

# A Cluster-head Selection and Update Algorithm for Ad Hoc Networks

Hao Wu and Zhangdui Zhong

State Key Laboratory of Rail Traffic Control and Safety  
Beijing Jiaotong University  
Beijing, China, 100044  
Email: {hwu, zdzh}@bjtu.edu.cn

Lajos Hanzo

School of ECS, University of Southampton  
Southampton, SO17 1BJ, UK  
Email: lh@ecs.soton.ac.uk  
<http://www-mobile.ecs.soton.ac.uk>

**Abstract**—A novel cluster-head selection and update algorithm “Type-based Cluster-forming Algorithm (TCA)” is proposed, which outperforms both the lowest node ID (LID) and the Weighted Clustering Algorithm (WCA) in the *ad hoc* network scenario considered. The system’s performance is investigated in a scenario, when the 50 communicating nodes belong to three different groups, for example, a group of rescue workers, fire-fighters and paramedics. It is demonstrated that the carefully designed protocol is capable of outperforming the above-mentioned benchmarks both in terms of a reduced number of cluster-head updates and cluster-change events. Hence its quality-of-service may be deemed higher.

## I. INTRODUCTION

Numerous clustering algorithms based on different optimization objectives have been proposed [1–7]. The most popular *ad hoc* clustering algorithms may be classified as follows.

1) *Lowest ID (LID) clustering algorithm*: The authors of [4] [5] proposed the idea of assigning a unique ID to each node in the network and then broadcasting the ID to all the neighbour nodes. In the next step, the IDs that were error-free decoded are compared and the node having the lowest ID is chosen as the cluster-head, while its single-hop neighbours become the cluster members. If a node belongs to multiple clusters, it may be viewed as a gateway between clusters. This algorithm is appealingly simple and the construction of clusters may be promptly completed, but the number of cluster-heads may become undesirably high. Furthermore, owing to the uneven distribution of nodes within a cluster, the packet delivery delay may become excessive. Finally, the selection of cluster-heads has to be frequently updated, which may be expected to impose a high control traffic load.

2) *The highest-degree (HD) clustering algorithm*: The goal of the HD algorithm [6] is to minimize the number of clusters, which is achieved as follows. Each node is aware of the number of its neighbour nodes, which acquired by interactively exchanging control messages. The node having the highest number of neighbours, i.e. the highest degree, is elected as the cluster-head. If the degree of several nodes is the same, the lowest-ID node becomes the cluster-head. Then the one-hop neighbour nodes of the cluster-head become ordinary members of the cluster. The above procedure is then repeated until all nodes join in a cluster. The number of cluster-heads is relatively low in this algorithm, hence it potentially reduces the average number of hops between the source node and the destination node. Furthermore, it also reduces the packet delivery delay. However, when using this principle, the geographic distribution of cluster-heads may become undesirable. When the node mobility increases, the network-load imposed by the increased cluster-head update increases the network’s maintenance overheads.

3) *Weighted clustering algorithm (WCA)*: The WCA takes into consideration the “ideal” degree (ideal cluster size)  $d$  formulated in

terms of striking the required trade-off amongst, numerous conflicting design factors, such as the transmission power, mobility and battery power of mobile nodes [7]. Then it assigns to each of these factors the appropriate weight before deciding on the suitable degree and on the choice of the cluster-head. The above-mentioned weight value is an important design factor and may be adaptively controlled according to the actual network conditions.

The design of cluster-forming algorithms have to be closely matched to the specific application environment considered, hence diverse cluster-forming algorithms may be applied to cater for different environments, in order to improve the attainable network performance. The existing cluster-forming algorithms typically assume that the nodes are unconnected at the commencement of communications, although this may not be the case. The class of emergency response system constitutes a typical application of *ad hoc* networks, such as those in battlefields, in disaster and rescue scenarios, etc. The mobile nodes typically have to carry out a task as a collaborative group [8] [9], but the nodes may be owned by different proprietors in the same geographic area. For example, fire-fighters, paramedics, soldiers might converge on the same area to carry out a rescue mission. Once the mission was accomplished, they will depart. When using a classic cluster-forming algorithm in such a scenario, the cluster-head selection may be random. When the nodes join or leave the zone of deployment, this typically occurs under independent instruction, which may lead to frequent cluster-head updates. This may even lead to network segmentation and to a sharp network performance degradation.

In order to reduce the update frequency of cluster-heads, we propose a type-based cluster-forming algorithm (TCA) for employment in emergency *ad hoc* networks. The rest of this paper is organized as follows. The basic philosophy of the proposed TCA is described in Section II. Section III elaborates on the calculation of the algorithm’s parameters. Section IV discusses the attainable performance of our TCA compared to the conventional LID and WCA algorithms, while, Section V concludes the paper.

## II. DESCRIPTION OF THE TYPE-BASED CLUSTER-FORMING ALGORITHM

### A. Basic idea

Let us commence by stipulating the following assumptions.

- 1) Each node is aware of its exact location information, which is ensured by a GPS-based or other positioning device;
- 2) Each node is equipped with an omni-directional antenna and operated in half-duplex mode;
- 3) An idealized zero-delay channel access protocol is used and the propagation delay is also assumed to be zero;

- 4) During the cluster-forming establishment stage, the network's topology does not change.

Type-based cluster-forming algorithm is eminently suitable for *ad hoc* emergency networks, where the nodes tend to move in a concerted action as a group. The algorithm typically employs a so-called stability factor  $S$  as the parameter to enable/activate the cluster-head election process. The stability factor  $S$  combines a diverse range of node features into a single factor, such as the relative mobility of nodes with respect to each other, their average distance, the degree of node connectivity, the residual battery power level, etc. It also indicates, whether a node is suitable for the role of cluster-head. When forming a cluster, it is more convenient to assign nodes of the same type - such as paramedics or fire-lighters - into a single cluster within a certain geographic range. The TCA includes the cluster-formation stage and cluster-update stage.

*B. Cluster-formation stage*

During the cluster-formation stage, each node is assigned a unique ID, which may be based on the above mentioned lowest ID (LID) algorithm, since the LID is a relatively simple, efficient and low-overhead technique. The mobile nodes periodically broadcast their status information using the HELLO message [10], and maintain the status information received from the nodes in the neighbourhood list. If an ordinary node was already assigned to a cluster, it can no longer participate in the cluster-head election process. However, an ordinary node is allowed to belong to multiple clusters, and if it does, it may be used as a gateway node, which is responsible for data relaying between clusters.

*C. Cluster-update stage*

The cluster-head elected during the cluster-formation stage may not remain the best choice for long in an emergency *ad hoc* network. Hence we have to monitor the performance of the cluster-head by periodically updating the stability factor  $S$  as follows.

**Step 1:** In the cluster-update stage each node periodically broadcasts the HELLO message to its neighbours, which is shown below.

ID	IP	Status	P	S
----	----	--------	---	---

ID: unique node identifier in the network;

IP: unique IP address in the network, with the format of  $\langle subnet\ identifier, host\ identifier \rangle$ , which is used for routing and distinguishing the node type;

Status: node status information, where the value 0 indicates that a node has not as yet joined any cluster; the value 1 indicates an ordinary node within a cluster; the value 2 represents a gateway node; value 3 means a cluster-head;

P: node location information;

S: node stability factor; each node builds up a neighbourhood list with the aid of the HELLO messages sent by its neighbours, assuming that the neighbours of Node A are  $B_1, B_2, \dots$ , where the explicit format is shown in Table I.

The neighbourhood list is periodically updated according to the HELLO messages received, where the interpretation of the each field is:

Neighbour ID: denotes the neighbour node's ID;

Neighbour IP address: with the format of  $\langle subnet\ identifier, host\ identifier \rangle$ , the *subnet identifier* indicates the logical or administrative affiliation of a node, where a given subnet represents the nodes belonging to the same logical rescue group or fire-fighting group, for example.

Neighbour status: at the commencement of network initialization, the neighbour status is assigned a value of 0, and when the clustering formulation is completed, the neighbour status may be 0,1,2 or 3 according to the actual situation;

Distance from neighbours: given the node location information, the cluster-head periodically calculates the distance from its neighbours, and the distance-history is saved as a chain list, such as  $D_{A,B_1}^1, D_{A,B_1}^2, \dots$ , which is updated in accordance with the FIFO principle. Then, this history list may be used as a parameter invoked for calculating the stability factor  $S$  during the cluster-head election stage.

The cluster-head update cycle is defined as  $N$  times the HELLO\_PERIOD  $H_p$  which represents the the broadcast cycle duration of the HELLO message and the value of  $N$  depends on the specific network topology considered. During every  $N \times H_p$  period, each mobile node calculates the relative mobility of  $M$ , the average sum of distances  $D_{sum}$  based on the location information of all neighbour nodes and node types, as well as on the stability factor  $S$  depending on  $M, D_{sum}$ , the connectivity degree  $d$  of the nodes and the node's remaining battery power  $E$ . The specific calculation of each parameter will be discussed in Section III;

**Step 2:** the mobile nodes send the value of  $S$  to their cluster-head. The cluster-head arranges the cluster members in an increasing order of  $S$  and reassigns the node IDs. The lower the stability factor  $S$ , the more likely that a node will be elected as a cluster-head. A node associated with a lower value of  $S$  indicates that it has sufficient battery power, a low mobility and a reliable connectivity associated with a high channel quality. Hence the lowest ID should be assigned to the node having the lowest value of  $S$ . If two or more nodes have the same value of  $S$ , the IDs should be assigned in accordance with the original ID sequence, as detailed in Table II. The new ID of a gateway node is assigned by its cluster-head having the lowest ID.

**Step 3:** the cluster-head sends the new IDs to its nodes within the cluster.

**Step 4:** the mobile nodes update their IDs and then reinitiate the cluster-formulation process.

Fig. II-C portrays the node-ID reassignment process within a cluster. More specifically, Fig. 1(a) shows the current node IDs within a cluster, while Fig. 1(b) displays the new node IDs after ID-update. Again, Table II shows the update of node IDs based on the value of  $S$ .

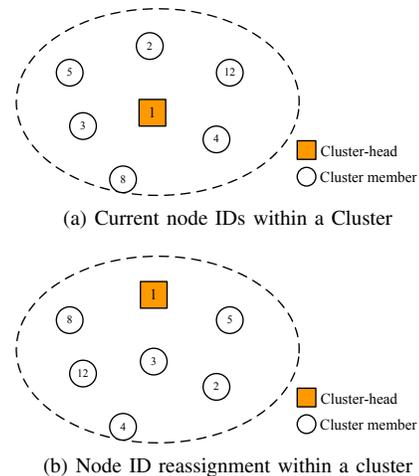


Fig. 1. Illustration of node IDs reassignment

TABLE I  
NEIGHBOUR LIST OF NODE A

Neighbour ID	Neighbour IP address	Neighbour status	Distance from neighbour			
			$D_{A,B_1}^1$	$D_{A,B_1}^2$	$\dots$	$D_{A,B_1}^N$
$B_1$	***.***.***.***	0	$D_{A,B_1}^1$	$D_{A,B_1}^2$	$\dots$	$D_{A,B_1}^N$
$B_2$	***.***.***.***	0	$D_{A,B_2}^1$	$D_{A,B_2}^2$	$\dots$	$D_{A,B_2}^N$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$

TABLE II  
NODE ID REASSIGNMENT IN TCA

ID number	$S$	New ID number
1	3	3
2	1.9	1
3	4.6	12
4	1.9	2
5	4.3	8
8	3.6	4
12	3.9	5

### III. CALCULATION OF PARAMETERS

Again, the nodes are assumed to be aware of their own location coordinates and each node is assigned a unique IP address in the network obeying the format of  $\langle \text{subnet identifier}, \text{host identifier} \rangle$ , which is used both for routing and for distinguishing the node types. The *subnet identifier* indicates the logical affiliation or group a node belongs to, while the *host identifier* indicates the unique description of the host within its own affiliation or group.

#### A. The relative mobility $M$

The relative mobility  $M$  is representative of the relative speed between a node and its neighbour nodes. A lower value of  $M$  indicates a lower relative mobility with respect to a neighbour node, thus a low-mobility node is more suitable for election as a cluster-head. For example, observe in 2 that Node  $A$  has  $m$  neighbour nodes, i.e. we have  $B_1, B_2, \dots, B_m$ . The required calculations are as follows.

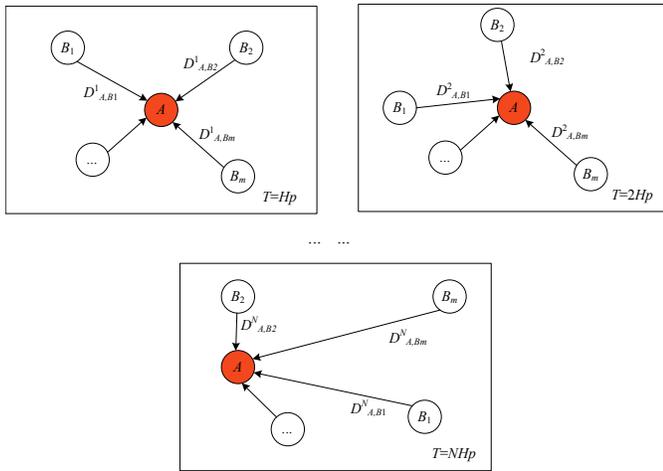


Fig. 2. Illustration of the average distance calculation of Node A

- 1) Calculate the average distance  $\mu$  between Node  $A$  and its neighbour  $B_1$  as:

$$\mu = (D_{A,B_1}^1 + D_{A,B_1}^2 + \dots + D_{A,B_1}^N)/N, \quad (1)$$

where  $D_{A,B_1}^1$  is the distance between Node  $A$  and  $B_1$  during the first  $H_p$  cycle, while  $D_{A,B_1}^2, \dots, D_{A,B_1}^N$  have similar definitions for the  $H_p$  cycles of  $2, \dots, N$ .

- 2) Calculate the variance of the distance between Node  $A$  and the neighbour node  $B_1$  during the  $H_p$  cycles as:

$$\delta_{A,B_1} = [(D_{A,B_1}^1 - \mu)^2 + (D_{A,B_1}^2 - \mu)^2 + \dots + (D_{A,B_1}^N - \mu)^2]/N, \quad (2)$$

- 3) Based on Equations (1) and (2), calculate the distance variance between Node  $A$  and each of its neighbour nodes, and then determine the weighted average  $M_A$ , which is referred to as Node  $A$ 's relative mobility:

$$M_A = (\omega_1 \delta_{A,B_1} + \omega_2 \delta_{A,B_2} + \dots + \omega_m \delta_{A,B_m})/m, \quad (3)$$

where,  $m$  is the number of neighbour nodes for Node  $A$ ; the weighting coefficient  $\omega$  is determined on the basis of the neighbour's IP address, by deciding whether the subnet identifier is the same as Node  $A$ 's. If so, then this indicates that Node  $A$  and the neighbour have the same affiliation or group ID. Hence we set  $\omega = \lambda$  ( $0 < \lambda < 1$ ), where  $\lambda$  is a constant, which may be adjusted, depending on the specific requirements of the network. Otherwise we set  $\omega = 1$ .

#### B. The average sum of distances $D_{sum}$

The variable  $D_{sum}$  represents the average sum of distances between a potential cluster-head and all its neighbours. A lower  $D_{sum}$  implies that the neighbour nodes are nearer to the potential cluster-head, and thus this node is indeed eligible as a cluster-head. The latest location information stored in the neighbour list is used in the calculation of  $D_{sum}$  as follows:

$$D_{sum-A} = (\omega_1 D_{A,B_1}^N + \omega_2 D_{A,B_2}^N + \dots + \omega_m D_{A,B_m}^N)/m, \quad (4)$$

where the determination of the weighting coefficient  $\omega$  was detailed in Section III-A.

#### C. The degree of connectivity $d$

The node connectivity degree  $d$  represents the number of single-hop neighbours, hence the degree of connectivity for Node  $A$  is:

$$d_A = \sum_{i=1}^m \frac{1}{\omega_i}, \quad (5)$$

where again, determination of the weighting coefficient  $\omega$  was described in Section III-A. Equation (5) indicates that if Node  $A$  and its neighbours belong to the same logical subnet, the degree of connectivity for Node  $A$  is increased by an amount of  $1/\omega = 1/\lambda$ , otherwise it by an amount of  $1/\omega = 1$ . Naturally, we have  $1/\lambda > 1$ .

#### D. The remaining energy $E$

We introduce the simplifying assumption that the energy consumption of each node imposed, while transmitting and receiving packets is identical. Hence the main difference in energy dissipation is that cluster-heads dissipate extra energy, while relaying the messages of

ordinary nodes. Thus, the total remaining energy of Node  $A$  may be expressed as:

$$E_A(t + \Delta t) = E_A(t) - E_A(\Delta t), \quad (6)$$

$$E_A(\Delta t) = P \cdot \Delta t, \quad (7)$$

where  $P$  is average power consumption of the cluster-head per unit time;  $\Delta t$  is the duration of time, while the node is acting as a cluster head. Based on the assumption that cluster-heads dissipate significantly more energy than ordinary nodes which in communicate with the value of  $m$ , each node calculates its own energy consumption for the period of  $\Delta t$ , and then directly determines the node's remaining energy.

#### E. The stability factor $S$

The stability factor  $S$  is used to judge, whether a node is suitable for acting as a cluster-head. The lower the stability factor  $S$ , the more suitable the node to act as a cluster-head, and vice versa. The stability factor  $S$  of Node  $A$  is calculated as:

$$S_A = \alpha_1 M_A + \alpha_2 D_{sum-A} + \alpha_3 / E_A + \alpha_4 |d_A - d|, \quad (8)$$

where we have  $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$ ,  $0 < \alpha_1, \alpha_2, \alpha_3, \alpha_4 < 1$ , with  $\alpha_1, \alpha_2, \alpha_3$  and  $\alpha_4$  representing the weight of the corresponding parameters in the cluster-head election process. Furthermore,  $d$  represents the "ideal" degree of node connectivity or the ideal cluster size, hence this factor has an influence on limiting the size of the cluster. They may be set to appropriate values based on the specific needs of the applications considered.

### IV. SIMULATION RESULTS

In this section, we characterize our TCA and benchmark it against the LID and WCA algorithms. We assume that 50 mobile nodes are randomly and uniformly distributed in a  $(100 \times 100)$  unit area. The duration of the simulated communication session was set to 600 time-units, where a time-unit is the period of  $N \times H_p$ . We invoke the group mobility model of [11], where the minimum speed of the nodes is 0, while the maximum speed ranges from 1 to 20 unit-distance/unit-time. The 50 nodes are classified into three groups: a rescue group, a fire-fighting group and a paramedical group. The number of nodes in each group is 20, 20 and 10. The initial battery energy of each node is identical, represented by  $E_0 = 100$  energy units. Furthermore, we have  $P = 0.1$  unit-energy/unit-time,  $\lambda = 0.6$ ,  $\alpha_1 = 0.5$ ,  $\alpha_2 = 0.25$ ,  $\alpha_3 = 0.05$ ,  $\alpha_4 = 0.2$ , and the ideal degree of connectivity is  $d = 10$ . The parameters of the WCA benchmark will set to be the same as those of our TCA. The results of 100 simulations, each having a duration of 600-unit-time, were averaged. By varying the maximum speed and transmission range of the nodes, we evaluated the performance of the above-mentioned three algorithms. When varying the nodes' maximum speed, the transmission range of the nodes was fixed to 30 distance-units. By contrast, upon varying the transmission range, the maximum speed was fixed to 10 velocity-units.

Our performance evaluations include the following metrics:

- 1) *The average number of cluster-heads* was calculated for 100 sessions, each representing a separate 600 time-units real-time communications session;
- 2) *The number of cluster-head updates per unit time* characterizes to the relative frequency of a cluster-head becoming an ordinary node or an ordinary node becoming a cluster-head;
- 3) *The number of nodes change events per unit time* which characterizes the relative frequency of cluster members traversing between different clusters.

Fig. 3 shows that the average number of cluster-heads varies as a function of the maximum node velocity. Our TCA exhibits a lower sensitivity to the node velocity than the LID and WCA. The improved resilience against node velocity fluctuations is attributed to the quantitative consideration of the relative node-mobility and long-term node-stability measures. In the TCA, we limited the size of each cluster, therefore the number of cluster-heads was relative stable, when the maximum velocity is varied.

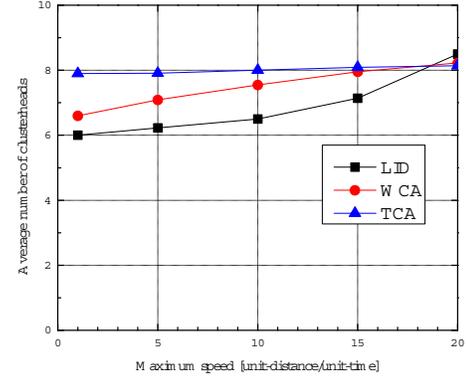


Fig. 3. Average number of cluster-heads vs maximum node speed

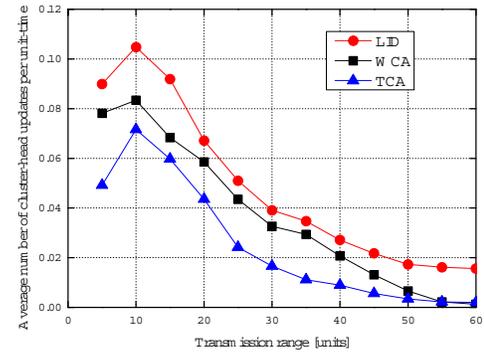


Fig. 4. Average number of cluster-head updates per unit time vs transmission range

The average number of cluster-head updates per unit-time is related to the transmission range, as shown in Fig. 4. Upon increasing the maximum velocity beyond about 10, the cluster-head remains capable of communicating with its cluster members. The TCA has a lower cluster update rate than LID and WCA.

Fig. 5 describes the relationship between the average number of cluster change events per unit-time versus the transmission range. Again, as the transmission range increases, the average number of cluster change events decreases monotonically. Since our TCA considers the stability of the cluster-heads, while minimizing the distance between the cluster-head and its cluster members, it reduces the number of cluster change events, hence rendering the cluster structure more stable. Similarly, compared to the LID and WCA, our TCA reduces the number of cluster-change events.

As the node speed is increased, the nodes roam more often outside the coverage range of their cluster-head, hence the cluster structure becomes more unstable. Similarly, the number of cluster-head updates and the cluster-change events become increasingly more frequent. Observe in Fig. 6 and Fig.7 that our TCA significantly improves

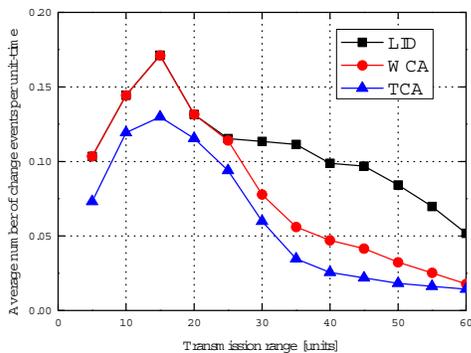


Fig. 5. Average number of cluster changes vs transmission range

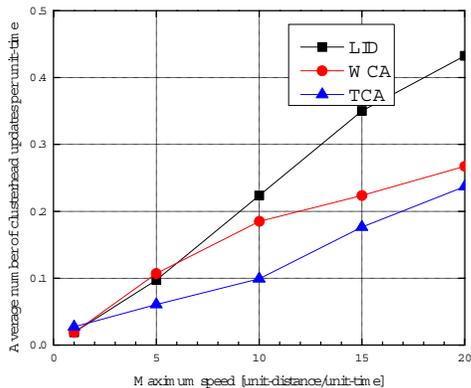


Fig. 6. Average number of cluster-head updates per unit time vs maximum node speed

the stability of the cluster structure, hence reducing the frequency of cluster-head update events, as the node mobility is increased.

### V. CONCLUSIONS

In this paper, we proposed a cluster-head selection and update algorithm for *ad hoc* emergency networks, where the nodes tend to move in a concerted action as a group. By employing the stability factor  $S$  as the parameter to enable/activate the cluster-head election process, it was more convenient to assign nodes of the same type into a single cluster within a certain geographic range when forming a cluster. The simulation results showed that our TCA outperformed the LID and WCA, since it reduced both the number of cluster-head update events and cluster change events. Hence its quality-of-service may be deemed higher. In the future, we would explore a more realistic joint system to improve the cluster-forming and update with the aid of the fuzzy controller.

### ACKNOWLEDGMENT

This paper is supported by Program for Changjiang Scholars and Innovative Research Team in University under Grant No. IRT0949 and the Joint State Program of the National Natural Science Foundation of China (NSFC) and the National Railway Ministry of China under project grant number 60830001. This paper is also supported by the State Key Laboratory of Rail Traffic Control and Safety under key project grant number RCS2008ZZ007. Furthermore, the financial support of the EPSRC UK and that of the EU under the auspices of the Optimix Project are gratefully acknowledged.

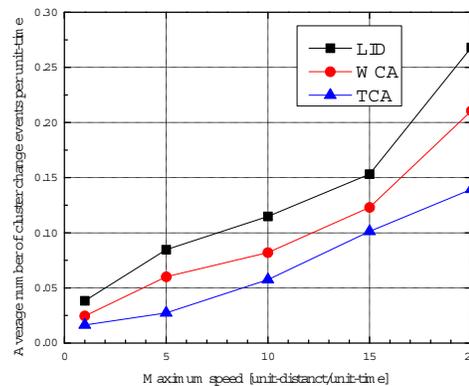


Fig. 7. Average number of cluster change events per unit time vs maximum node speed

### REFERENCES

- [1] Y. Xu and W. Wang, "Topology stability analysis and its application in hierarchical mobile ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1546 – 1560, Jun. 2009.
- [2] C. C. Tseng and K. C. Chen, "Organizing an optimal cluster-based ad hoc network architecture by the modified quine-mccluskey algorithm," *IEEE Communications Letters*, vol. 11, no. 1, pp. 43–45, Feb. 2007.
- [3] X. Zhang, H. Su, and H.-H. Chen, "Cluster-based multi-channel communications protocols in vehicle ad hoc networks," *IEEE Journal on Wireless Communications*, vol. 13, no. 5, pp. 44–51, Nov. 2006.
- [4] C. R. Lin and M. Gerla, "A distributed architecture for multimedia in dynamic wireless networks," in *Proceeding of IEEE Global Telecommunications Conference, GLOBECOM '95.*, vol. 2, 13-17 Nov 1995, pp. 1468 – 1472.
- [5] —, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, Sep. 1997.
- [6] M. Gerla and T. Tsai, "Multicluster, mobile, multimedia radio network," *Wireless Networks*, vol. 1, no. 3, pp. 255–265, 1995.
- [7] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: A weighted clustering algorithm for mobile ad hoc network," *Cluster Computing*, vol. 5, no. 2, pp. 193–204, Apr. 2002.
- [8] Y. Yi, M. Gerla, and K. Obraczka, "Scalable team multicast in wireless ad hoc networks exploiting coordinated motion," *Ad Hoc Networks*, vol. 2, no. 2, pp. 171–184, Apr. 2004.
- [9] G. Pei, M. Gerla, and X. Hong, "LANMAR: Landmark routing for large scale wireless ad hoc networks with group mobility," in *First Annual Workshop on Mobile and Ad Hoc Networking and Computing, MobiHOC. 2000*, 11 Aug. 2000, pp. 11–18.
- [10] S. Basagni, M. Conti, S. Glordano, and I. Stojmenovic, *Mobile Ad Hoc Networking*. Wiley-IEEE PRESS, 2004.
- [11] X. Hong, M. Gerla, G. Pei, and C. Chiang, "A group mobility model for ad hoc wireless networks," in *Proc. of the ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM)*, Aug. 1999, pp. 53–60.