CBL: A Clustering Scheme for VANETs

Lucas Rivoirard
Univ Lille Nord de France,
IFSTTAR, COSYS, LEOST
F-59650 Villeneuve d'Ascq, France
lucas.rivoirard@ifsttar.fr

Martine Wahl
Univ Lille Nord de France,
IFSTTAR, COSYS, LEOST
F-59650 Villeneuve d'Ascq, France
martine.wahl@ifsttar.fr

Patrick Sondi
Univ. Littoral Côte d'Opale,
LISIC - EA 4491
F-62228 Calais, France
patrick.sondi@univ-littoral.fr

Marion Berbineau
Univ Lille Nord de France,
IFSTTAR, COSYS
F-59650 Villeneuve d'Ascq, France

Dominique Gruyer
IFSTTAR, COSYS, LIVIC
F-78000 Versailles, France

***Abstract*—Routing protocols for vehicular ad hoc networks resort to clustering to optimize network performance. In existing proposals, cluster-heads are chosen based on various metrics such as the number of its direct neighbors, the quality of the links, *etc*. Other clustering techniques consider the geographic environment of the roads, and they choose one cluster-head for each space subdivision. The clustering scheme proposed in this work combines the information on road configuration, vehicle mobility and link quality in order to build a structure similar to vehicular network infrastructure, while relying only on the vehicles. The evaluations show that this scheme allows creating and maintaining during a significant time a small number of stable connected groups, in most cases, just one in each traffic direction. This clustering scheme can be integrated into any reactive, proactive, or geographic ad hoc routing protocol in order to optimize the flooding and simplify route maintenance. And it allows the routing protocol to operate without any global location information service.**

Keywords–Clustering; Routing protocols; Cooperative vehicles; V2V; VANET; Performance evaluation.

I. INTRODUCTION

With the wide deployment of 3G/4G infrastructures, new mobile services are proposed to drivers through smartphones or built-in car devices. The concept of the connected vehicle and that of the autonomous vehicle are now effective at least in real-world testing. The development of both the IEEE 802.11p and the upcoming 5G includes machine-to-machine direct communications, which also refers to vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) communications. In this context, enhancing vehicular ad hoc network (VANET) routing protocols is mandatory to ensure that this paradigm will play its role in future intelligent transportation systems. The routing protocol can lead to either a flat topology, without hierarchy [1], or a clustered topology, with a node hierarchy. Clustering is one important technique that can be used by ad hoc routing protocols to optimize network management. A cluster is a virtual division of the network into groups. Resorting to clusters optimizes the range of packet flooding by limiting the packet transmission to one or more clusters.

A cluster includes different types of nodes. The group leader is called "cluster-head". Its "ordinary member" nodes are connected to the cluster-head. "Gateway nodes" are members of several clusters, thus making a link between them. The clustering schemes can generate separated clusters (without gateway nodes) or not. Clustering methods are active, passive, or hybrid. In active clustering, dedicated control messages are sent for cluster management. In passive clustering, clusters are created on demand when data need to be transmitted. In hybrid clustering, information needed for cluster management is added to the packets. The clustering size is also characterized by the number of hops. For instance, in one-hop clusters, each member node is directly connected to its cluster-head.

Over two decades, many clustering schemes have been proposed in order to enhance the performance of ad hoc routing protocols according to various link or node metrics [2]–[4]. Especially in the case of VANET, the road traffic environment and the velocity due to vehicle mobility are important factors in the design of a clustering scheme. In order to evolve from the plethora of existing approaches towards standardized solutions, the European Telecommunications Standards Institute (ETSI) has recently published the Geonetworking requirements [5] that fix the design guidelines of VANET architecture.

This work presents an approach inspired by infrastructure-based vehicular networks. The proposed hybrid clustering scheme leads to an emerging structure, a virtual backbone in the VANET, similar to that obtained with Road Side Units (RSUs) deployed along the roads equipped for vehicular communications. In this way, any routing protocol that uses this clustering scheme can operate in both infrastructure-based and infrastructure-less VANET. When operating in infrastructure-less mode, the structure built by the proposed clustering scheme aims to offer a stable link service between the nodes for the applications, despites the high mobility of the VANET nodes. In addition, the CBL clustering scheme uses only the position and velocity information of the closest node neighborhood. Therefore, unlike most the geographic-based routing protocols, no global knowledge of the locations of the nodes is needed, and any global location service that would require an infrastructure is not necessary. The structure built by CBL allows the routing protocol supplying unicast and broadcast message exchange.

The paper is organized as follows. Section II presents a related work on clustering schemes for ad hoc networks, especially VANETs. Section III depicts the clustering scheme designed for VANET in this work. Section IV deals with its performance evaluation. We finally conclude.

II. Related work

Many clustering methods for mobile applications were first studied for Mobile Ad hoc NETworks (MANETs) [6]–[10]. The Lowest ID and the Highest Degree methods [6] offer good performance in MANET, but they do not consider mobility. An attempt to introduce mobility in the cluster-head election by considering the distances between the nodes is achieved through the Mobility Based Clustering method [8]. However, in highly dynamic ad hoc networks, it is necessary to combine several criteria, which leads to the proposal of the Weighted Clustering Algorithm [9] based on a weighted sum of several criteria values. Current studies introduce the notion of communication interest among devices as an extra parameter in the Weighted Clustering Algorithm, such as in [11] using the communication interest among MANET nodes, the physical proximity, and the energy availability to create coalitions. MANET nodes are distributed randomly in the space without favoring a geographical direction. But, VANET nodes are subject to mobility constraints such as road infrastructure with specific rules, strong variations in relative speeds between the vehicles, *etc*. Therefore, the studies that adapt clustering schemes to VANETs pay attention to the development of relatively time-stable clusters. Two main approaches can be found. The first one does not introduce logical relations between the clusters, and is based on a cost function [12]–[18], or a fuzzy logic function [19]. The second approach creates logical connections between the clusters through an optimal set of relay nodes [20]–[23], defined as a backbone, in order to improve the traditional forwarding scheme. However, most of these studies focus on the way the cluster-head is chosen at network level, without considering the special features of road applications in a global system approach.

With this in mind, the Dynamic Backbone Assisted protocol [20] uses either the distance or the communication rate as the metric to elect the cluster-head; [22] the distance; the BackBone Routing protocol [21] the speed, the traffic direction, and the quality of transmission; the Connected Dominating Set-Stable Virtual Backbone [23] the speed, the distance, and the direction. [23] searches for cluster-heads having low speeds to stabilize the chain structure. [20]–[22] consider the transmission range R as a relevance zone where there is a cluster-head [20], or a cluster-head between two associated cluster nodes located at borders of R to enable communication with two other clusters [21], or four cluster-heads distributed every $\frac{1}{4}R$ [22]. [23] puts no constraint in terms of number of cluster-heads. Furthermore, [21] introduces the notion of upstream and downstream message transmissions.

The scheme proposed in this paper will lead to the construction of a single backbone in each traffic direction, formed with cluster-heads that are dynamically chosen by the other nodes in order to form and stabilize a structure over time. As in [23], no condition will be put on the number of cluster-heads. Unlike [20]–[23], no thresholds other than temporal thresholds will be applied. Unlike [11], to obtain application-independent backbones and to preserve the collaboration between nodes, the idea of communication interest among vehicles or devices of vehicles is not considered. Also, the notions of upstream and downstream relays [18] will be exploited.

III. CBL clustering algorithm

The Chain-Branch-Leaf (CBL) clustering scheme designed in this work builds a backbone that will allow both the communications in close neighborhood, and remote communications between distant vehicles according to the specific needs of each application. Communication between close vehicles will be necessary for on-board functions (sensor functions, geo-localization, extended perception, *etc.*) that will have to share variables periodically (speed, acceleration, positioning information, *etc.*) for their inner process. Such variables will be useful to coordinate the relative movements of vehicles in future autonomous systems. Communication over long distance will be necessary for distributed applications that need to forward messages, upstream or downstream the traffic flow, to remote vehicles. As an example, forwarding messages upstream can help prevent a risk of bottleneck, and transmitting messages downstream can allow vehicles to inform about the approach of a priority vehicle (police car, fire truck).

A. Assumptions

We assume that each vehicle is equipped with a GPS that enables self-localization, also that it can determine its speed. It also has a wireless ad hoc communication card (802.11p for an example) enabling communication with the other vehicles up to a certain range in line-of-sight.

B. Definitions

CBL is a hybrid distributed algorithm: each communication node initiates its own process. It creates a hierarchy between the nodes in order to build 1-hop clusters so that each node of a cluster can directly communicate to the cluster-head without going through another intermediary node. CBL can be implemented inside any ad hoc routing protocols. It uses Hello messages such as those supplied by OLSR to build its hierarchical structure. However, other ad hoc routing protocol can be used in adding beacon messages to it if such periodic messages are not already provided.

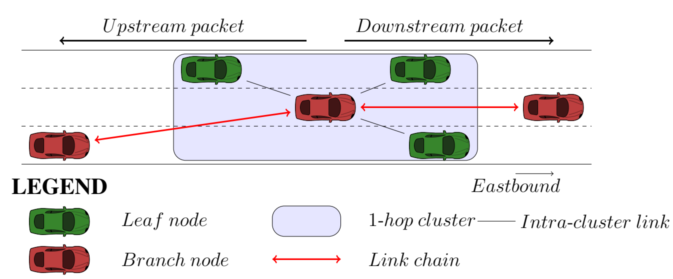


Figure 1. Example of a situation on a three-lane one-way highway

Some definitions are specified as follows:

- **A branch node** (Fig. 1) is a cluster-head node that is elected by other nodes (branch or leaf). It emits HELLO messages like every node, but it is the only one allowed to emit topology control messages (TC), to forward application messages, and to participate in the construction of a chain. In order to control the propagation of a message, based on the application request specified in the header fields, a branch node can forward it to:

 ISBN: 978-1-61208-573-9

- its leaf nodes;
- upstream branch node;
- downstream branch node;
- all branch nodes (including branch nodes of another traffic direction).

Our CBL implementation assumes that these destination options are coded into the link code of the original format of the packets defined in OLSR protocol [24].

- **A leaf node** is an ordinary node which tries to connect itself to the closest branch node. If no branch node is detected, the leaf node elects the neighbor moving with the lowest speed and in the same traffic direction, as a branch. A leaf node sends both HELLO and application messages of which it is the originator.
- **A chain** is a virtual backbone made up of a sequence of branch nodes. Ideally, one chain should be created per traffic direction. On longitudinal road context such as highways, the chains behave as a virtual backbone similar to the one that should be obtained with an infrastructure. It offers to its branch nodes a path to forward application messages over long distance.
- **BranchChoice** is a field added in the HELLO message and containing the address of the elected branch to which the HELLO originator node is connected.
- **The Connection Time (CT)** is the time during which two nodes N_i and N_j could communicate if they kept the same speed. This metric, also called "contact time", has been used in [25]–[27]. CT is approximated using (1). This equation takes into account the positions of the nodes ($[X_i, Y_i]$ for the node N_i and $[X_j, Y_j]$ for the node N_j) their speeds (V_i for the node N_i and V_j for the node N_j), and the maximum radio range (R_{max}):

$$CT = \frac{-(ab+cd) + \sqrt{(a^2+c^2) * R_{max}^2 - (ab-bc)^2}}{a^2+c^2} \quad (1)$$

$$\left\{ \begin{array}{lcl|lcr} a & = & V_i cos(\sigma_i) - V_j cos(\sigma_j) & b & = & X_i - X_j \\ c & = & V_i sin(\sigma_i) - V_j sin(\sigma_j) & d & = & Y_i - Y_j \end{array} \right.$$

C. HELLO message

Ad hoc network initialization is triggered by the emission of HELLO messages (the beacons) at node level. The Time-To-Live (TTL) value of the HELLO messages is set at 1 to contain its broadcast in the 1-hop neighborhood. At CBL network initialization, there is no chain, no branch, and every node are leaf nodes.

The following information concerning the sender node is added to standard HELLO messages:

- its current location X,Y in Cartesian coordinates (coming from the longitude and latitude in GPS data);
- its current speed V and its steering angle σ, which provides the node direction (extracted from the on-board computer or GPS data for example);
- its current type: leaf or branch node;
- the elected branch node (BranchChoice) to which it is connected;
- the elected upstream branch node to which it is connected (if type=branch) or empty (if type=leaf);
- the elected downstream branch node to which it is connected (if type=branch) or empty (if type=leaf);
- the current list of its 1-hop neighbors with link types;
- the validity time timestamp (Vtime) that indicates the validity period of the information contained in a HELLO message.

D. Local 1-hop neighbor and node internal variables

When processing HELLO messages, each node creates and maintains a local table containing the list of its 1-hop neighbors. We introduce a variable called "Elected" to distinguish in the table whether the neighbor has elected the node either as a branch (the neighbor is a leaf) or as a branch in its chain (the neighbor is a branch). In this case, the variable "Elected" is set at true. Otherwise, it is set at false. This table contains the neighbor addresses and for each neighbor: the link type (UNSPEC if unspecified, LOST if the link is lost, ASYM if the communication link is unidirectional, and SYM if the communication link is bidirectional), the timestamp T1 of the last HELLO messages received from the neighbor, the timestamp T2 when the link became symmetric, its position (X,Y), its speed (V), its direction (σ), and its type (branch or leaf). The table also includes for each neighbor the value of the "Elected" variable.

Each node records and maintains internal variables. These latter are the addresses of the elected upstream and downstream branch nodes, the addresses of BranchChoice, and the timestamp T3. T3 contains the date of a the last HELLO message received from any leaf or branch node that elected it as a relay branch node.

E. CBL scheme

Each reception of a HELLO message by a node N_i and coming from a neighbor node N_j triggers the following procedure (Fig. 2).

1) Algorithm 1. Update the 1-hop neighbor table: CBL scheme uses the same algorithm than the OLSR protocol to update the 1-hop neighbor table and link type. Moreover, it checks for each neighbor in the table if the duration elapsed since the last HELLO message received from it (recorded in

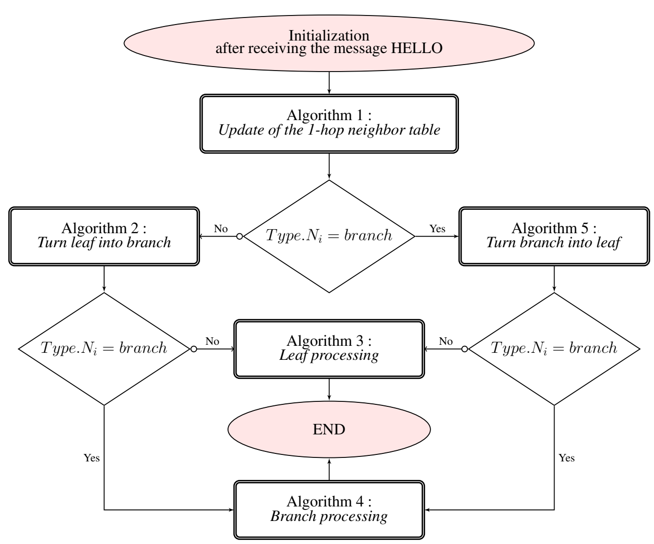


Figure 2. Procedure applied when receiving a HELLO message

timer T1) is higher than Vtime. In that case, the neighbor is removed from the table.

2) Algorithm 2. Turn leaf into branch: when a neighbor N_j has chosen N_i as a branch (the BranchChoice of N_j is set at the address of N_i), the node N_i turns its type into branch. Then, N_i updates its timer T3, initializes its BranchChoice to empty, and writes in its neighbor table that it is elected by N_j. If N_i was already a branch, it will just add automatically the node N_j as its elected upstream or downstream branch nodes to form a chain taking into account its relative position (up- or downstream).

3) Algorithm 3. Leaf processing: a leaf N_i selects a node as its branch, then it writes the branch address in its BranchChoice variable. When several branch candidates are detected, some optimizations are introduced in the choice process. In this work, N_i first looks for branch nodes driving in the same direction, then it chooses the closer according to the distance. Notice that two nodes drive in the same direction when:

$$|\sigma_{Ni} - \sigma_{Nj}| < \sigma_{max} \quad (2)$$

When N_i detects no branch around after a time greater than Vtime with respect to the timer T2, the node sets in its BranchChoice the address of the leaf that is driving in the same direction. Nevertheless, if more than one leaf is a candidate, then the address of the candidate node having the lowest speed is put in its BranchChoice. The chosen leaf will become a branch after receiving a HELLO message from N_i. Selecting branch nodes with low speeds ensures the stability of the chain because, according to [23], the lower the relative speed between the branch nodes, the better the radio communication.

When N_i has N_j address in its BranchChoice variable, if N_j is still a branch node, N_i updates its time counter T3 with the last reception timestamp of the HELLO message from N_j. However, if N_i received no HELLO message from N_j for a time longer than Vtime, it initializes its BranchChoice to empty in order to join a new cluster.

4) Algorithm 4. Branch processing: a branch node N_i participates in the creation of a chain. To this purpose, it elects an upstream branch and a downstream one from its current position, taking into account its trajectory direction. The election process selects the branch nodes that, firstly, currently exists and that, secondly, have not yet joined a chain or have chosen N_i as a branch node of their chain.

By keeping the same upstream and downstream branch nodes for N_i, while N_i location is maintained between them (no overtaking) and N_i is still in their transmission range (it receives their HELLO message), the algorithm favors the stability of the chain.

When N_i detects no branch node either upstream or downstream, it selects among its 1-hop neighbors the leaf node driving in the same direction, provided that this latter brings at least one more link, via its own 1-hop neighborhood, to a new node previously unknown from N_i. Next, if several leaf nodes are found, N_i selects, as BranchChoice, the address of the one that has the highest CT value (see equation 1). This election will change this leaf node into a branch node after the reception of the next HELLO message from node N_i. To select a new branch among leaf nodes, we do not consider a fixed distance threshold to avoid restricting the scheme to only few highway contexts. The advantage of using a communication network metric is to decrease the probability of choosing the closest node, and therefore the risk of frequently breaking the chain when vehicles overtake.

N_i updates the “Elected” variable for N_j in its 1-hop neighbor table: if N_j is connected to N_i, the variable is set at true and N_i updates its timer T3, otherwise the variable is set at false.

The branch node N_i checks for the two selected nodes (upstream and downstream) in its chain the duration elapsed since it received the last HELLO message from this latter. Every neighbor having a duration higher than Vtime is removed.

When the sender N_j is already in the selected nodes (upstream and downstream) in the chain of N_i, this latter checks that the position of N_j (upstream or downstream), its type, and its direction have not evolved, and that N_i address appears in the selected nodes (upstream and downstream) in the chain of N_j. If these conditions are fulfilled, N_i updates the counter T3. If the relative position has changed, it corrects the position of N_j in the or removes it.

5) Algorithm 5. Turn branch into leaf: a branch N_i goes back to leaf type when it received no more HELLO messages from any of the nodes that elected it for a duration above Vtime. This process refers to the value T3.

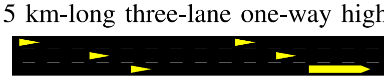
IV. PERFORMANCE EVALUATION

This section presents the performance evaluation of the proposed CBL scheme over varying highway scenarios and network traffic conditions. SUMO [28] is used in order to generate the mobility traces of the vehicles over three different road networks. The clustering scheme itself is modeled with Matlab.

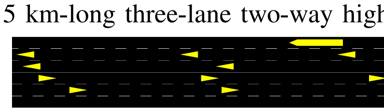
A. SUMO models

Three different road networks are modeled using the SUMO simulator:

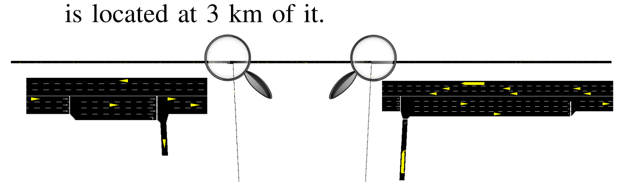
- R1: 5 km-long three-lane one-way highway;



- R2: 5 km-long three-lane two-way highway;



- R3: 5 km-long three-lane two-way highway, an entrance and a highway exit. The exit is located at 1.8 km from the beginning of the section, while the entrance is located at 3 km of it.



Three different traffic density cases are considered for this evaluation. In each case, a ratio of 1/6 trucks and 5/6 cars is considered. The different traffic densities used are listed in Table I. In the network R3, 25% of the vehicles arrive via the highway entrance, 25% of the vehicles take the exit and 50% of the vehicles just cross the whole road section.

 ISBN: 978-1-61208-573-9

TABLE I. SCENARIOS AND VALUES OF ROAD TRAFFIC DEMAND

Density	Car traffic (veh/h/direction)	Truck traffic (veh/h/direction)	R1	R2	R3
Low	500	100	S1	S4	S7
Medium	2000	400	S2	S5	S8
High	4000	800	S3	S6	S9

Where S1 to S9 are the scenarios.

TABLE II. KINEMATIC PARAMETERS FOR CARS AND TRUCKS

	Acc	Dcc	L	σ	τ	MG	MS	SF	SD
Units	m/s^2	m/s^2	m	-	s	m	km/h	-	-
Cars	2	3	5	0.5	1	2.5	150	1	0.1
Trucks	1	2	15	0.5	1	5	130	0.84	0.1

The default car following model included in SUMO simulator is a variant of the Krauß model: each vehicle drives up to its "desired speed", while maintaining a perfect safety distance with the leader vehicle (*i.e.* the front vehicle). The speed limit is set at 130 km/h, which corresponds to the legal speed limit on highways in France. We define two types of vehicles: cars and trucks. Vehicles are modeled by a set of parameters (the values are given in Table II):

- Acc: the acceleration capability of vehicles;
- Dcc: the deceleration capability of vehicles;
- L: the vehicle length;
- σ: the Krauß driver imperfection (between 0 and 1);
- τ: the driver-desired minimum time headway;
- MinGap (MG): the offset to the leading vehicle when standing in a jam;
- MaxSpeed (MS): the maximum velocity of the vehicle;
- SpeedFactor (SF): the vehicles expected multiplicator for lane speed limits;
- SpeedDev (SD): the standard deviation of the speedFactor;

To achieve realistic car following behavior [29], it is necessary to use speed distributions for the desired speed. Otherwise, if all vehicles have the same desired speed, they will not be able to catch up with their leader vehicle, thus causing unrealistic situation. Therefore, two other parameters have been introduced in order to use speed distributions in SUMO: speedFactor and speedDev. For instance, using $speedFactor = 1$ and $speedDev = 0.1$ will result in a speed distribution where 95% of the vehicles drive at a speed ranging from 80% to 120% of the legal speed limit (Fig. 3).

B. Matlab simulation

Simulation time for each of the nine scenarios is 500 s. Nodes send a HELLO message every 1 s. The thresholds Vtime are set at 3 s. The free space propagation model is used, with a transmission range of 500 m.

C. Performance metrics

Seven performance metrics are considered. The average values reported are picked up when the network is stable (between 200 s and 500 s):

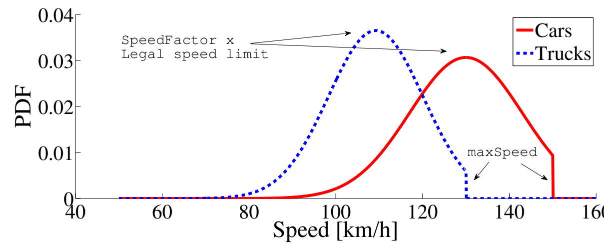


FIGURE 3. PROBABILITY DENSITY FUNCTION OF DESIRED SPEED FOR CARS AND TRUCKS.

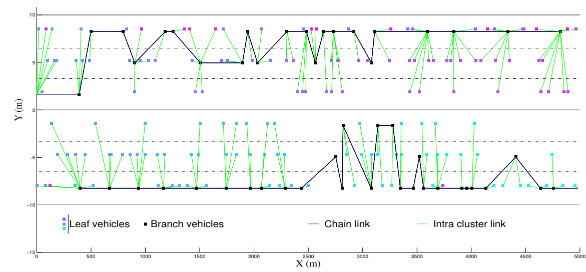


FIGURE 4. STATE OF CHAIN CONSTRUCTION IN THE CASE OF SCENARIO 5 AT TIME T=500S

1) NB_Chains: number of chains in the network
2) Branch/Chain: number of branch nodes per chain
3) 1hop/Branch: number of 1-hop neighbors (in the same traffic direction)
4) Leaf/Branch: number of leaf nodes per branch node
5) Branch_time: duration that a node remains a branch
6) Leaf_time: duration that a leaf node remains attached to the same branch node
7) Leaf/Vanet: percentage of leaf nodes in the network

D. Results

The objective of these evaluations is to analyze the structure created by CBL. Simulating the scenario S5, we observed that CBL leads to two separate chains, one in each road traffic direction (Fig. 4).

Therefore, in this paper, only the results of S2 scenario related to R1 road configuration are commented on, since this latter is the usual configuration of highway traffic in one direction (S2 represents the medium of the three studied densities). However, the results of all the scenarios are in Table III.

About 70% of the time (Fig. 5), there is only one chain as targeted for S2 scenario. Sometimes the chains are broken, mostly due to the changes in the order of the branch nodes inside the chain, but they are quickly reconstructed. The cumulative duration when there are more than one chain is about 30% of the simulation time. It is noticed that chain breaks increase with the density and the road configuration, R3 reaching the highest scores due to a lot of vehicles entering or leaving the road section (see Table III).

When the traffic becomes stable (Fig. 6), after about 150 s of simulation, there are up to 100 vehicles on the highway. We see that up to 75% of the nodes are of leaf type, and only 25% are actually branch nodes. This shows the ability of CBL to optimize the flooding of broadcast traffic since only branch nodes are allowed to relay it upstream, downstream, or both

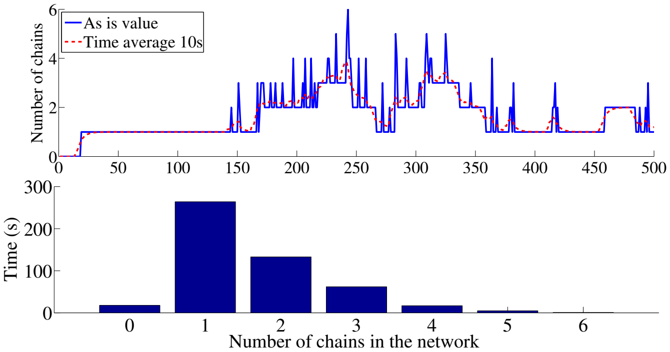


FIGURE 5. NUMBER OF CHAINS AND ACCUMULATED TIME IN THE CASE OF SCENARIO 2

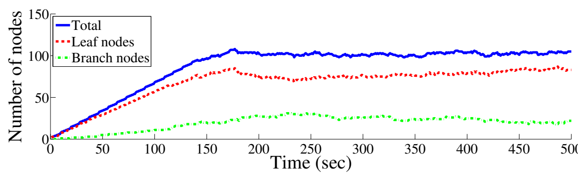


FIGURE 6. NODE TYPE PARTITIONING IN THE CASE OF SCENARIO 2

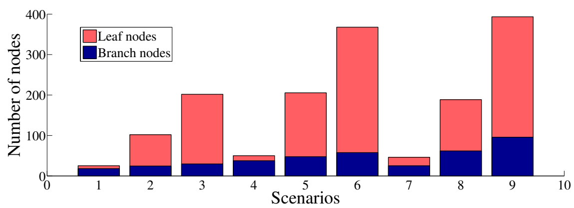


FIGURE 7. NODE TYPE PARTITIONING FOR EVERY SCENARIO (S1 TO S9)

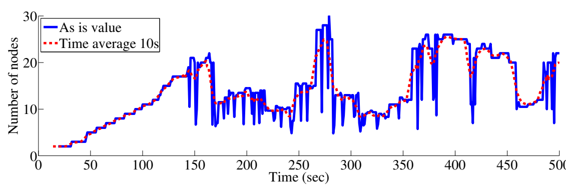


FIGURE 8. NUMBER OF BRANCH NODES PER CHAIN IN THE CASE OF SCENARIO 2

directions according to application requirements. These results are confirmed in every scenario (Fig. 7), except in S1, S4, and S7 where the traffic density is low, and therefore the clustering is less efficient (more than 50% of branch nodes). Intuitively, when the network is sparse, the vehicles are more spaced and there are more isolated nodes that become branch nodes.

Looking at Fig. 8 and 5 together shows that there are about 25 branch nodes when there is only one chain and about 10 branch nodes per chain in the presence of several chains (chain breaks). Each node has an average of 20 1-hop neighbors (Fig. 9). The results show that an average of 5 nodes (25% of the neighbors) choose the same branch node (Fig. 10). A small number of branch nodes are chosen by 75% of their 1-hop neighbors, others by only 10% mainly due to their position (at the end of the chain, *etc.*). CBL parameters may be tuned through time threshold value (Vtime) in order to improve the balance between the number of 1-hop neighbors of a node and that of the leaf nodes choosing it as their branch.

Each selected node remains a branch about 70 s (Fig. 11). Even for a vehicle moving at the lowest speed of 80 km/h, it stays a branch over 1.5 km (3 times the maximum range), which is a significant distance even on a highway. Moreover,

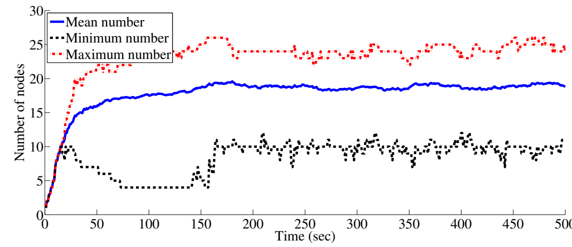


FIGURE 9. NUMBER OF 1-HOP NEIGHBORS IN THE CASE OF SCENARIO 2

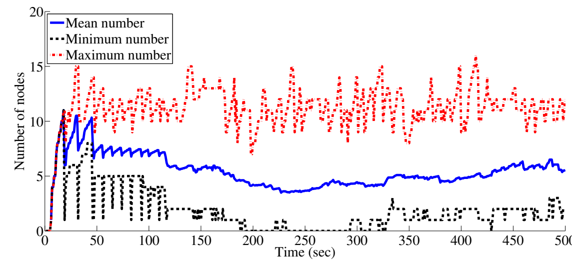


FIGURE 10. NUMBER OF LEAF NODES ATTACHED TO A BRANCH NODE IN THE CASE OF SCENARIO 2

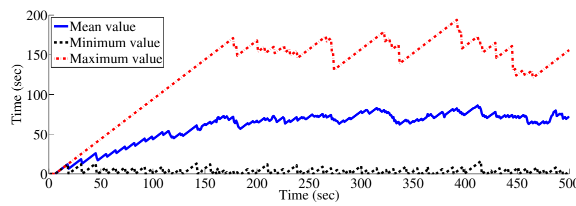


FIGURE 11. DURATION THAT A NODE REMAINS A BRANCH IN THE CASE OF SCENARIO 2

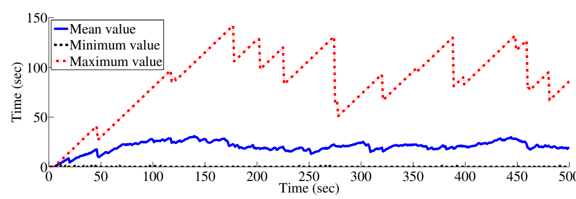


FIGURE 12. DURATION THAT A LEAF REMAINS ATTACHED TO THE SAME BRANCH NODE IN THE CASE OF SCENARIO 2

TABLE III. MEAN RESULT VALUES FOR ALL SCENARIOS

Scenario	1	2	3	4	5	6	7	8	9
NB_Chains	1.65	1.96	2.04	3.29	2.68	3.63	3.36	6.36	9.29
Branch/Chain	13.7	15.9	18.9	12.1	20.4	17.8	9.4	10.4	11.3
1hop/Branch	3.9	18.8	38.6	4.8	21.2	38.2	4.1	17.7	37.8
Leaf/Branch	1.94	4.75	7.03	1.94	4.89	6.84	2.31	3.43	4.42
Branch_time(s)	64.4	71.9	70.0	65.1	73	73.5	59.4	53.6	57.7
Leaf_time(s)	18.8	20.6	21.3	15.6	22.5	22.6	8.6	23.8	16.5
Leaf/Vanet(%)	28	76	85	25	77	84	46	67	76

See section IV-C for the definition of performance metrics.

each leaf node remains attached to the same branch for 20 s on average (Fig. 12). It is also known that most V2V safety applications have a message transmission periodicity ranging from 50 ms to 500 ms. Consequently, that 70-second time represents at least up to 40 alerts from a leaf node relayed by the same branch node to the entire network. These values are almost the same over all 9 scenarios (Table III).

V. Conclusion and Future Work

In the CBL clustering scheme, the vehicles that move at lower speed in the same traffic direction are good candidates (branches) for building a stable backbone that we call a chain. The greater the number of vehicles, the longer the chain. Each vehicle moving faster is a leaf that attaches itself to a branch node covering its current location in order to communicate with the entire VANET. The evaluations show that CBL leads to a structure that may improve VANET performance regarding several metrics. First, the branch nodes represent only 25%, thus allowing optimization of the flooding of broadcast traffic. Then, among all the neighbors of a given branch, only those having the better link quality with this latter (25% to 55%) actually choose it as a branch. The others select other branch nodes, which will result in a global structure with better link quality in the VANET. Finally, this study shows that CBL leads to significant stability since there is only one chain 70% of the time. A node elected as a branch remains a branch for 70 s, and it can serve each of its leaf nodes for 20 s. At 130 km/h, such a leaf would have moved over 720 m while being connected to the same branch node, which is longer than the communication range offered by a fixed RSU. As a clustering approach, CBL can be used in a global VANET architecture including also V2I communications, which makes it compliant to ETSI Geonetworking requirements. However, unlike other geographic routing protocols, CBL is not dependent from any global location service. Future work will consist in finding optimal values of CBL parameters for different traffic conditions, and in comparing with other clustering schemes.

Acknowledgment

The authors acknowledge the support of the CPER ELSAT2020 project which is co-financed by the European Union with the European Regional Development Fund, the French state and the Hauts de France Region Council.

References

[1] F. D. d. Cunha, A. Boukerche, L. Villas, A. C. Viana, and A. A. F. Loureiro, "Data Communication in VANETs: A Survey, Challenges and Applications," INRIA Saclay ; INRIA, report 8498, Mar. 2014, URL: https://hal.inria.fr/hal-00981126/document [retrieved: 2017-06].

[2] R. S. Bali, N. Kumar, and J. J. Rodrigues, "Clustering in vehicular ad hoc networks: Taxonomy, challenges and solutions," *Vehicular Communications*, vol. 1, no. 3, pp. 134–152, Jun. 2014.

[3] S. M. AlMheiri and H. S. AlQamzi, "MANETs and VANETs clustering algorithms: A survey," in *8th IEEE GCC Conference and Exhibition (GCCCE)*. IEEE, Feb. 2015, pp. 1–6.

[4] C. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan, "A Comparative Survey of VANET Clustering Techniques," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2016.

[5] "Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality," *European Telecommunications Standards Institute*, no. ETSI EN 302 636-4-1, May 2014.

[6] M. Gerla and J. Tzu-Chieh Tsai, "Multicluster, mobile, multimedia radio network," *Wireless Networks*, vol. 1, no. 3, pp. 255–265, Sep. 1995.

[7] C. Geng, F. Nocetti, J. Gonzalez, and I. Stojmenovic, "Connectivity based k-hop clustering in wireless networks." IEEE Comput. Soc, 2002, pp. 2450–2459.

[8] P. Basu, P. Basu, N. Khan, and T. D. C. Little, "A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks," *International workshop on wireless networks and mobile computing*, pp. 413–418, 2001.

[9] 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, 2002.

[10] S. Dhurandher and G. Singh, "Weight based adaptive clustering in wireless ad hoc networks," in *International Conference on Personal Wireless Communications*. IEEE, Jan. 2005, pp. 95–100.

[11] E. E. Tsiropoulou, S. T. Paruchuri, and J. S. Baras, "Interest, Energy and Physical-Aware Coalition Formation and Resource Allocation in Smart IoT Applications," in *Information Sciences and Systems (CISS), 2017 51st Annual Conference on*. IEEE, 2017, pp. 1–6.

[12] S. A. Mohammad and C. W. Michele, "Using traffic flow for cluster formation in vehicular ad-hoc networks," in *35th Conference on Local Computer Networks*. IEEE, Oct. 2010, pp. 631–636.

[13] M. N. Avcil and M. Soyturk, "ReSCUE: Relatively Stable Clustering for Unbiased Environments in VANETs." International Wireless Communications and Mobile Computing Conference: IEEE, Aug. 2015, pp. 1049–1055.

[14] R. Mehra, R. S. Bali, and P. Kaur, "Efficient clustering based OLSR routing protocol for VANET," in *Symposium on Colossal Data Analysis and Networking*. IEEE, Sep. 2016, pp. 1–7.

[15] F. Wei, S. Yan, C. Shanzhi, and Z. Longhao, "A mobility metrics based dynamic clustering algorithm for VANETs," in *International Conference on Communication Technology and Application*. IET, 2011, pp. 752–756.

[16] M. Azizian, S. Cherkaoui, and A. S. Hafid, "DCEV: A distributed cluster formation for VANET based on end-to-end realtive mobility," in *International Wireless Communications and Mobile Computing Conference*. IEEE, Sep. 2016, pp. 287–291.

[17] H. R. Arkian, R. E. Atani, A. Pourkhalili, and S. Kamali, "A Stable Clustering Scheme Based on Adaptive Multiple Metric in Vehicular Adhoc Networks." *Journal of Information Science & Engineering*, vol. 31, no. 2, pp. 361–386, 2015.

[18] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks." IEEE Computer Society, Jan. 2002, pp. 3866 – 3875.

[19] H. Arkian, R. Atani, and S. Kamali, "FcVcA: A fuzzy clustering-based vehicular cloud architecture," in *7th International Workshop on Communication Technologies for Vehicles*. IEEE, 2014, pp. 24–28.

[20] M. Di Felice, L. Bedogni, and L. Bononi, "Dynamic backbone for fast information delivery invehicular ad hoc networks: an evaluation study," in *the 8th ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*. ACM, 2011, pp. 1–8.

[21] C. Wu, S. Ohzahata, and T. Kato, "Can we generate efficient routes by using only beacons? Backbone routing in VANETs," in *24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*. IEEE, Sep. 2013, pp. 2929–2934.

[22] W. Xiaonan and Q. Huanyan, "Constructing a VANET based on cluster chains," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 2497–2517, 2014.

[23] M. A. Togou, A. Hafid, and P. K. Sahu, "A stable minimum velocity cds-based virtual backbone for vanet in city environment," in *39th Annual Conference on Local Computer Networks*. IEEE, 2014, pp. 510–513.

[24] T. Clausen *et al.*, "Optimized link state routing protocol (OLSR)," *IETF*, no. RFC 3626, 2003.

[25] W. Su, S.-J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *International Journal of Network Management*, vol. 11, no. 1, pp. 3–30, 2001.

[26] Y. Li, D. Jin, Z. Wang, L. Zeng, and S. Chen, "Exponential and Power Law Distribution of Contact Duration in Urban Vehicular Ad Hoc Networks," *IEEE Signal Processing Letters*, vol. 20, no. 1, pp. 110–113, Jan. 2013.

[27] I. B. Jemaa, "Multicast Communications for Cooperative Vehicular Systems," phdthesis, Mines ParisTech, 2014.

[28] S. R. Santana, J. J. Sanchez-Medina, and E. Rubio-Royo, "How to Simulate Traffic with SUMO," in *Computer Aided Systems Theory EUROCAST 2015*. Cham: Springer International Publishing, 2015, vol. 9520, pp. 773–778.

[29] L. Rivoirard, M. Wahl, P. Sondi, M. Berbineau, and D. Gruyer, "Using Real-World Car Traffic Dataset in Vehicular Ad Hoc Network Performance Evaluation," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 12, pp. 390–398, 2016.