Multipoint Relaying vs Chain-Branch-Leaf Clustering Performance in Optimized Link State Routing-based Vehicular Ad hoc Networks

Lucas Rivoirard, Martine Wahl, Patrick Sondi

Multipoint Relaying vs Chain-Branch-Leaf Clustering Performance in Optimized Link State Routing-based Vehicular Ad hoc Networks

Lucas Rivoirard[1], Martine Wahl[1], Patrick Sondi[2]

***Abstract*—Routing protocols for vehicular ad hoc networks resort to clustering in order to optimize network performance. Concerning the Optimized Link State Routing (OLSR) protocol and the plethora of its derivatives, the multipoint relaying (MPR) technique has proven its efficiency as an accurate clustering scheme over the last two decades. However, it has been emphasized recently that the MPR technique, which was originally designed for open areas, does not benefit from the particular configuration of road sections which are intrinsically spatially constrained. A clustering scheme exploiting this particularity, namely Chain-Branch-Leaf (CBL), has been introduced in order to enhance the flooding of broadcast traffic, including that related to routing operations. In this paper, both MPR and CBL are evaluated through MATLAB simulation over several scenarios based on realistic road configurations and traffic generated with SUMO simulator. The results show that CBL actually reduces the number of nodes acting as relays (cluster-heads) in the network, thus decreasing the routing traffic related to creation and retransmission of topology control (TC) messages. Also, they show that, with CBL, the nodes chosen as relays remain longer in this role, thus favoring the overall network stability, and that most of the nodes remain attached longer to the same relay than with the MPR technique.**

***Index Terms*—Clustering; Routing protocols; Cooperative vehicles; V2V; VANET; Performance evaluation.**

I. INTRODUCTION

Vehicular Ad Hoc Network (VANET) makes it possible to imagine a wide range of Intelligent Transport System (ITS) applications in terms of comfort, road traffic optimization and safety. The complexity of these systems is inherent to the VANET architectures.

A first VANET architecture is based on vehicle-to-infrastructure (V2I) communications. Road side units (RSU) are deployed along the roads (i.e. the infrastructure) at regular intervals. This architecture allows a centralized management of the dissemination of vehicle messages, thus offering efficient scheduling and optimization. However, the deployment of RSUs is expensive not only at the time of their installation, but also during their lifetime due to maintenance costs. Both the management of the equipment obsolescence and the compatibility of the various applications which are embedded in RSUs and deployed in different regions are two challenges. In addition, V2I architecture raises a critical issue: how will the driver assistance applications still be operational when the infrastructure is down, and what about the areas that are not equipped?

In order to address this latter issue, vehicle-to-vehicle (V2V) communication that works without any preexisting infrastructure has been introduced as a second architecture. The vehicles collaborate in a distributed manner to form an ad hoc network [1]. Especially, V2V communication can help to bypass a failure in the infrastructure by providing a complementary and redundant communication structure in order to guarantee an adequate level of quality of service for safety applications. However, V2V communication relies on distributed routing algorithms which performance is more unpredictable than that of V2I protocols.

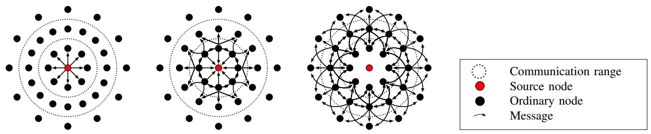
The V2V routing protocols can lead to either a flat topology or a hierarchical topology [2]. The organization of the network according to a hierarchy consists in differentiating certain nodes of the network by giving them a particular role or specific functions. This hierarchical organization is one key factor that can be used by ad hoc routing protocols to optimize network management and to improve scalability. Several approaches have been proposed in order to achieve this hierarchical organization such as multipoint relaying (MPR), clustering schemes and backbone-based approaches.

In order to optimize the flooding of broadcast traffic, the Optimized Link State Routing protocol (OLSR) [3] resorts to the concept of MPR nodes. These latter are the only nodes allowed to generate and to broadcast, through the entire network, the link state information used when building and updating the routing tables. Also, only MPR nodes are able to relay the messages from a source node to the destination. During MPR selection, each node in the network selects the smallest subset of its symmetric one-hop neighbors that allows it to reach every node in its two-hop neighborhood. A broadcast message transmission is illustrated Fig. 1. At the beginning, a source node (drawn in red) sends a broadcast message to its one-hop neighbors. When the blind flooding strategy is used (Fig. 1 (a)), each node retransmits the message that it receives, which creates a significant number of redundant retransmissions of the same message. When the MPR technique is applied (Fig. 1 (b)), only the MPR nodes (drawn in green and blue) retransmit the message, which limits the number of redundant retransmissions and therefore the load

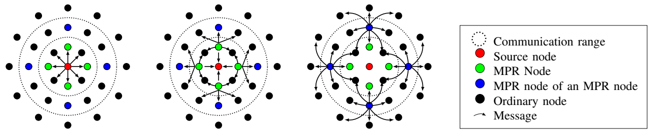
[1]Univ Lille Nord de France, IFSTTAR, COSYS, LEOST, F-59650 Villeneuve d'Ascq first-name.name@ifsttar.fr

[2]Univ. Littoral Côte d'Opale, LISIC - EA 4491, F-62228 Calais, France

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.



(a) The blind flooding strategy



(b) Multipoint Relaying (MPR) technique

Fig. 1. Blind flooding and MPR strategies

of the communication network due to routing traffic.

Clustering is another way to achieve a virtual division of the network into groups. Resorting to clusters optimizes the range of packet flooding by limiting the packet retransmission to one or several clusters according to a predefined strategy. A cluster includes different types of nodes: one group leader that is a "cluster-head" connected to a set of nodes called "ordinary members". "Gateway nodes" are members of several clusters, thus making a link between them (Fig. 2). The clustering schemes can generate separated clusters (without gateway nodes) or not (with gateway nodes). 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, the information needed for cluster management is added to the packets. The cluster size can be also characterized by the number of hops. For instance, in one-hop clusters, each ordinary member node is directly connected to its cluster-head.

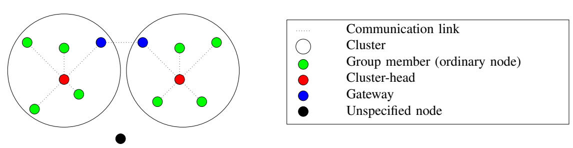


Fig. 2. Node status in clustering schemes

Over the last 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 [4]. Many of them were first studied for Mobile Ad hoc Networks (MANETs) [5]–[7]. 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. The previous approaches have been completed recently with many proposals targeting specifically VANETs by building time stable clusters [8]–[11]. In order to achieve the connection between the clusters in road traffic, several backbone approaches have been also proposed in the literature [12]–[16].

In order to evolve from the plethora of existing approaches towards standardized solutions, the European Telecommunications Standards Institute (ETSI) has published the Geonetworking requirements [17] recently that fix the design guidelines of VANET architectures. Geonetworking is a geographic-based routing protocol that supposes the existence of a location service that collects the location information of the vehicles and provides them upon request.

In a recent work [18], we presented in detail and discussed all these references. We also introduced a clustering scheme, chain-branch-leaf (CBL), that builds a virtual backbone in the VANET, similar to that obtained with RSUs, while relying only on V2V communications. CBL functioning requires only the position and velocity information of the vehicles in the closest neighborhood (one hop). Therefore, unlike most of the geographic-based routing protocols that would require an infrastructure at least for the location service, CBL can perform without any preexisting infrastructure. Moreover, CBL supports unicast communications and oriented broadcast (upstream or downstream) in the road traffic, which can be useful for some applications such as cooperative perception and cooperative localization [19]. Through simulation, we already showed that CBL reduces routing traffic load [18] in OLSR in comparison with the MPR, without degrading network performance for the applications, notably on end-to-end delay and packet delivery ratio [20].

However, our previous studies did not investigate the structures created by both MPR and CBL in order to explain why CBL performs better while selecting 75% less relays than MPR. Therefore, the purpose of this work is to present a comparative analysis of the structures created in an OLSR-based VANET by both the MPR technique and the CBL clustering scheme, and analyze their impact on network stability and performance. The rest of the paper is organized as follows. Section II describes the functioning of the native OLSR protocol using multipoint relaying. Section III recalls briefly the functioning of CBL, and provides all the details about its implementation in OLSR in replacement of the MPR technique. Section IV presents the performance evaluation carried out through simulation over MPR-OLSR and CBL-OLSR and a comparative analysis of the related results.

II. OLSR ROUTING PROTOCOL

The Optimized Link State Routing protocol (OLSR) [3] is a hierarchical protocol that uses proactive route search. The objective is to list all the intermediate nodes (relays) that will relay application packets from a sender node to one or more destination nodes. To that end, each node maintains several tables, including the routing table, the one-hop neighbor table, the two-hop neighbor table, the table of MPR nodes elected by the node, and the MPR Selector table (*i.e.* the table of the neighbor nodes that elected this node as a MPR). This routine uses two types of routing messages: HELLO messages for discovering the neighborhood and TC messages for sharing routing tables.

Each node periodically broadcasts routing packets through the network, independently from any packet transmission request. These routing packets contain the list of the routes which the node is aware of. The routing table contains at least the address of a potential destination node and the address of the first designated relay (next hop) in the one-hop neighbor table that announces a route to that destination node. According to OLSR protocol specification, a metric may be associated with each route. By default, this cost is the number of relay nodes needed to reach the destination (number of hops). When transmitting in unicast or multicast mode, a node sends its packets to the address of the next hop node to the destination node(s). Upon receipt of each packet of the message, the relay node proceeds the same way by sending it to the next relay node on the route to the destination node(s).

The first step of the OLSR protocol is to discover the neighboring nodes. Each node periodically broadcasts a HELLO message (parameter *HELLO_INTERVAL* set by default to 2 s) that contains information about its neighborhood. This information is the address and the connection type of the neighbors (symmetric, asymmetric, MPR, lost, or not specified). If the link is symmetric, this means that the link between the two nodes is valid and bidirectional. If the link is asymmetric, the link is unidirectional. If the connection has a MPR type, it means that this neighbor node is a MPR. By receiving these messages, neighboring nodes process the information (by updating their routing tables), but they do not relay this message. In fact OLSR protocol uses a TTL (*Time To Live*) counter system of which the value is included in each packet. The source node determines the TTL counter value. When a message is relayed by a node, the TTL counter is decreased by 1. A node that receives a message of which the TTL counter reaches the value of 1 does not retransmit this message. In the case of the HELLO messages, the TTL counter is always set to 1 by its source node. In addition, when a neighborhood change is detected, the routing table is recalculated to update the information. If a node has not sent any HELLO message for a time longer than a specific threshold *Vtime* (parameter *Neighbor hold time*), it is deleted from the routing tables. This timer system is used to ensure that the information is not obsolete. The OLSR protocol transmits the values needed to calculate this delay through the *Vtime* field in the header of the messages transmitted within OLSR packets.

Nodes selected as MPR broadcast TC (Topology Control) packets at a frequency set through the *TC interval* parameter. These messages are only relayed by MPR nodes. A TC packet from a node contains information about the neighboring nodes that selected it as a MPR node. By receiving TC packets, each node in the network updates its routing tables and calculates the paths to other nodes in the network using Dijkstra's shortest path algorithm with the number of hops as metric. The operation is as follows. The N_i node receives a TC message initially sent by the N_k node. The TC message contains the list of N_j nodes that have selected the N_k node as their MPR node. There is a route through the N_k node to join the N_j nodes. If the N_j nodes are present in the routing tables of the N_i node, the node calculates the current number of relay nodes that allow it to join the N_j nodes. If the number of relay nodes on the route passing through the N_k node is lower, then the N_i node updates the routing table.

In all these operations, the MPR nodes play a key role. The structure built in the VANET by OLSR protocol in order to reduce routing traffic load and save radio ressources for the application relies on MPR nodes. During the MPR selection, each node N_i in the network selects the smallest set of its one-hop neighbors, allowing it to reach every node in its two-hop neighborhood. Only MPR nodes selected by N_i are allowed to forward broadcast traffic originating from this latter. The selection procedure for node N_i can be described as follows:

- Node N_i puts in a set *U* (*U* for Uncovered) all its two-hop neighbors.
- Each neighboring node N_j having its *Willingness* parameter value set to *WILL-ALWAYS* is automatically added to the MPR set of N_i. Every two-hop neighbor connected to the nodes selected at this step is removed from *U*.
- The node N_i calculates for each one-hop neighbor N_j, the number *D1* (degree) of the nodes in the two-hop neighborhood of N_i that can be reached through node N_j. If a node N_j is the only one-hop node with a communication link to a two-hop neighbor in *U*, then node N_j is added to the MPR set of N_i. The node N_i removes any node in *U* connected to the chosen node N_j.
- The following procedure is repeated until *U* is empty (which means each two-hop neighbor of the node N_i has a communication link to one of the selected MPRs). For each one-hop neighbor node N_j, the node N_i calculates the number *D2* (reachability) of nodes in *U* that have a communication link with node N_j. Node N_i adds to its MPR set, the node N_j with the highest *N_willingness* parameter value and a non-zero value for *D2*. In the case of multiple candidate nodes, the node with the highest value for *D1* is added to the MPR set of N_i. The nodes from *U* connected to N_j are removed from *U*.

III. CBL CLUSTERING PROPOSAL

A. Definitions

CBL is a completely distributed algorithm described in detail in [18]: each communication node initiates its own process. CBL creates a hierarchy between the nodes in order to build one-hop clusters so that each node of a cluster can directly communicate to the cluster-head without going through another relaying node. It defines two kinds of nodes (Fig. 3): branch nodes and leaf nodes. Both kinds of nodes emit periodic HELLO messages in order to build a structure, called a chain, that connects the nodes in each traffic direction.

In order to build stable chains, CBL uses a metric called connection time or contact time (*CT*). This metric evaluates the duration of the connection between the network nodes and allows the nodes managing the election of their branch node. Precisely, branch nodes, leaf nodes, chain, and connection time are defined as follows:

- **A branch node** (Fig. 3) is a cluster-head node that is elected by the other nodes (branch or leaf) in its one-hop neighborhood. 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. When relaying a message, according to the application request specified in the header fields, a branch node can forward it to:

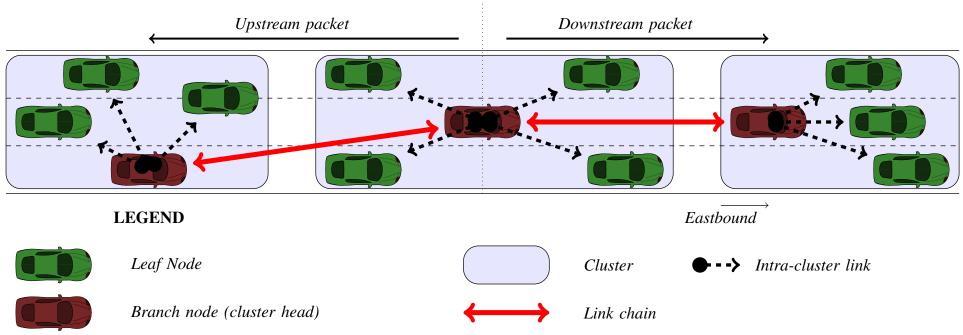


Fig. 3. Example of a CBL structure on a three-lane one-way highway

  - its own leaf nodes;
  - its upstream branch nodes;
  - its downstream branch nodes;
  - its branch nodes (even in other traffic direction).

  These destination options are coded into the message type of the original format of the packets defined in OLSR protocol [3] (see Table I).
- **A leaf node** is an ordinary node which has 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 generates and transmits only HELLO messages and applications data traffic.
- **A chain** is a virtual backbone made up of connected 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 would be obtained with an infrastructure RSUs deployed along the road. It offers to its branch nodes a path to forward application messages over long distance.
- **BranchChoice**, the address of the branch node chosen by a leaf node. The BranchChoice field is empty if the node is a branch node;
- **ChainUP**, the address of the branch node chosen for relaying upstream traffic. The ChainUP field is empty if the node is a leaf node;
- **ChainDO**, the address of the branch node chosen for relaying downstream traffic. The ChainDO field is empty if the node is a leaf node;
- **The Connection Time (CT)** is the duration expected for the communication between two nodes N_i and N_j when they keep the same speed. This metric, also called "contact time", has been used in [21]–[23]. CT is approximated using (1). This equation takes into account the position of the nodes ($[X_i, Y_i]$ for the node N_i and $[X_j, Y_j]$ for the node N_j), their speed (V_i and V_j), their steering angle (σ_i and σ_j), and the radio range (R_{max})):

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

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

B. CBL scheme

The algorithms involved in CBL behavior are described in details in [18]. In this section, we only summarize the CBL process. At the time the initialization of the network starts, no group is formed, all the nodes are ordinary (leaf) nodes. The HELLO messages exchanged by the nodes activate the topology creation process. In addition to the information usually transmitted in OLSR HELLO messages, a CBL HELLO message contains additional information such as the position and type (branch or leaf) of its originator. A node N_i performs several tasks each time it receives a HELLO message from a neighbor N_j. Firstly, it updates the table containing the list of one-hop neighbors as in OLSR. Secondly, according to the type (branch or leaf) of the receiver node N_i, it respectively changes the branch status of N_i into the leaf status if N_i no longer received HELLO messages from any neighboring node that had elected it for some remaining time, or it turns the leaf status of N_i into the branch status when the HELLO message from node N_j announces that N_j has elected N_i as a branch node. Lastly, if N_i is a branch node, it elects or updates the upstream (ChainUP) and downstream (ChainDO) branch nodes. Otherwise, if N_i is a leaf node, it elects or updates its own branch node (BranchChoice).

C. Implementation of CBL in the OLSR protocol: CBL-OLSR

CBL is actually a clustering scheme that can be integrated in any routing protocol, provided that this latter maintains a table of the one-hop neighbors by the means of periodic messages. In this work, in order to perform an accurate comparison with the multipoint relaying technique, we have chosen to integrate CBL in OLSR protocol (CBL-OLSR). In this version, the CBL branch nodes inherit the functionalities of the MPR nodes in native OLSR: generating and forwarding TC messages through the network, and relaying efficiently broadcast traffic from their leaf nodes beyond its one-hop neighborhood (through ChainUP and ChainDO).

D. Packet format in CBL-OLSR

The implementation of CBL within the OLSR protocol requires a first modification to include the message direction information. This information is added to the OLSR message header at the message type level, initially coded on 8 bits.

However, in native OLSR, the message type can only take integer values between 1 and 4. The proposed implementation uses the first four bits to code the direction information (see Table I and Fig. 3). When a message is received by a branch node, it is forwarded as follows:

1) If the value of the first bit is 1, the message is forwarded to every branch nodes within the one-hop neighborhood (including branch nodes of another traffic direction). This retransmission mode is the classical *broadcast*.
2) If the value of the second bit is 1, the message is forwarded to the downstream branch node.
3) If the third bit value is 1, the message is forwarded to the upstream branch node.
4) If the value of the fourth bit is 1, the message is forwarded to the electing leaf nodes.

Message type	1	2	3	4	5	6	7	8
Broadcast transmission	1	-	-	-				
Relaying to @ChainUP	-	1	-	-	Compliant			
Relaying to @ChainDO	-	-	1	-	with OLSR			
Relaying to electing leaf nodes	-	-	-	1				

TABLE I. Modification of the "Message Type" of OLSR packet in order to take into account the retransmission modes available in CBL-OLSR.

The message direction information allows CBL-OLSR to support the four communication scenarios [24] for automotive applications specified by the ETSI: point-to-point, point-to-multipoint, *GeoAnyCast* and *GeoBroadcast*. In CBL-OLSR, the point-to-point and point-to-multipoint communication scenarios are performed respectively by the *unicast* and *multicast* mechanisms of the OLSR protocol. The *GeoAnyCast* and *GeoBroadcast* scenarios can be achieved using the position information contained in the one-hop neighbor table and the message direction information (Table I shows the message type encoding). Let us consider a geographic zone of relevance for a message of a given source node. This latter identifies the relative position upstream or downstream of this area of interest. It sends an application message (containing a field identifying the position and size of the geographic area) to the branch nodes within its own chain in the right direction. At each relaying branch node that receives a message with *GeoAnyCast* or *GeoBroadcast* scenario, the node checks if the position of the next branch node is in the target zone. In the case of a *GeoAnyCast* scenario, when this zone is reached, the relaying node overwrites the TTL field of the message and sets its value to 1, thus allowing a last retransmission. In the case of a *GeoBroadcast* scenario, when this zone is reached, the relaying node overwrites the "Message Type" field of the message and sets its value to "1001xxxx", thus allowing local broadcast to all branch nodes and leaf nodes. Each branch node that is located in the target area and that receives this message will broadcast it. The others ignore it.

E. Header of the HELLO message in CBL-OLSR

Implementing CBL in OLSR implies some modifications to the HELLO packet format in order to include the information about the node type (branch or leaf), its position, its speed, and its steering angle. Only 32 additional bits have been added to the HELLO packet header of the native OLSR (Fig. 5) in order to limit the impact on communication resources, since HELLO messages are sent periodically with a high frequency. The different variables are coded as follows :

- **Speed**, on 8 bits, from 0 to 256 km/h with a resolution of 1 km/h.
- **Steering angle**, on 8 bits, from 0 to 360 degree according to the North, with a resolution of 1.5.
- **Longitude**:
  - (Second Longitude), on 10 bits, coding the minutes and seconds of longitudinal positioning angle.
  - (dsec Lon), on 4 bits, coding the tenths of seconds of longitudinal positioning angle.
- **Latitude**:
  - (Latitude Second), on 10 bits, coding the minutes and seconds of latitude positioning angle.
  - (dsec Lat), on 4 bits, coding the tenths of seconds of latitude positioning angle.
- **Vehicle type (T)**, on 1 bit (0: leaf node, 1: branch node).

The destination nodes of a HELLO message are in the one-hop neighborhood of the source node, thus within its direct communication range. The network technology used in this work is IEEE 802.11p, which has a maximum radio range of 1 km. Therefore, the nodes do not have to transmit the position values related to the angle and the tenths of minutes of the angle in the HELLO message, since these values are almost the same for the nodes within the same communication range.

F. Content of the HELLO message for CBL-OLSR

A modification at the *link code* level has also been made in order to include the node choices regarding the chain structure (@BranchChoice, @ChainUP, and @ChainDO). In native OLSR, the *link code* is encoded on 8 bits, the last four bits coding the link type and the node type of the neighbor as illustrated in Fig. 4. The values taken by the "Link Type" and "N.Type" fields are as follows:

- Link Type=1, link is not specified (*UNSPEC*);
- Link Type=2, link is lost (*LOST*);
- Link Type=3, unidirectional link (asymmetric, *ASYM*);
- Link Type=4, bidirectional link (symmetric, *SYM*);
- N.Type=1, there is at least one symmetric link with this neighboring node (*SYM_NEIGH*);
- N.Type=2, there is at least one symmetric link with this neighboring node that has been selected as MPR (*MPR_NEIGH*);
- N.Type=3, there is no symmetric link with this neighboring node (*NOT_NEIGH*).

In CBL-OLSR, when Link Type=SYM_LINK and N.Type=MPR_NEIGH, the information about @BranchChoice, @ChainUP, and @ChainDO are included in the HELLO message. To that purpose, three bits of the *link code* are then used, as illustrated in Table II, to code the storage order of these neighboring node addresses in the fields of the message following that *link code*.

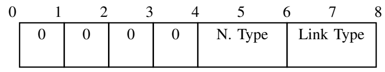
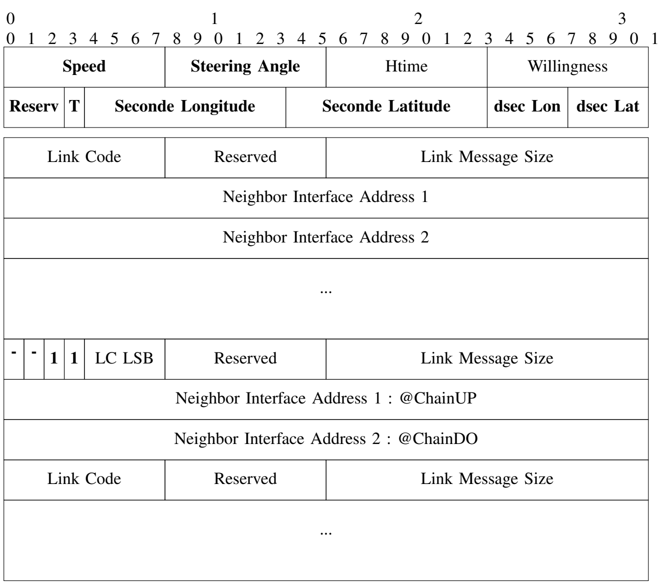


Fig. 4. Link code defined by OLSR protocol

Link code	1	2	3	4	5 6 7 8
@ChainUP in 1st position	-	-	-	1	
@ChainUP not elected	-	-	-	0	
@ChainDO in 1st position	-	-	1	0	
@ChainDO in 2nd position	-	-	1	1	Compliant
@ChainDO not elected	-	-	0	-	
@BranchChoice in 1st position	-	1	0	0	
@BranchChoice in 2nd position	-	1	1	0	
@BranchChoice in 2nd position	-	1	0	1	with OLSR
@BranchChoice in 3rd position	-	1	1	1	
@BranchChoice not elected	-	0	-	-	

TABLE II. Proposed modifications of the link code for CBL-OLSR in order to include the order of appearance of the choices for relaying nodes (@BranchChoice, @ChainUP, and @ChainDO)

Fig. 5 shows an example of a HELLO message sent by a branch node that has selected a branch node upstream (@ChainUP) and another downstream (@ChainDO).



LC LSB : the 4 Least Significant Bits of the OLSR Link Code coded such as LinkType=4 and N.type=2

Fig. 5. Example of a HELLO message modified for CBL-OLSR

IV. Performance evaluation

This section presents the simulation-based performance evaluation of OLSR, carried out with MATLAB, when using respectively the MPR technique and the CBL clustering scheme over various highway scenarios and network traffic conditions.

A. Road configuration and simulation parameters

SUMO (Simulation of Urban MObility) [25] is used in order to generate the mobility traces of the vehicles over three road networks, R1, R2, and R3:

- R1 is a 5 km-long three-lane one-way highway;
- R2 is a 5 km-long three-lane two-way highway;
- R3 is a 5 km-long three-lane two-way highway, with an entrance and a highway exit. The exit is located at 1.8 km and the entrance at 3 km from the beginning of the road section.

Also, low, medium and high traffic density are modeled in using a ratio of 1/6 trucks and 5/6 cars (Table III); therefore defining three scenarios per road network. 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. The same parameters are used for SUMO simulation than those taken in our previous study [18].

TABLE III. 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.

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

B. Comparison of CBL-OLSR and native OLSR structures

As mentioned in section I, previous simulation-based evaluation through OPNET showed that CBL reduces the global number of TC generated and retransmitted in the VANET in comparison with the MPR technique [18]. Moreover, it was shown that CBL actually offers better performance to application traffic regarding the delay and packet delivery metrics [20]. In these evaluations, we will compare the structures created in a OLSR-based VANET, respectively when using CBL and MPR. Therefore, fifteen structural metrics (Ms) are introduced, and they can be described as follows:

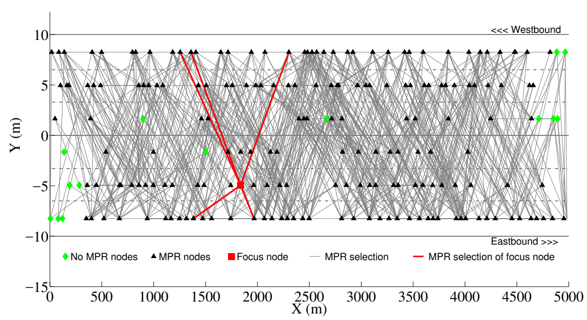
- Ms1 (Leaf/Vanet): number of leaf nodes in the network
- Ms2 (NB_Chains): number of chains in the network
- Ms3 (Branch/Chain): number of branch nodes per chain
- Ms4 (one-hop/Branch): number of one-hop neighbors (in the same traffic direction)
- Ms5 (Leaf/Branch): number of leaf nodes that attach to the same branch node
- Ms6 (Branch_time): duration that a node remains as a branch node
- Ms7 (Leaf_time): duration that a leaf node remains attached to the same branch node
- Ms8 (Branch/Node): the number of branch nodes selected by a node (branch or leaf) in the network
- Ms9 (Branch/Broadcast): the number of branch nodes involved in a single broadcast transmission
- Ms10 (Non_MPRs/Vanet): number of nodes that do not have a MPR status
- Ms11 (Nodes/MPR): number of nodes that have selected a node as a MPR (size of its MPR selector set)

- Ms12 (MPR_time): duration that a node remains a MPR
- Ms13 (NToMPR_Time): duration that a node remains attached to the same MPR
- Ms14 (MPRs/Node): number of MPRs selected by a node (size of the MPR set of the related node)
- Ms15 (MPR/Broadcast): number of MPR nodes involved in a single broadcast transmission

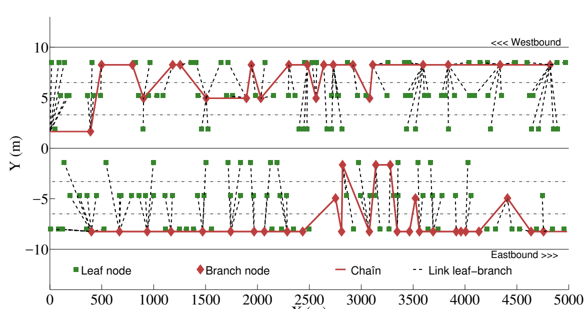
Multipoint relaying technique design assumes an open area, and it leads to a mesh structure where each node can reach the others through several relays. Once every nodes have performed their MPR selection, the network contains a large majority of MPR nodes (Fig. 6a). When assuming a physically constrained area such as road traffic, it seems reasonable to choose only one relay for upstream traffic, and another one for downstream. CBL design follows that idea, and it leads to a structure similar to a chain along the road in each traffic direction (Fig. 6b). The nodes, which speed is close to that of the major part of the traffic, evolve in the chain as branch or leaf nodes, while those moving with higher speed only evolve as leaf from branch to branch.

Only the results of S3 scenario, related to R1 road configuration, will serve for the comparative analysis. However, the results for all the other scenarios are summarized in Table IV for CBL-OLSR metrics, and Table V for MPR-OLSR metrics.

About 50% of the time (Fig. 7), CBL builds a single connected chain in the network. Sometimes the chain is broken into two or three smaller chains, mostly due to the changes in the order of the branch nodes (overtaking) inside the chain, but it is quickly reconstructed.



(a) State of the connections between VANET nodes and their selected MPRs. Focus on one Node and its MPRs



(b) State of the chains built by CBL: two separate chains are created, one in each road traffic direction.

Fig. 6. State of the structures built in scenario S5 at time T=500s

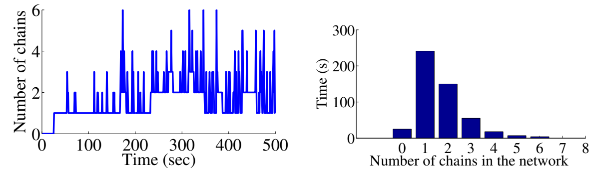


Fig. 7. Number of chains (Ms2) and accumulated time in the case of scenario 3 with CBL

Fig. 9 and 7 together show that there are about 30 branch nodes when there is only one chain, and about 10 branch nodes per chain in the presence of several chains. We noticed that the chain breaks increase with the density and the road configuration, R3 scenarios reaching the highest scores due to a lot of vehicles entering or leaving the road section (see Ms2 in Table IV).

When the traffic becomes stable (Fig. 8), after the first 150 s of simulation, there are about 200 vehicles on the road section. Using CBL-OLSR, 85% of the nodes are of leaf type and only 15% actually act as relays (branch nodes). MPR-OLSR leads to 90% of the nodes acting as relays (MPR status) and only 10% are non-MPR nodes. Since not all the MPR nodes are involved in the retransmission of a specific broadcast traffic, these results only explain why CBL-OLSR generates less routing traffic than MPR-OLSR. Indeed, though the traffic related to HELLO messages is the same for both protocols, only 15% of the nodes generate TC messages using CBL-OLSR, while 90% when using MPR-OLSR.

These results are confirmed in every scenario, 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. However, even in this cases, the results obtained using CBL-OLSR are still better than those provided by MPR-OLSR. This is confirmed in every scenario (see Ms10 in Table V), where the proportion of non-MPR nodes never exceeds 25%.

One explanation to these results is the fact that in MPR-OLSR each node may select several MPRs, as much as necessary in order to cover every two-hop neighbors. Thus, at any time, the union of the different sets of MPRs selected by each node may have almost the same size than the entire VANET. Contrary to that, CBL-OLSR forces each node to select only one or two branch nodes, depending on whether the node is a leaf or a branch. Almost all the nodes crossing the area covered by a branch node will attach to this latter, and adopt themselves a leaf status. As a result, CBL selects a number of branch nodes approximately equal to the length of the road section divided by the double of the communication range. The denser the VANET is, the lower the percentage of branch nodes will be.

Each node has an average of 38 one-hop neighbors (Fig. 10) using both protocols, which is normal since the network discovery process through HELLO messages is exactly the same. However, an unexpected result shows that both protocols achieve almost the same number of nodes attached to each

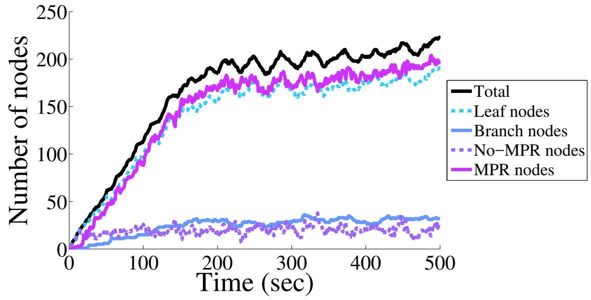


Fig. 8. Node type repartition (Ms1 and Ms10) in scenario S3

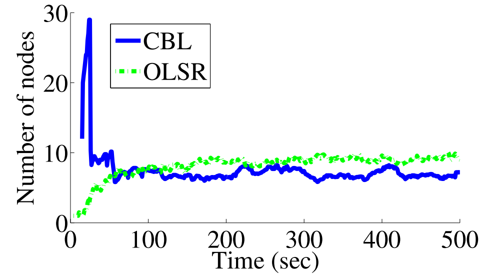


Fig. 11. Number of nodes attached to a relay node (Ms5 and Ms11) in scenario 3

relay (Fig. 11): the number of leaf nodes per branch (Ms5) is almost equal to the number of selectors per MPR (Ms11). This observation is limited to scenario 3. In the other scenarios, there are more nodes in the MPR selector sets than leaf nodes per branch. The first reason is that each MPR selector set includes both non-MPR nodes and other nodes that are themselves MPR nodes, while the leaf nodes that attach to a branch in CBL cannot be branch nodes themselves. The second reason is that a leaf node cannot attach to several different branch nodes in CBL-OLSR, while in MPR-OLSR a node may select several different MPR nodes.

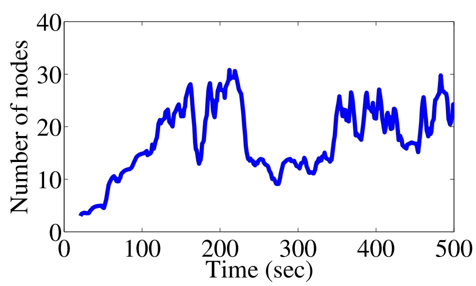


Fig. 9. Number of branch nodes per chain (Ms3) in scenario 3

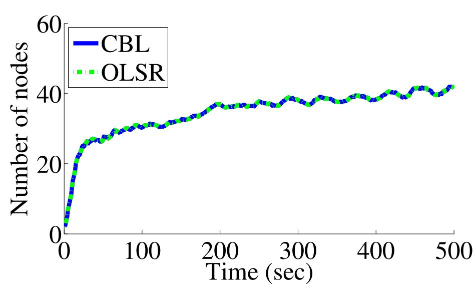


Fig. 10. Number of one-hop neighbors (Ms4) in scenario 3

Each selected node remains a branch for about 70 s (Fig. 12). Even for a vehicle moving at the lowest speed of 80 km/h allowed on French highways in normal traffic conditions, it remains a branch over 1.5 km (three times the maximum range), which is a significant distance. Indeed, in the same conditions, a RSU would serve the nodes that attach to it only over a distance two times the maximum range. On the other hand, the time that a node remains a MPR is approximately 35 s, which is the half of the branch duration. The difference between the branch duration and the MPR duration tends to decrease in the two-way highway (R2) and in particular road traffic configurations, such as those with an entrance and an exit (R3). Despite that, the duration that a node remains a branch node is longer than the one a node remains a MPR node in every scenario (see Ms6 in in Table IV, and Ms12 in Table V).

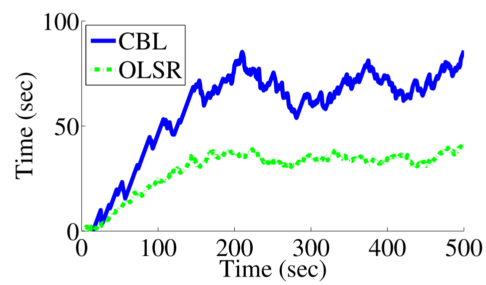


Fig. 12. Duration that a node remains a relay node (Ms6 and Ms12) in scenario 3

Each leaf node remains attached to the same branch for about 21 s (Fig. 13). For most V2V safety applications that are known to have a message transmission periodicity ranging from 50 ms to 500 ms, these 21 s represent enough time to send at least up to 40 alerts from a leaf attached to the relaying branch node to the entire network. These values are almost the same for every scenario (see Ms7 in Table IV). Except in the scenarios S1 and S7, that have the lowest vehicle density profiles, the duration that a node remains attached to the same MPR is lower than the duration a leaf remains attached to its branch node (see Ms13 in Ms12 in Table V). In scenario S3, this duration using MPR-OLSR is approximately 9.4 s, half of the duration observed when using CBL-OLSR. Both the duration a node remains a relay (branch node or MPR node) and the duration that a node (leaf or MPR selector) remains attached to the same relay (branch or MPR) suggest that the structure created using CBL-OLSR is more prone to stability than that obtained with MPR-OLSR.

Fig. 14 shows the number of relay nodes selected by each node, otherwise the size of the MPR Set of each selecting node in MPR-OLSR, and the number of branch nodes selected by each node in CBL-OLSR (a single one for leaf nodes, and up to two for branch nodes, which implies an average close to one due to the high proportion of leaf nodes). In scenario S3, after the stabilization of the network, this number reaches approximately 4 when using MPR-OLSR. The results in every

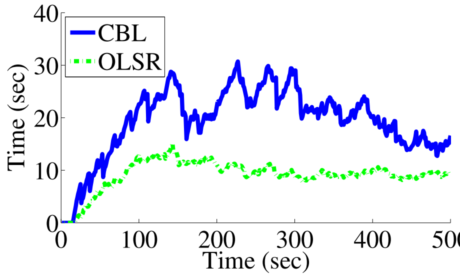


Fig. 13. Duration that a node remains attached to the same relay node (Ms7 and Ms13) of scenario 3

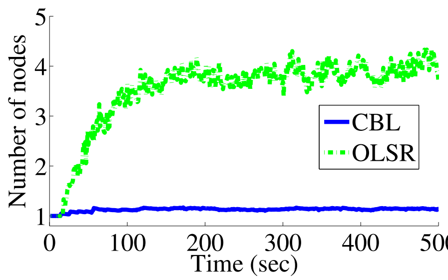


Fig. 14. The number of relaying nodes (Ms8 and Ms14) in scenario 3

scenarios present the same proportion between CBL-OLSR and MPR-OLSR, except in S1 (lowest VANET size, lowest traffic density). In other words, each broadcast traffic generated in the network is retransmitted about four times more when using MPR-OLSR than when using CBL-OLSR.

These comparisons may seem a bit unfair for the MPR technique. Indeed, the MPRs are expected to guarantee that every two-hop neighbors will be reached, which is not strictly guaranteed with branch nodes. When using CBL, each leaf node elects a single branch node and a branch node elects one or two branch nodes (upstream and downstream), while MPR-OLSR imposes the selection of as much as necessary MPR nodes in order to reach the two-hop neighborhood. It would be difficult to establish which of them achieving an efficient one-hop based clustering (CBL) or two-hop based clustering (MPR) is philosophically better. However, it is possible to establish which approach actually performs well regarding broadcast traffic flooding (routing and application ones), and the performance for unicast traffic. We noticed that, for this latter category, previous work [18] concludes that CBL is better. Also, it has been showed that by reducing the number of relays (branch nodes), CBL-OLSR generates lower TC routing traffic than OLSR. Finally, Fig. 15 shows that the number of MPR nodes involved in the retransmission of a broadcast message from a source to the entire network (through its MPRs, then the MPRs of its MPRs, and so on) is almost three to four times higher than the total number of branch nodes involved in the same process, in every scenario except S1. This ensures that CBL also reduces the traffic related to TC message retransmissions.

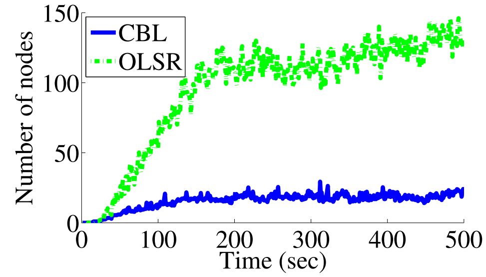


Fig. 15. The number of relay nodes involved in a broadcast transmission (Ms9 and Ms15) in scenario 3

TABLE IV. Mean result values in all scenarios for CBL-OLSR.

Network	R1			R2			R3		
Scenario	S1	S2	**S3**	S4	S5	S6	S7	S8	S9
Density	-	+	**++**	-	+	++	-	+	++
Node	25	102	**198**	50	205	369	46	189	390
Ms1 (%)	28	76	**85**	25	77	84	46	67	76
Ms2	1.65	1.96	**2.04**	3.29	2.68	3.63	3.36	6.36	9.29
Ms3	13.7	15.9	**18.9**	12.1	20.4	17.8	9.4	10.4	11.3
Ms4	3.9	18.8	**38.6**	4.8	21.2	38.2	4.1	17.7	37.8
Ms5	1.94	4.75	**7.03**	1.94	4.89	6.84	2.31	3.43	4.42
Ms6 (s)	64.4	71.9	**70.0**	65.1	73	73.5	59.4	53.6	57.7
Ms7 (s)	18.8	20.6	**21.3**	15.6	22.5	22.6	8.6	23.8	16.5
Ms8	1.60	1.20	**1.13**	1.62	1.20	1.14	1.41	1.26	1.20
Ms9	17.6	24.1	**29.5**	36.9	47.8	56.6	25.3	61.5	94.1

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

TABLE V. Mean result values for all scenarios with MPR-OLSR.

Network	R1			R2			R3		
Scenario	S1	S2	**S3**	S4	S5	S6	S7	S8	S9
Density	-	+	**++**	-	+	++	-	+	++
Node	25	102	**198**	50	205	369	46	189	390
Ms10 (%)	22.1	7.4	**10.2**	6.9	6.5	7.5	11.5	7.8	8.0
Ms11	3.0	6.8	**9.0**	5.6	11.2	13.9	5.1	10.8	14.3
Ms12 (s)	39.1	41.6	**34.6**	53.4	53.8	53.7	43.9	48.3	50.7
Ms13 (s)	29.0	13.0	**9.4**	10.9	8.0	6.6	10.1	8.4	6.0
Ms14	1.8	3.30	**3.3**	3.0	5.2	6.2	2.8	5.2	6.7
Ms15	13.9	66.9	**118**	37.2	176	317	31.4	156	308

V. Conclusion and Future Work

When using CBL clustering scheme, the vehicles that move at lower speed in the same traffic direction are good candidates (branch nodes) 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 15% to 25%, contrary to MPR nodes that never decrease under 75% of the VANET nodes. In this way, CBL allows a better optimization of the flooding of broadcast traffic, including TC message generation and retransmissions. Indeed, we showed that four times more MPRs are involved in the flooding of a given broadcast traffic, when compared to the number of the branch nodes involved in the same situation. Then, among all the one-hop neighbors of a given branch, only those having the better link quality with this latter (25% to 55%) actually choose it as their branch (cluster-head). The others select other branch nodes, which will result in a global structure with better quality link in the VANET. Finally, this study shows that CBL leads to significant stability since a node elected as a branch remains a branch for 70 s, and it can serve each of its leaf nodes for 20 s, while MPR nodes only remain in this role for a duration about 35 s and is attached to its selectors only 9 s.

Future work will consist in finding optimal values of CBL parameters for different road configurations and traffic conditions, including when integrated in other routing protocols for vehicular ad hoc networks than OLSR.

References

[1] L. Rivoirard, M. Wahl, P. Sondi, M. Berbineau, and D. Gruyer, "CBL: A Clustering Scheme for VANETs," in *VEHICULAR 2017-The Sixth International Conference on Advances in Vehicular Systems, Technologies and Applications*, 2017, pp. pp–19.

[2] 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].

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

[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] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] C. Wu, S. Ohzahata, Y. Ji, and T. Kato, "How to utilize interflow network coding in vanets: A backbone-based approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2223–2237, 2016.

[14] P. K. Sahu, E. H.-K. Wu, J. Sahoo, and M. Gerla, "Bahg: Backbone-assisted hop greedy routing for vanet's city environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 199–213, 2013.

[15] 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.

[16] 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.

[17] "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.

[18] L. Rivoirard, M. Wahl, P. Sondi, M. Berbineau, and D. Gruyer, "Chain-Branch-Leaf: A clustering scheme for vehicular networks using only V2v communications," *Ad Hoc Networks*, vol. 68, pp. 70–84, Jan. 2018. [Online]. Available: https://doi.org/10.1016/j.adhoc.2017.10.007

[19] L. Rivoirard, M. Wahl, P. Sondi, D. Gruyer, and M. Berbineau, "A cooperative vehicle ego-localization application using v2v communications with cbl clustering," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, June 2018, pp. 722–727.

[20] L. Rivoirard, M. Wahl, P. Sondi, M. Berbineau, and D. Gruyer, "From multipoint relaying to chain-branch-leaf: Improving the clustering in olsr for vehicular ad hoc networks," in *2017 IEEE Symposium on Communications and Vehicular Technology (SCVT)*, Nov 2017, pp. 1–5.

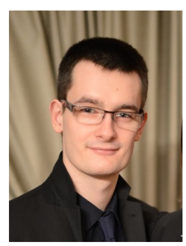
[21] W. Su, S.-J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *International Journal of Network Management*, vol. 11, no. 1, pp. 3–30, 2001.

[22] 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.

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

[24] "Intelligent Transport Systems (ITS); vehicular communications; geonetworking; part 2:scenarios," Technical Report ETSI EN 302 636-2 V1.2.0 (2013-07), 2013.

[25] 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.



Lucas Rivoirard received the Engineer degree in Transport and Mobility Engineering from the Ecole Nationale des Travaux Publics de l'État, and a Master degree from INSA Lyon. He joined IFSTTAR in 2015 where he is preparing his PhD thesis. His research areas of interest include protocol engineering and quality of service in wireless communications for advanced driving assistance systems and cooperative vehicles.

Dr. Martine Wahl received the Engineer degree in electronics from Polytech Paris-Sud (France) in 1991 and the PhD from Grenoble Institute of Technology in 1997. She joined the IFSTTAR institute as a full time researcher in embedded communication in 1998. Her research areas of interest include communication protocols for wireless and onboard communication systems.

Dr. Patrick Sondi received his PhD in Computer Sciences in 2010 at the University of Valenciennes. He joined the University of Littoral and Côte d'Opale as Associate Professor in 2013. His research areas of interest include protocol engineering, quality of service, security and simulation of wireless networks, especially their application in transportation systems.