A Multi-Head Clustering Algorithm in Vehicular Ad Hoc Networks

Shou-Chih Lo, Yi-Jen Lin, and Jhih-Siao Gao

Abstract—**Clustering is an important research topic in wireless networks, because cluster structures can facilitate resource reuse and increase system capacity. In this paper, we present a new clustering algorithm that considers both node position and node mobility in vehicular ad hoc environments. The proposed algorithm intends to create stable clusters by reducing reclustering overhead, prolonging cluster lifetime, and shortening the average distance between cluster heads and their cluster members. Most important, this algorithm supports single and multiple cluster heads. Simulation results show the superiority of our clustering algorithm over the other three well-known algorithms.**

Index Terms—**VANET, clustering, multi-head, head election, mobility.**

I. INTRODUCTION

Clustering is a technique to group nodes into several clusters. Each node in the cluster structure plays one of three roles: Cluster Head (CH), Cluster Gateway (CG), and Cluster Member (CM). A CH is a leading node of a cluster and is responsible to coordinate all CMs in its cluster. A CG is a border node of a cluster that can communicate nodes belonging to different clusters.

In mobile wireless networks, clustering is a practical skill to reduce the complexity of network management [1], [2], [3]. For example, CHs can allocate channel resource (time slots or frequency spectrums) to their CMs to avoid any happenings of transmission collisions and increase resource utilization within a cluster. Moreover, a virtual backbone network can be built by CHs and CGs to manage all routing jobs. This enables a scalable wireless routing protocol.

Clustering has many application domains. However, developing efficient clustering techniques in mobile environments is not an easy job. Node mobility will frequently destroy existing cluster structures. Reclustering overhead becomes an important cost metric.

In this paper, we consider a data sharing application in a Vehicular Ad hoc Network (VANET). VANET is a specialized Mobile Ad Hoc Network (MANET) that connects vehicles and roadside facilities. The major function of VANET is to provide real-time services and emergency warnings for drivers and passengers. VANET provides both Roadside-to-Vehicle Communication (RVC) and Inter-Vehicle Communication (IVC).

Our data sharing application is motivated by a typical scenario, where a passenger in a car would like to download an interested multimedia file from neighboring cars via IVC. Here, we use the cluster structure to facilitate the finding, uploading, and downloading of multimedia files. Vehicles that are willing to share data are grouped into clusters. In a cluster, CMs can upload their shared data and query interested data to the CH(s). CMs can also download interested data from the CH(s). To speed up data downloading, a bit-torrent downloading mechanism from multiple seed nodes (CHs) is recommended. Therefore, we need to construct a cluster with multiple CHs.

Several clustering algorithms [4]-[10] have been proposed for mobile networks. However, they have some weaknesses when applied to our data sharing application. First, these algorithms do not support an arbitrary number of CHs within a cluster. Second, most of them are designed for MANET but VANET. VANET has its own unique features such as highly dynamic topology, sufficient energy and storage, and geographical environment constrains. A mobile node in MANET can move in arbitrary directions but can only move along the street in VANET. Moreover, most vehicles are equipped with GPS (Global Positioning System) devices. The location and mobility information about a vehicle is available, which facilitates the design of a more efficient clustering algorithm.

We would first introduce our proposed single-head clustering algorithm by detailing the process of CH election. Then, a multi-head clustering algorithm is illustrated. The remainder of this paper is organized as follows. A brief survey on clustering algorithms in mobile environments is given in Section 2. Our proposed algorithm is presented in Section 3. We evaluate the performance in Section 4. Finally, we draw significant conclusions in Section 5.

II. RELATED WORK

A clustering algorithm includes two technical parts: cluster establishment and cluster maintenance. In cluster establishment, we need to identify the role (CH, CG, or CM) of each node in the network. The election of CHs is a core technique. In cluster maintenance, reclustering (cluster merge and split) needs to be performed to handle the effects of node mobility and node failure. An efficient clustering algorithm should take care of performance metrics such as cluster lifetime and reclustering overhead.

We classify existing clustering algorithms according to two categories: cluster structure and election criterion. From

Shou-Chih Lo and Jhih-Siao Gao are with the Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan (e-mail: sclo@mail.ndhu.edu.tw, d9721005@ems.ndhu.edu.tw).

Yi-Jen Lin was with the Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan (e-mail: evelyn0324@hotmail.com).

the structure point of view, clustering algorithms either have different topologies (overlapping or non-overlapping) or CH numbers. Most of clustering algorithms select one CH in a cluster. Two CHs are selected in [11]. A CG can belong to more than one cluster in overlapping clusters while a CG belongs to only one cluster in non-overlapping clusters. The distance between a CM and a CH is typically one-hop. Overlapping clusters (such as [12], [13], and [14]) and non-overlapping clusters (such as [15]) have the advantages of high robustness and low channel contention respectively.

Different evaluation methods can be applied in CH election and can be classified into five types. With an ID-based method, a node with the largest or smallest identification number (ID) in its neighborhood is elected to be a CH. With a degree-based method, a node with the largest degree value (indicating the number of its direct neighbors) in its neighborhood is elected as a CH. Local nodes with a competition-based method compete to be a CH. With a mobility-based method, a node that has stable relative mobility to its neighbors is elected as a CH. In a multiple-metrics-based method, several evaluation factors are combined together by a weighted average function to qualify a node for a CH.

In the following, we introduce some well-known clustering algorithms. LID (Lowest ID) [4] elects a node with the smallest ID as a CH. This approach is simple but may cause several reclusterings and small-size clusters. HCC (High Connectivity Clustering) [4] elects a node with the largest degree value as a CH. HCC creates a less number of clusters than LID, so the maintenance cost is low. However, HCC is more unstable than LID to node mobility. MMDA (Max-Min D-hop clustering Algorithm) [5] and RCC (Random Competition based Clustering) [6] use a competition method to elect CHs. WCA (Weighted Clustering Algorithm) [7] and CEMCA (Connectivity, Energy and Mobility driven Clustering Algorithm) [10] consider several factors such as ID, node degree, power level, mobility, and location in the CH election.

LCC (Least Cluster head Change) [8] improves LID and HCC on reclustering overhead. Only two conditions cause a new CH election: 1. Two clusters merge into one cluster. 2. A CM cannot listen to any activity of surrounding CHs. Instead of using IDs, MOBIC (MOBIlity metrics Clustering) [9] uses relative mobility in CH election to reduce the effect of node mobility. A mobile node that is relatively less mobile with respect to its neighbors is elected as a CH. The relative mobility is computed by the distance change of a node observed from the other node. MOBIC measures signal strength to derive the distance between two nodes. This measurement is simple but inaccurate.

DPP (Directional Propagation Protocol) [11] extends MOBIC into vehicular environments. Two CHs that are located at the head and the tail of a cluster are selected. CBMAC (Cluster-Based Medium Access Control) [16] uses three criteria in CH election: degree, position, and relative speed.

III. CLUSTERING ALGORITHMS

At first, the design of our single-head clustering algorithm is explained. Then, the multi-head version is introduced. Two assumptions are made: Each mobile node has a unique ID and is equipped with a GPS device.

A. Election Criterion

In a data sharing application, it is fairer for CMs that a CH is nearer the center of a cluster, because the hop count of a data transmission path from a CM to the CH is similar. Moreover, a CH should have stable relative mobility to its CMs for reducing CH reelection times. Based on these concepts of center position and relative mobility, we measure the RPM (*Relative Position and Mobility*) of each mobile node as the criterion of CH election.

Each mobile node is supposed to periodically broadcast a "HELLO" packet to its one-hop neighbors. This HELLO packet carries information about the location, motion vector, and RPM of a mobile node. By listening to these HELLO packets, a node maintains a neighbor table that records the mobility-related data of all its current one-hop neighbors (including the node itself). Clusters with nodes having diverse moving directions are unstable, so we restrict all nodes in a cluster to have the same moving direction. Therefore, each node records only those neighbors that drive with the same direction as it in its neighbor table.

If a node has m entries in its neighbor table, without loss of generality, we assume that the first entry records its own mobility-related data. Three data fields (x_i, y_i, v_i) are recorded in the ith entry, which indicate the current location (x_i, y_i) and the current moving speed (v_i) of the recorded node.

The RPM of a node is computed as the following steps:

1) Compute the center position from these m entries in the neighbor table as (1).

$$P_c = (x_c, y_c) = \frac{1}{m}(\sum_{i=1}^{m} x_i, \sum_{i=1}^{m} y_i) \qquad (1)$$

2) Compute the relative distance to this center position as (2).

$$Rel\_Dist_{1c} = |P_1 - P_c| = \sqrt{(x_1 - x_c)^2 + (y_1 - y_c)^2} \quad (2)$$

3) Sort these m moving speeds (scalar values) and find the median as (3).

$$v_c = Median\{Sort\{v_1, v_2, ..., v_m\}\} \qquad (3)$$

4) Compute the relative speed to the median speed as (4).

$$Rel\_Speed_{1c} = |v_1 - v_c| \qquad (4)$$

5) Compute the RPM (between 0 and 1) as (5).

$$RPM = \alpha \cdot \frac{Rel\_Dist_{1c}}{Max\{Rel\_Disc_{jc}, j \in 1 \sim m\}} + \\ (1-\alpha) \cdot \frac{Rel\_Speed_{1c}}{Max\{Rel\_Speed_{jc}, j \in 1 \sim m\}} \qquad (5)$$

α is a weighted factor which is set as 0.6 in our experiments. A mobile node declares itself a CH when its RPM is the smallest one in its neighbor table.

B. Cluster Establishment

In the cluster establishment phase, three node roles are used: UN (Undecided Node), CH, and CM. A UN is a node that is not currently belonging to any cluster. A node will play one of these three roles and may transit from one role to another role if cluster structure is changed.

The following notations are used in our latter discussion:
- BI (Broadcast Interval): time interval for a node to broadcast a HELLO packet.
- CI (Contention Interval): time duration after two encounter CHs start competing to be a CH.
- CD (Contention Distance): distance gap after two encounter CHs start competing to be a CH.
- TI (Timeout Interval): time duration after an unreachable neighbor is removed from a neighbor table.
- $|UN\_NUM_i|$: number of UNs in the neighbor table of node i.
- UN_BOUND: threshold value for a node to start the CH election.
- $|CM_i|$: number of CMs joining to a CH node i.
- Four control packets are used in our clustering algorithm:
- HELLO: periodically broadcast packet by each node that carries the mobility-related data, RPM, node role, and |CM| values.
- JOIN_INVITE: broadcast packet issued from a CH to invite any possible CMs.
- JOIN_REPLY: broadcast packet issued from a CM to acknowledge a join invitation from a CH.
- CH_RESIGN: broadcast packet issued from a CH when deciding to resign from a CH.

Cluster construction steps are explained below:
1) Initialize each node to be a UN.
2) Each node broadcasts a HELLO packet per BI.
3) A node i starts the CH election as $|UN\_NUM_i| \geq$ UN_BOUND. A node declares itself a CH when its RPM is the smallest one in its neighbor table. To break the tie, the smallest Rel_Dist, Rel_Speed, and ID are considered in sequence.
4) A new CH broadcasts a JOIN_INVITE packet to its neighbors.
5) A UN joins to a cluster when it receives a JOIN_INVITE packet from a CH that is driving with the same direction as it. This UN replies a JOIN_REPLY packet to the CH.

C. Cluster Maintenance

We individually explain the transition of each node role as follows. The role transition diagram is depicted in Fig. 1.
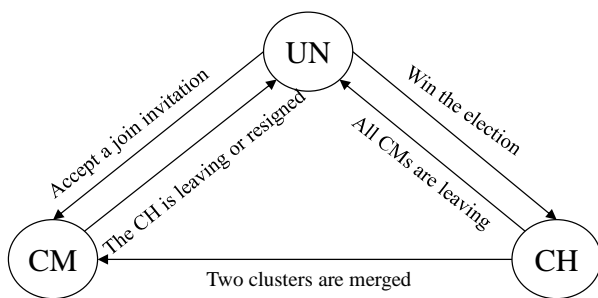

Fig. 1. Role transition diagram in a single-head cluster.

UN:
1) A UN becomes a CH if winning the election

2) A UN becomes a CM if receiving a JOIN_INVITE packet from a CH that is driving with the same direction as it.

CH:
1) Two CHs i and j will enter a competition process if the following conditions are true:
   (a) CHs i and j contact (can communicate with each other) for a time period greater than CI.
   (b) CHs i and j drive in the same direction.
   (c) The distance between i and j is smaller than CD.
2) In the competition process, the CH having the larger number of CMs wins the competition. That is, if $|CM_i| \geq |CM_j|$, then CH j resigns from a CH by broadcasting a CH_RESIGN packet to its neighbors. In this case, CH j becomes a CM of CH i.
3) A CH removes a CM from its CM list if it does not receive any HELLO packet from this CM over a time period of TI. If the CM list is empty, this CH returns as a UN.

CM:
1) If a CM does not receive any HELLO packet from its CH over a time period of TI, this CM returns as a UN.
2) A CM returns as a UN, if it receives a CH_RESIGN packet from its CH.

D. Multi-Head Clustering

The multi-head version is extended from the single-head version. We first construct clusters using the single-head algorithm. The selected CH in each cluster is called a master CH (MCH). Then, we select some CMs from a cluster to be slave CHs (SCHs). Therefore, each multi-head cluster contains one MCH and several SCHs. In a vehicular environment, a cluster usually has a rectangular shape that captures a certain road segment. Having the positions of all CMs, an MCH can compute the bounded rectangle that encloses all its CMs. Then, this rectangle is divided into some sectors (with the number being specified) as the example shown in Fig. 2. The MCH selects one SCH that has the locally smallest RPM in each sector (except for the sector the MCH is located in).
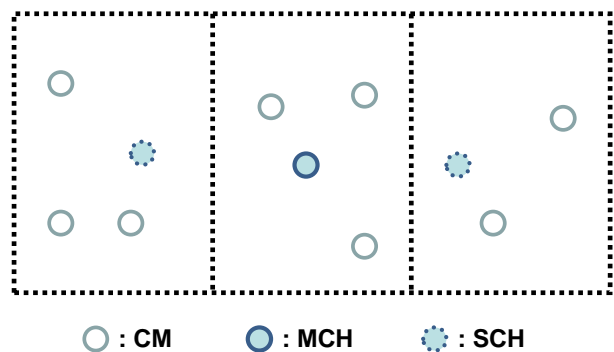

○ : CM   ◉ : MCH   ⦿ : SCH

Fig. 2. Multi-head election.

A new control packet SCH_ASSIGN is introduced, which is used by an MCH to announce the SCH list. A CM transits to a SCH when receiving a SCH_ASSIGN packet and finding that its ID is in the SCH list. The MCH periodically announces the SCH list by using the same HELLO packet. In the cluster maintenance, the procedures of UN, CH (as MCH), and CM remain unchanged. An additional node role of SCH

is introduced in our role transition diagram as shown in Fig. 3. Here, an SCH returns to as a CH when its MCH is leaving or resigned.
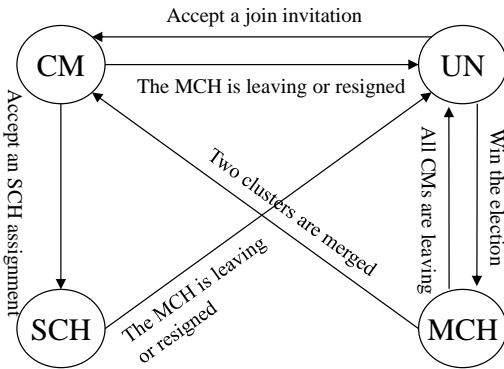

Fig. 3. Role transition diagram in a multi-head cluster.

IV. PERFORMANCE EVALUATION

To evaluate the performance of our proposed algorithm, we carry out simulations using NS-2 [17]. We compare our algorithm (abbreviated as CPM, Center-Position and Mobility) with LCC, LID, and MOBIC. The single-head version of CPM is concerned, since the other compared targets are all single-head approaches. The cost metrics used in our experiments are listed below:

Average number of clusters: average number of clusters by averaging system observations per ten seconds.

Average number of CMs: average number of CMs in a cluster by averaging system observations per ten seconds.

Average cluster lifetime: average lifetime of a cluster.

Average idle time: average time duration for a node remaining as a UN in the system.

Average resident time: average time duration for a CM to stay in the same cluster.

A. Simulation Model

We consider a real street environment which is imported from the TIGER (Topologically Integrated Geographic Encoding and Referencing) database [18]. A city street map of size 1500m   1500m is used as shown in Fig. 4. Each road has two lanes (forward and reverse). Under this street model, 200 vehicles are generated and their moving patterns are controlled by VanetMobiSim [19]. Each vehicle moves with a speed from 10 m/sec to 20 m/sec. The parameter settings in our simulation are listed in Table I. Each cost result is computed through the average of ten simulation runs.

Fig. 4. Simulated street environment.

TABLE I: PARAMETER SETTINGS IN EXPERIMENTS.

Parameter	Value
Simulation time	300 s
MAC protocol	802.11p
Transmission range (TR)	100 m – 250 m
Broadcast interval (BI)	0.5 s
Contention interval (CI)	4×BI
Timeout interval (TI)	2×BI
Contention distance (CD)	TR/2
UN_BOUND	1

B. Simulation Results

The experimental results are depicted in Figs. 5~9. LCC and LID are ID-based clustering algorithms. They will group vehicles on both forward and reverse lanes, so their constructed clusters tend to be large and hence the average number of clusters is small. These results are more significant if larger transmission ranges are used. However, under high node mobility, these two algorithms are not stable. The average cluster lifetime is short. LCC relaxes certain criteria on reclustering as compared with LID. The cluster lifetime of LCC is a little bit higher than that of LID. Since the CH election is simple for ID-based approaches, a UN can easily join to a surrounding cluster. Therefore, the average idle time is short. However, the cluster is not stable and tends to be reconstructed. A CM frequently joins to a different cluster, so the average resident time is short. This reduces the opportunity of a CM to successfully download data from CHs. As a conclusion, these ID-based clustering algorithms are not suitable for our data sharing application in vehicular environments.
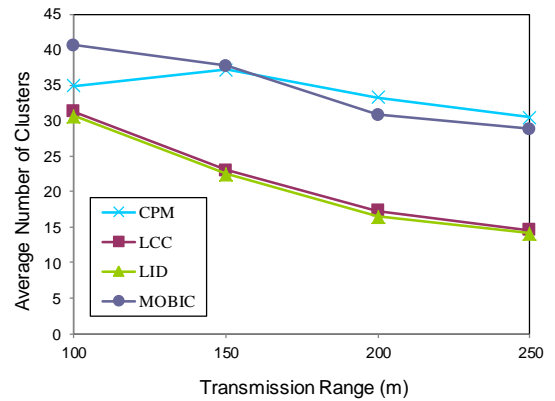

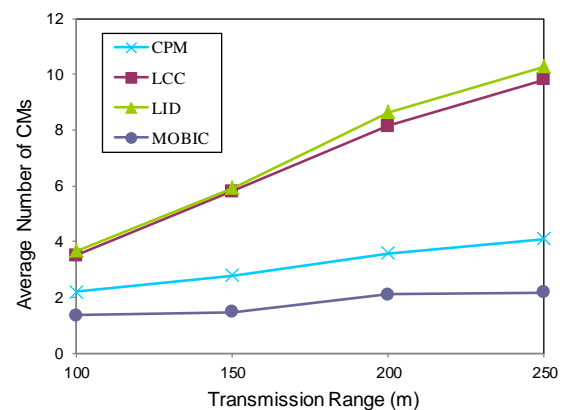Fig. 5. Average number of clusters vs. TR.

Fig. 6. Average number of CMs vs. TR.

Next, we compare the performances of MOBIC and CPM. MOBIC is a mobility-based clustering algorithm while CPM is a multiple-metrics-based one. As can been seen in the figures, MOBIC generates smaller clusters than CPM. Also, vehicles on forward and reverse lanes are not separated in MOBIC, so its cluster lifetime is shorter than CPM. Those vehicles that occasionally join to a CH with a different moving direction as them will frequently become UNs as the CH is leaving. Moreover, the CH election in MOBIC takes the most time among all. To compute the relative mobility of one node against the other node, two successive HELLO packets are required. Therefore, the average idle time in MOBIC is long. However, those vehicles that drive the same direction with the CH will keep joining the same cluster, so the average resident time of MOBIC is long. As a conclusion, MOBIC is still not suitable for our data sharing application, because small-size clusters and long-idle periods prevent the searching of a verity of shared data.

Compared with other algorithms, CPM provides stable clusters with long lifetime. Moreover, the short idle time and the long resident time qualify CPM for a good clustering algorithm in data sharing. The idle time of CPM is higher than LID and LCC, because we disallow any clusters with size of 1. There may have single UN in our system and these single UNs are useless to data sharing.
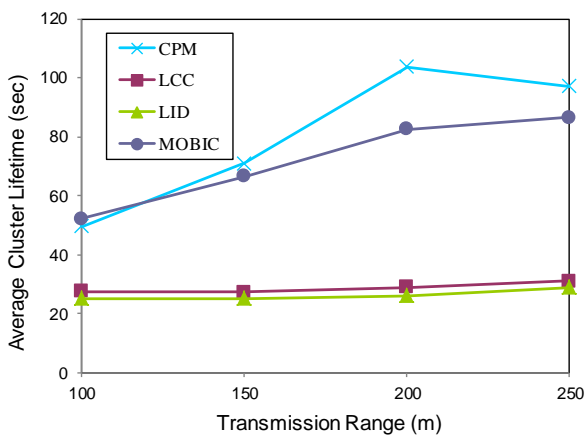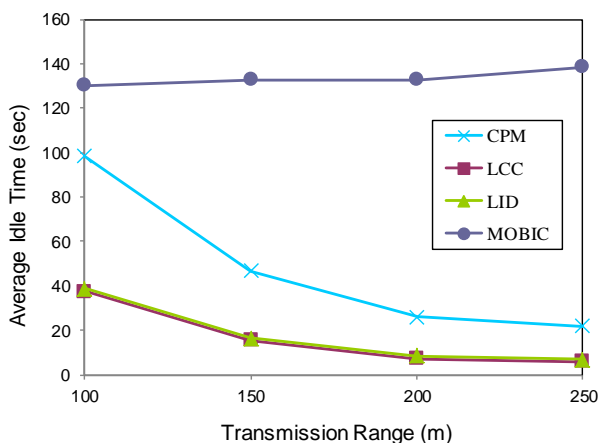

Fig. 7. Average cluster lifetime vs. TR.

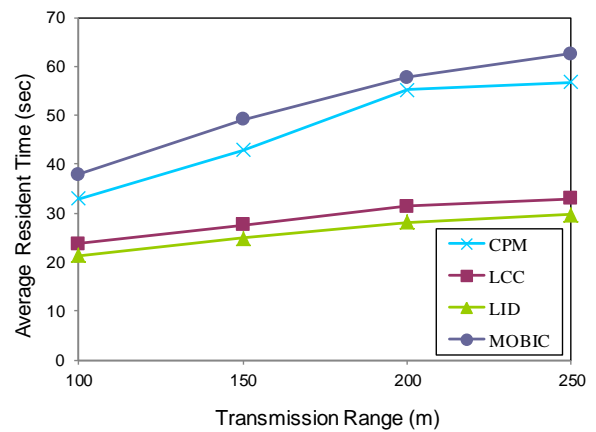

Fig. 8. Average idle time vs. TR.

Fig. 9. Average resident time vs. TR.

V. Conclusion

We have introduced the importance of clustering to network and application designs. For example, grouping nodes into cluster structures make a network topology easy to maintain, and grouping users of the same interests facilitates data sharing. In this paper, we consider a data sharing application in a vehicular environment. Vehicles nearby are grouped into a cluster in which some nodes are selected as cluster heads. These cluster heads serve as local file servers that enable surrounding nodes to upload and download shared data. We therefore propose a multi-head clustering algorithm.

Our proposed algorithm can select a given number of cluster heads with a uniform distribution in space. Since node mobility has a great influence on cluster maintenance, both node position and node mobility are considered in our cluster head election. Different node roles are identified with a clear role transition diagram. Compared with traditional clustering algorithms for mobile environments, we additionally consider the unique feature in vehicular environments that vehicles drive along the street either with the same or opposite driving direction. Our experimental results reveal that the proposed algorithm generates stable clusters with long lifetime.

In the future, we will consider more applications for our multi-head clustering technique. The cooperation behavior among multiple cluster heads need to be further discussed and evaluated.

Acknowledgment

References

[1] A. B. McDonald and T. F. Znati, "A mobility-based framework for adaptive clustering in wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1466-1487, August 1999.

[2] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265-1275, September 1997.

[3] T. C. Hou and T. J. Tsai, "A access-based clustering protocol for multihop wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 7, pp. 1201-1210, July 2001.

[4] M. Gerla and J. T. Tsai, "Multiuser, mobile, multimedia radio network," *Wireless Networks*, vol. 1, pp. 255-265, October 1995.

[5] A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh, "Max-min d-cluster formation in wireless ad hoc networks," in *Proc. IEEE Infocom Conf.*, pp. 32-41, March 2000.

[6] K. Xu and M. Gerla, "A heterogeneous routing protocol based on a new stable clustering scheme," in *Proc. Milcom Conf.*, pp. 838-843 vol. 2, October 2002.

[7] M. Chatterjee, S. K. Sas, and D. Turgut, "An on-demand weighted clustering algorithm (WCA) for ad hoc networks," in *Proc. IEEE Globecom Conf.*, pp. 1697-1701, November 2000.

[8] 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 Conf.*, pp. 197-211, April 1997.

[9] P. Basu, N. Khan and T. D. C. Little, "A mobility based metric for clustering in mobile ad hoc networks," in *Proc. Int. Conf. Distributed Computing Systems Workshop*, pp. 413-418, April 2001.

[10] F. D. Tolba, D. Magoni, and P. Lorenz, "Connectivity, energy and mobility driven clustering algorithm for mobile ad hoc networks," in *Proc. IEEE Global Telecommunications Conf.*, pp. 2786-2790, November 2007.

[11] T. D. C. Little and A. Agarwal, "An information propagation scheme for VANETs," in *Proc. IEEE Intelligent Transportation Systems Conf.*, pp. 155-160, September 2005.

[12] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," in *Proc. IEEE Infocom Conf.*, pp. 2652-2661, March 2004.

[13] P. Krishna, N. H. Vaidya, M. Chatterjee and D. K. Pradhan, "A cluster-based approach for routing in dynamic networks," *ACM Sigcomm Computer Communication Review*, vol. 27, no. 2, pp. 49-64, April 1997.

[14] T. Wu and S. Biswas, "A self-reorganizing slot allocation protocol for multi-cluster sensor networks," in *Proc. Int. Symp. Information Processing in Sensor Networks*, pp. 309-316, April 2005.

[15] J. Y. Yu and P. H. J. Chong, "3hBAC (3-hop between adjacent clusterheads): a novel non-overlapping clustering algorithm for mobile ad hoc networks," in *Proc. IEEE PACRIM Conf.*, pp. 318-321, August 2003.

[16] Y. Gunter, B. Wiegel, and H. P. Grossmann, "Cluster-based medium access scheme for VANETs," in *Proc. IEEE Intelligent Transportation*, pp. 343-348, September 2007.

[17] The network simulator. [Online]. Available: http://www.isi.edu/nsnam/ns/.

[18] Topologically integrated geographic encoding and referencing. [Online]. Available: http://www.census.gov/geo/www/tiger.

[19] Vanet Mobi Sim. [Online]. Available: http://vanet.eurecom.fr/.

Shou-Chih Lo received the B.S. degree in computer science from National Chiao Tung University, Taiwan, in 1993, and the Ph.D. degree in computer science from National Tsing Hua University, Taiwan, in 2000. Since 2004, he has been with the Department of Computer Science and Information Engineering of the National Dong Hwa University in Taiwan, where he is currently an Associate Professor. His current research interests are in the areas of mobile and wireless networks and delay-tolerant networks.

I-Cheng Lin received the B.S. degree in Computer Science from National Dong Hwa University, Taiwan, in 2009. Since 2009, she has been with the company of the Trend Micro in Taiwan, where she is currently a software engineer.

Jhih-Siao Gao received his B.S. and M.S. degree in Computer Science Information Engineering from Shu-Te University, Taiwan, in 2006 and 2008. Currently he is working towards the Ph.D. degree in Computer Science and Information Engineering at National Dong Hwa University, Taiwan. His research interests focus on wireless communications network.