A New Aggregate Local Mobility (ALM) Clustering Algorithm for VANETs

Evandro de Souza
University of Alberta
Department of Computing Science
Edmonton, Alberta, Canada T6G 2E8
Email: evandro@cs.ualberta.ca

Ioanis Nikolaidis
University of Alberta
Department of Computing Science
Edmonton, Alberta, Canada T6G 2E8
Email: yannis@cs.ualberta.ca

Pawel Gburzynski
University of Alberta
Department of Computing Science
Edmonton, Alberta, Canada T6G 2E8
Email: pawel@cs.ualberta.ca

***Abstract*—We present a beacon-based clustering algorithm aimed at prolonging the cluster lifetime in VANETs. We use a new aggregate local mobility criterion to decide upon cluster re-organisation. The scheme incorporates a contention method to avoid triggering frequent re-organisations when two *clusterheads* encounter each other for a short period of time. Simulation results show a significant improvement of cluster lifetime and reduced node state/role changes compared to previous popular clustering algorithms.**

***Index Terms*—Vehicular Ad Hoc Networks, Clustering Algorithms, Wireless.**

I. Introduction

A Vehicular Ad Hoc Network (VANET) is a specialised Mobile Ad Hoc Network (MANET) whose objective is inter-vehicle communication, usually with the intention of providing services related to road traffic, like dissemination of safety and traffic condition messages, control of vehicle flow formations, and also including Internet connectivity [1]. A vehicle is a robust and stable platform which can easily afford the kind of processing, sensing, storage, and energy capabilities necessary to support nontrivial wireless applications. Thus, it offers considerable advantages compared to other MANET platforms, because the applications are not going to be constrained by the energy requirements or weight of the requisite equipment.

The idiosyncratic characteristics of an inter-vehicle communication network call for solutions specifically crafted to the environment. As pointed out by Li *et al.* [2], solutions procured for general MANETs are either not going to work, or are bound to exhibit grossly substandard performance, because of these properties of vehicular systems:

- Highly dynamic topology due to vehicle speed
- Frequent disconnection of the network, especially when vehicle density is low
- No need to obsessively save on energy, storage, or processing power
- Propensity for geographic-oriented applications
- Predictable (highly correlated) dynamics allowing for effective modeling and prediction
- Hard delay constraints due to high mobility

Similar to MANETs, VANETs assume no specialised hardware infrastructure to control network stability. They require that regular nodes (running embedded protocols) execute functions normally carried out by routers. As the number of highly mobile nodes increases, the limitations of controlling such networks become apparent. Specifically, flat routing schemes exhibit serious scalability problems (caused by the fragility of paths built of highly mobile members), and hidden terminals adversely impact data transfers. Thus, one would like to identify within the volatile topology of the vehicular nodes some properties that remain stable for relatively long periods of time and use them as the basis of forming useful protocols.

A popular technique of introducing stable structures in otherwise unstable networks, including VANETs, is to allow the mobile nodes to self-organise into *c*lusters. Those clusters then reduce the burden of running higher-level protocols on the global population of individual nodes. For example, they provide a hierarchy within the network which helps with routing. In addition, clustering can be used as a means to increase node coordination and decrease the number of nodes interfering with each other, and also to mitigate the hidden terminal problem.

Considerable work, discussed in the next section, has gone into the development of clustering algorithms that would be suitable for highly dynamic topologies. We note that since the purpose of clustering in VANETs is to imbue the system with a stable underlay, the figure of merit for clustering ought to be the relative resilience of clusters at various speeds and traffic scenarios. In urban environments or on highways, the vehicular dynamics of motion create groups of vehicles moving relativity close together; this is called *platooning* [4]. These platoons of vehicles frequently pass one another in opposite directions or mingle on traffic lights. On such occasions, it is important for the clustering algorithm to maintain stability as best as possible, giving preference for groups of nodes with better cohesion. One way to measure the cohesion is by tracking the aggregate mobility [5].

In our paper, we present a new beacon-based clustering algorithm whose objective is to extend the lifetime of a cluster. The key component of our approach is a variant of the so-called Aggregate Local Mobility (ALM) measure as the criterion triggering cluster re-organisation. We also incorporate a contention–based scheme to prevent over-eager re-organisation of clusters when two *clusterheads* accidentally get in each other's range for a short period of time.

Our simulation studies strive to capture as much of real-life vehicle traffic scenarios as possible. For this leg of the model, i.e., to simulate vehicle traffic, we use the SUMO (Simulation of Urban Mobility) system [13]. The networked component of the simulator has been programmed in SMURPH with wireless extensions [15] yielding a high fidelity model of wireless communication. Our experiments show that the proposed scheme brings in a significant increase in the cluster lifetime, thus improving network stability and, in particular, decreasing the number of situations when nodes have to change their status within the cluster.

II. Previous Work

Clusters are conceptual structures where several nodes self-organise into a group around their momentarily selected representative called the *clusterhead*. This special node subsequently assumes the role of coordinator for the remaining members of the cluster. Depending on the scheme, the clusterhead may act as a relay for intra-cluster communication and/or a gateway interfacing other cluster members to nodes in other clusters. More generally, it often makes better sense to delegate the role of gateways to other members of the cluster, depending on their proximity to other clusters. This is shown in Fig. 1 where we identify three basic states/roles of a node: *clusterhead* (CH), regular *member node* (MN), and *gateway* (GW). Sometimes, an additional state called *Undecided* (UN) is used for the initial state of a node. According to Yu *et al.* [3], clusters provide three basic benefits: 1) spatial reuse of resources, 2) emergence of a virtual backbone, 3) improved network stability and scalability from the viewpoint of a regular member, which only sees and communicates with nodes inside its own cluster.

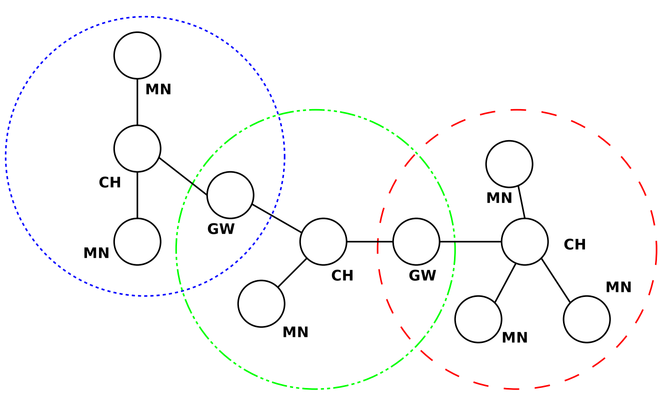


Fig. 1. A configuration of clusters

Clustering algorithms for MANETs have been extensively studied in [6], [3], [7], [8]. However, those studies do not address the case of vehicular movement. The most popular model of mobility assumed in the vast majority of research on MANETs is the Random Waypoint Model, which, basically, moves the nodes completely at random. Its alternatives, e.g., the Gauss-Markov Mobility Model from [9], do essentially the same thing in a slightly different way. Their common attribute is the complete failure to capture the strong correlations inherent in any realistic pattern of vehicular movement. Vehicles do not change randomly their direction at any instant or take a direction than does not agree with the street topology.

It is inevitable that the highly dynamic topology of VANETs will disturb cluster formation and re-organisation, increasing the cluster instability. Therefore, a clustering algorithm must strive to maintain cluster stability and retain the cluster contents and structure for as long as possible, as otherwise, the frequent re-clustering processes will degrade the performance of communication. Cluster stability in VANETs can be improved by taking advantage of the specific limitations and correlations imposed on vehicular mobility.

Most clustering approaches for MANETs embrace the dominant role of the CH on the formation and maintenance of the cluster. According to Gunter *et al.* [10], the basic techniques are: Lowest-ID, Highest-Degree, and Beacon-Based. The first two are inconvenient in VANET environments. The third approach, proposed by Santos *et al.* [11], is based on regular beacon transmissions or *hello* messages, which advertise the state of the node. Based on the state of the neighbours (i.e., nodes in its transmission range), a node can select its own state. The main purpose of this approach is to try to minimise cluster changes. A CH will only consider a change of its state if it receives a message from another CH. A CH receiving a hello message from another CH will remain in the same state if it has more MNs on its cluster than the sender. This simple criterion favours larger clusters and does not take into account the mobility of the cluster members, how cohesive the smaller cluster is, or if the clusters are moving in opposite directions. Also, with large neighbourhoods, the clusters will have the tendency to grow uncontrollably, thus potentially overloading their clusterheads.

Peng *et al.* [12] use a different clustering approach. They design a utility-based methodology for cluster formation. The utility function uses as parameters the *closest position to average* and the *closest velocity to average*. Periodically, each node broadcast its status to the neighbours. After receiving this information, each vehicle chooses its CH based on the results produced by the utility function: the node with the highest value is chosen as the CH. That study attempts to enhance the classical clustering algorithms by making them aware of some of the idiosyncrasies of vehicular movement; however, it still applies many fixed weights and parameters that fail to adapt to the traffic dynamics. For example, the cluster formation interval is fixed, which implies a synchronous formation of clusters, which in turn may not be a good approach for cluster re-organisation. Not all nodes need to re-evaluate their conditions at the same intervals.

The study of a cluster-based medium access scheme for VANETs conducted by Gunter *et al.* [10] had as its objective to develop a medium access scheme for vehicular ad hoc networks based on vehicle clustering. It shows that the grouping of vehicles has an advantage not only for routing, but also for other layers. It proposes a Cluster-Based Medium Access protocol (CBMAC) to provide a fairer medium access. After the cluster formation, the CH takes over the responsibility for allocating bandwidth to its members. The algorithm used for cluster formation is based on CBLR [11]. It gives special attention to situations where CH has to change its state. The first circumstance is when a CH has no members on its cluster; then it changes its state to *undecided*. The second circumstance is when two CHs appear in the transmission range of one another. One CH has to change its state and become a member of the other cluster. The decision is based on a weighted factor that considers connectivity, mobility and mean distance to the neighbours.

The connectivity factor is defined as the difference between the actual number of neighbours and an optimum number of neighbours. The mobility factor is the mean relative velocity with respect to the neighbour nodes. The distance to the neighbours means the mean value of all distances to the neighbours. The combination of these measures produces a weight which is used to decide which CH should survive the confrontation. The value of that weight is included in all hello messages.

The proposed algorithm was studied in three scenarios with different traffic densities in a middle-sized city environment. The main criterion for performance evaluation is the lifetime of the CH. The experiments demonstrated that the probability that a node will be a CH for a short period of time is higher than when the period is long. An increase in traffic density was observed to amplify this effect. It is clear that with the increase in the traffic density, more CHs will be in range of other CHs, and many of them will be forced to change their state. However, there is no compelling reason to force such a CH to change its state immediately: neighbouring clusterheads may operate together for a certain amount of time.

III. The Proposed Approach

The CBLR algorithm [11] exemplifies a simple approach to cluster formation and maintenance. Its major drawbacks are its bias toward larger clusters and the tendency of having a large number of nodes in the CH state, e.g., a node having no neighbours will end up as the clusterhead of a trivial cluster. The clustering algorithm proposed by Gunter *et al.* [10] attempts to remedy this problem by using a measure of mobility to stabilize the CH state; however, the clusterheads may still change their status rapidly (and unnecessarily) if they get into one another's range.

The general concept of *aggregate mobility* has been studied as a way to improve the stability of clusters. Basu *et al.* [5] proposed a relative mobility metric for MANET and applied it in their MOBIC algorithm. With their scheme, the received signal strength (RSS) at the receiving node is used as an indication of the distance between the sender and the receiver. The ratio between two successive takes of that measure for the periodic *hello* messages provides an indicator of the relative mobility between the two nodes. As an additional technique of stabilizing clusters, they suggest to dampen the trigger, i.e., to delay re-clustering for a certain predefined amount of time when two clusterheads move into one another's range. This way accidental contacts between (otherwise stable) clusterheads will not cause unnecessary and intermittent re-organizations.

Our proposal follows the same general idea, but employs a subtler set of rules. First, we require that the clusterheads exchange more than one packet within a certain amount of time (dubbed the contention time) in order to begin considering a re-organization. Second, the node's decision regarding its status change is based on its perception of the aggregate local mobility (ALM). The clusterhead with the lower ALM maintains its state, while the other changes it. Another difference is to prevent a regular member (an MN node) from immediately changing its status to CH when it stops receiving beacons from its last clusterhead and there is no other clusterhead in the neighbourhood. When something like that happens, the MN node will first go to the UN state. This change postpones the creation of a new clusterhead, which could trigger an unnecessary re-clustering process, and also gives more time to the MN node to detect another clusterhead that it can subscribe to.

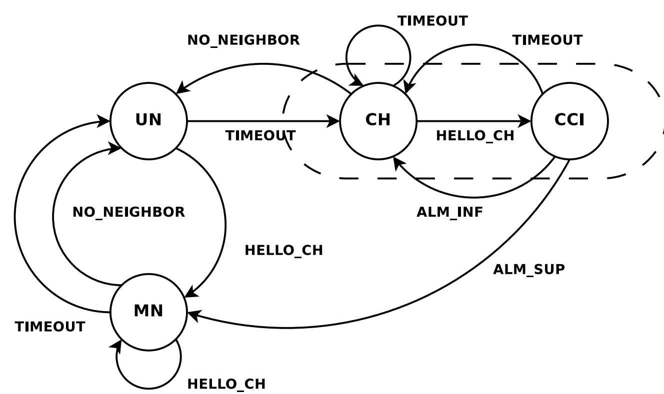


Fig. 2. The proposed protocol

To implement these changes, we propose the protocol whose finite state machine is shown in Fig. 2. A node starts in the UN state where it listens to the wireless channel and periodically sends hello packets to announce its presence in the neighborhood. It also sets up a timer to wait for clusterheads to materialise in its range. Upon reception of a hello message from a clusterhead (HELLO_CH), the node immediately changes its state to MN. While in MN, the node only changes its state in two circumstances: 1) no CH hello message has been received within the timeout interval (meaning that there is no clusterhead in the neighborhood), 2) there are no neighbours, which condition is detected via another timeout – while expecting *any* hello message.

The case of a node in CH state is more complex. If a clusterhead does not receive packets from any other clusterhead, it continues in the present state. If the node detects that is has no neighbours (general hello timeout), then it will return to the UN state. The important part happens when two clusterheads get into one another's range and exchange packets. After receiving a packet from the other CH, the first clusterhead enters the contention mode (CCI). This means that it will wait for a certain amount of time (the contention time) to see if it receives another packet from the same CH. If that does not happen before the contention timer expires, the first message is simply ignored and the node continues as a clusterhead. Otherwise (there is a second HELLO_CH), the node has to decide whether to continue as a clusterhead or to change its state. The decision is based on the ALM weight (announced in the hello messages). The node with the lower ALM will remain in the CH state (ALM_INF), while the other will change to MN (ALM_SUP).

The ALM weight is calculated in a way similar to [5], but slightly different. Instead of using the RSS, which is highly unreliable, we assume that nodes know their geographical locations (within some easily tolerable error), e.g., from GPS. One can reasonably expect that a contemporary (or in fact slightly futuristic) vehicle equipped with a sophisticated wireless communication system can easily afford a GPS receiver. The ratio between two successive takes of the distance between a node and its neighbour is used to define the relative mobility between the two. Specifically, the relative mobility of node Y with respect to node X is:

$$M_Y^{rel}(X) = log\frac{Dist_{previous}}{Dist_{current}} \quad (1)$$

The measure of a node's ALM is the variance of its relative mobility over all neighbours X_j, i.e.,

$$M_Y = var(M_Y^{rel}(X_j)) \quad (2)$$

IV. Tools & Methodology

Our proposed clustering algorithm was evaluated through detailed simulation. Vehicle mobility was simulated with SUMO [13], which uses a traffic model developed by Stefan Krauss *et al.* [14]. A SUMO model is characterised by parameters describing vehicle acceleration, deceleration, size and maximum speed. Topology information, like maximum lane speed and the number of lanes are also required. In our experiments, we used the average vehicle length of $5m$, the acceleration rate of $0.8m/s^2$, the deceleration rate of $4.5m/s^2$, and the maximum speed of $36m/s$.

A box topology of roads was assumed with each edge representing a street or avenue segment of $1km$ length. Each edge was composed of two lanes going in opposite directions. The set of maximum speed lanes (LMS) was specified as follows: 10, 15, 20, 25, and $30m/s$. The vehicles would enter the scenario at input rates (IR) of 240, 48, 24, 16, and 9.6 cars/min. Depending on the combination of the maximum lane speed and the input rate, the average number of vehicles in a simulation experiment ranged from 45 cars/min to 402 cars/min, as described in Tab. I.

Our proposed clustering algorithm was compared with GDMAC, the Generalised Mobility Adaptive Clustering algorithm, developed by Basagni [6]. That well known algorithm aims to reduce the impact of mobility on the structure of clusters. It employs a parameter, *H*, to control the weight difference necessary for a re-clustering trigger to be effective, and a parameter *K*, to control the spatial density of the clusterheads. In our experiments, those parameters were kept within the ranges suggested in [6].

TABLE I
AVERAGE VEHICLE DENSITY

LMS	IR				
	240	**48**	**24**	**16**	**9.6**
10	391	402	253	168	100
15	307	299	176	117	70
20	270	264	140	92	55
25	239	242	128	80	47
30	238	241	123	78	45

The node weight values were randomised at the beginning of the simulation. The layer 2 functionality, required by Basagni's algorithm to acknowledge new links and link failures, was implemented using periodic beacons and periodic clean-ups. The periods used were $0.5s$ and $3s$ respectively.

With our proposed scheme, a node also issues periodic hello messages when in the UN or CH state. The period between two beacons was set to the same value as for Basagni's algorithm, as was the inter-cleanup interval. The hello messages included the information necessary to calculate ALM: node ID, node status, node ALM and position.

Both clustering algorithms were simulated in SIDE/SMURPH [15], using its implementation of the IEEE 802.11 MAC with DCF. The transmission power was set to 18 dBm, the transmission rate to 10 Mbs, and the background noise to -110.0 dBm.

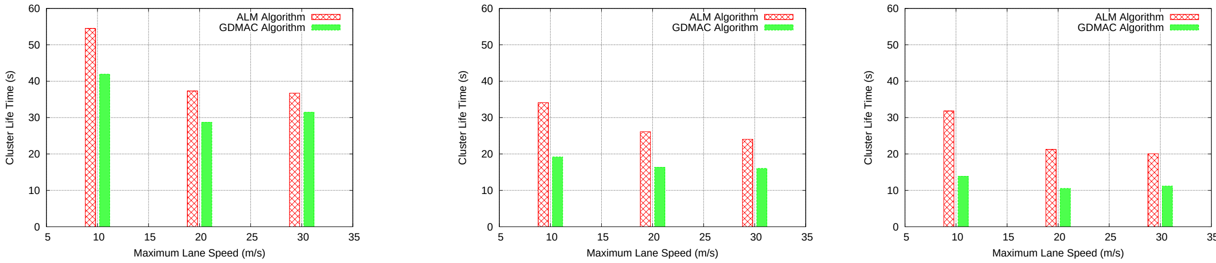
V. Results

As the environment of a vehicular network is particularly difficult from the viewpoint of stability of communication paths, our primary goal was to study the degree of persistence of clusters, which provide the semi-infrastructure of the network. The following metrics were used in our study:

- **Normalised Cluster Lifetime**: the average percentage of time that a node acted as a clusterhead during its total lifetime, normalised to the average lifetime of all nodes that were clusterheads at some moment (the same metric was used in [6]);
- **Individual Cluster Lifetime**: the average duration of a continuous period during which a node acted as a clusterhead;
- **Percentage of Clusterheads**: the ratio of the number of clusterheads to the total number of nodes;
- **Status Change**: the average number of status (role) changes per node during its lifetime.

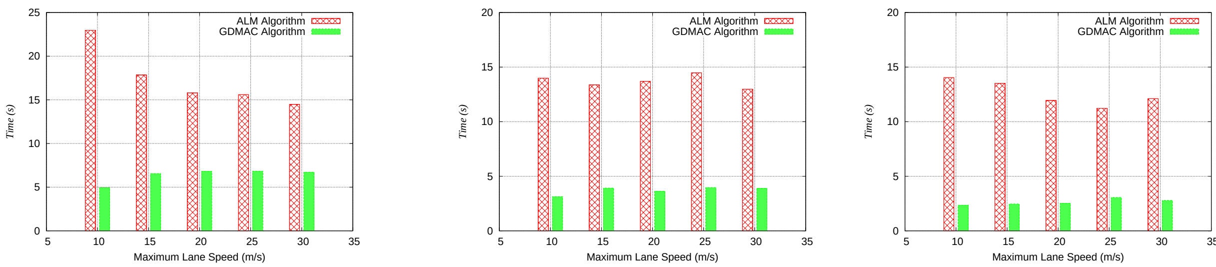
A. Cluster Lifetime

The first measure of cluster lifetime (the normalised cluster lifetime) shows the proportion of time a node acted as a clusterhead relative to the lifetime of all nodes that ever acted as clusterheads. A higher value of this measure can be interpreted as an indication of improved stablility, because it means that the global role of clusterheads has been spread over



(a) 9.6 cars/min (b) 24 cars/min (c) 240 cars/min

Fig. 3. Normalized Cluster Lifetime



(a) 9.6 cars/min (b) 24 cars/min (c) 240 cars/min

Fig. 4. Individual Cluster Lifetime

a fewer number of nodes. As shown in Fig. 3, our algorithm outperforms GDMAC in all circumstances.

As the normalised cluster lifetime aggregates all the periods that a node has acted as a clusterhead, it fails to reflect the frequency of switchovers. For example, it is possible that the same, relatively narrow, population of nodes have shared all the responsibility for delegating clusterheads; however, within that population, the roles were switched many times thus causing excessive disturbance in communication. The individual cluster lifetime, presented in Fig. 4, shows the average length of a continuous period when a node was acting as a clusterhead, thus eliminating this possible flaw.

This time the superiority of our scheme is considerably more pronounced than in the previous case. This large difference can be explained as follows. First, as the traffic density increases, the probability of a node to remain as a clusterhead for a short period of time is higher, as pointed out by Gunter *et al.* [10]. The second reason is due to a characteristic of the GDMAC algorithm. Consider s scenario where a node is the clusterhead of a cluster with several members. Suppose that for some reason, this clusterhead stops beaconing its presence. After some time, the cluster members declare the CH node dead and execute the procedure *Link_Failure* [6]. Every cluster member on this cluster executes this procedure. If some cluster members have no other CH in their neighbourhood and they are neighbours among themselves, they will all become clusterheads almost at the same time. There will be a period (around one beacon time) where they have to re-organise making up their mind as to who will be the new clusterhead. Some of them, the ones that have been forced into the temporary CH role by the *Link_Failure* procedure, will soon change their status to MN, contributing very short intervals to the averaged individual cluster lifetime.

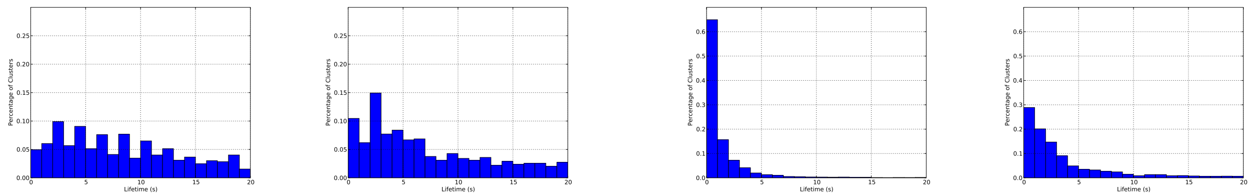
This scenario is fairly common in vehicular mobility scenarios and its high frequency drastically decreases the average CH time for GDMAC. This large contribution of short lifetime span is clearly visible in Fig. 5. For GDMAC, under heavy traffic, most (60%) of clusterhead lifetime is lower than 1 second.

B. Percentage of Clusterheads

Our proposed algorithm exhibits a slightly higher clusterhead density than GDMAC for light traffic conditions with the difference increasing for heavier traffic. Fig. 6 shows that the percentage of CH increases on faster lanes, to a point where 20% of the nodes are clusterheads.

C. Status Changes

The measure of status changes (Fig. 7) indicates how stable a node is in its role. If there is a large number of status changes, a node will have difficulty to fulfill its duties for higher protocol layers, which will negatively impact the whole communication system. The metric is calculated taking into account also the nodes in the MN state. The results show a reduction in the frequency of status changes when the maximum lane speed ranges from 10 to 20 m/s, and a slight increase after that. It is clear that GDMAC creates more instability by changing the status of its nodes considerably more often. The same happens under light and heavy traffic.



(a) ALM Algorithm (b) GDMAC Algorithm

Fig. 5. Individual Cluster Lifetime for *max lane speed* 10 m/s and *input rate* 240 cars/min on left and *max lane speed* 30 m/s and *input rate* 9.6 cars/min on right

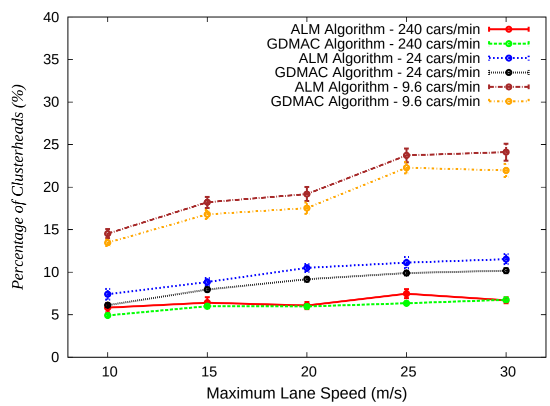


Fig. 6. Clusterhead Density

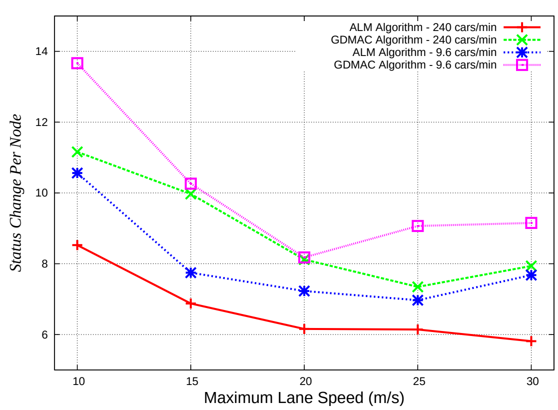


Fig. 7. Status Changes per Node

VI. Conclusion

VANETs exhibit idiosyncratic characteristics demanding networking solutions especially geared to their environment. By self-organising themselves into clusters, vehicular nodes create a hierarchy within the network, which helps them optimise resources and reduce communication burden. However, the highly dynamic topology of a vehicular network results in propensity for frequent cluster formation and re-organisation, which decreases cluster stability.

We have proposed a beacon-based clustering algorithm achieving a significantly higher cluster stability than previous schemes. The essence of our approach is to use aggregate local mobility to decide which clusterheads should retain their roles after encountering other clusterheads. Further improvement is achieved by delaying the re-clustering action in order to prevent over-eager re-clustering upon an accidental contact, e.g., occurring when two clusters pass one another while moving in opposite directions.

References

[1] L. Wischhof, A. Ebner, and H. Rohling, "Information dissemination in self-organizing intervehicle networks," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 6, no. 1, pp. 90–101, March 2005.

[2] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *Vehicular Technology Magazine, IEEE*, vol. 2, no. 2, pp. 12–22, June 2007.

[3] J. Y. Yu and P. H. J. Chong, "A survey of clustering schemes for mobile ad hoc networks," *Communications Surveys & Tutorials, IEEE*, vol. 7, no. 1, pp. 32–48, Qtr. 2005.

[4] H. Moustafa, *Vehicular networks : techniques, standards, and applications*, H. Moustafa and Y. Zhang, Eds. CRC Press, 2009.

[5] P. Basu, N. Khan, and T. Little, "A mobility based metric for clustering in mobile ad hoc networks," *Distributed Computing Systems Workshop, 2001 International Conference on*, pp. 413–418, Apr 2001.

[6] R. Ghosh and S. Basagni, "Mitigating the impact of node mobility on ad hoc clustering," *Wireless Communications and Mobile Computing*, vol. 8, no. 3, pp. 295 – 308, 2008.

[7] K. Cai, "Design and analysis of a connected dominating set algorithm for mobile ad hoc networks," Master's thesis, The University of British Columbia, April 2004.

[8] J. Sucec and I. Marsic, "Hierarchical routing overhead in mobile ad hoc networks," *Mobile Computing, IEEE Transactions on*, vol. 3, no. 1, pp. 46–56, Jan-Feb 2004.

[9] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.

[10] Y. Gunter, B. Wiegel, and H. P. Grossmann, "Cluster-based medium access scheme for vanets," *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pp. 343–348, 30 2007-Oct. 3 2007.

[11] R. A. Santos, R. M. Edwards, and N. L. Seed, "Inter vehicular data exchange between fast moving road traffic using an ad-hoc cluster-based location routing algorithm and 802.11b direct sequence spread spectrum radio," in *PostGraduate Networking Conference*, May 2003.

[12] P. Fan, J. G. Haran, J. Dillenburg, and P. C. Nelson, "Cluster-based framework in vehicular ad-hoc networks," *Lecture Notes in Computer Science*, vol. 3738, pp. 32–42, October 2005.

[13] D. Krajzewicz, G. Hertkorn, P. Wagner, and C. Rossel, "Sumo (simulation of urban mobility): An open-source traffic simulation," in *Proc. Middle East Symposium on Simulation and Modeling*, September 2002.

[14] S. Krauss, P. Wagner, and C. Gawron, "Metastable states in a microscopic model of traffic flow," *Physical Review*, vol. 55, no. 5, pp. 5597–5602, May 1997.

[15] P. Gburzynski and I. Ioannis Nikolaidis, "Wireless network simulation extensions in side/smurph," in *Proceedings of the 2006 Winter Simulation Conference*, 2006.