [29] W. N. N. Hung, X. Song, E. M. Aboulhamid, and M. A. Driscoll, “BDD minimization by scatter search,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 8, pp. 974–979, 2002.
- [30] W. N. N. Hung and X. Song, “On optimal cell assignments in PCS networks,” in *Proceedings of the IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pp. 225–232, Phoenix, Ariz, USA, April 2002.
- [31] K. H. Wang and B. Li, “Group mobility and partition prediction in wireless ad-hoc networks,” in *Proceedings of the IEEE International Conference on Communications (ICC ’02)*, vol. 2, pp. 1017–1021, 2002.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

