Proceedings of 7th Transport Research Arena TRA 2018, April 16-19, 2018, Vienna, Austria

A new clustering structure for VANET

Lucas RIVOIRARD [a]*, Martine WAHL [a], Patrick SONDI [b], Marion BERBINEAU [a], Dominique GRUYER [c]

[a] *Univ Lille Nord de France, F-59000 Lille, IFSTTAR, COSYS, LEOST, F-59650 Villeneuve d'Ascq*
[b] *Univ. Littoral Côte d'Opale, LISIC - EA 4491, F-62228 Calais, France*
[c] *IFSTTAR, COSYS, LIVIC, F-78000 Versailles, France*

Abstract

The Vehicular Ad-Hoc Network (VANET) paradigm offers the opportunity of extending Intelligent Transport System (ITS) by supporting its applications through vehicle-to-vehicle communications, notably in the areas where the infrastructure is inexistent, in failure, or overloaded. However, the complexity induced by ad hoc network management raises many challenges that have to be solved such as the sharing of bandwidth resources, the limitations on the duration of the connections between the vehicles, and the application-specific quality of service (QoS) requirements. Recently, the Chain-Branch-Leaf clustering scheme (CBL) has been proposed for vehicle-to-vehicle ad hoc routing that combines the information of road configuration, vehicle mobility, and link quality in order to build an efficient clustering connecting the entire VANET through a flexible backbone. This work presents a comparative study between the native Multipoint relaying clustering used in the Optimized Link State Routing (OLSR) and CBL scheme. The results show that CBL reduces significantly the routing traffic overhead compared to native OLSR, thus freeing up more bandwidth for ITS applications and reducing the IP delays for peer-to-peer applications.

Keywords: Clustering; Routing protocols; Cooperative vehicles; V2V; VANET; Performance evaluation

* Corresponding author. Tel.: +0033 320438503
E-mail address: lucas.rivoirard@ifsttar.fr, martine.wahl@ifsttar.fr

1. 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, or safety. The architecture complexity of these systems comes from the nature of networks and the variety of envisaged applications. Many challenges have to be solved such as the sharing of frequency bands resources, the duration limitation of the connections between vehicles, or the system scalability for heterogeneous fleet with different densities.
There are different architectures designed for VANET. The most basic one is based on vehicle-to-infrastructure (V2I) communications. Road side units (RSU) are deployed along roads (i.e. the infrastructure) at regular intervals. The V2I architecture offers a centralization of messages at the infrastructure level allowing exchange scheduling and optimization. However, the deployment of RSUs is expensive not only at the time of their installation, but also during their lifetime because of the maintenance cost. Both the management of the equipment obsolescence and the compatibility of various, even numerous, applications integrated on-board RSUs and deployed in different regions are two real challenges. In addition, it raises the issue of knowing whether the driver assistance applications will be functional when some road side units break down or the V2I system is down.
To address this latter issue, vehicle to vehicle (V2V) communication architecture works without infrastructure. Vehicles collaborate in a decentralized manner to form an ad hoc network. The computational time and the bandwidth are reduced because they are distributed among vehicles. The operating of V2V networks is modular and flexible. When a node is down, the V2V structure can be rebuilt on the fly. Especially, V2V communication can help to face up to failed infrastructure in providing a complementary and redundant communication structure in order to guarantee an adequate level of quality for safety applications. However, V2V communication need distributed routing algorithms, enabling end-to-end data to be routed within the network. Such algorithms, defined in the network layer, are more complex than those of V2I. They require taking into account the specific context of vehicular applications.
This paper deals with a new ad hoc network topology called Chain-Branch-Leaf (CBL) which optimizes the communication medium in defining regions of interest aiming to limit the message spread. A clustering scheme is presented that combines information on road configuration, vehicle mobility, and link quality in order to build a structure similar to a vehicular network infrastructure, but that is only based on V2V communication. This clustering scheme can be added to any ad hoc routing protocol in order to optimize the flooding and to simplify route maintenance. The security considerations will not exposed in this work and are being investigated in other collaborative research work Chebbi et al (2017).

The purpose of this work is to present a comparative analysis of this clustering scheme compared to the multipoint relay (MPR) system of the Optimized Link State Routing protocol (OLSR). For that, CBL has been implemented into a copy of OLSR protocol. Other works attempt to improve the OLSR protocol in the context of VANETs. Chauhan et al (2016) presented a literature review of them. For instance, in order to optimize values of OLSR parameters, Toutouh et al (2012) defined an optimization problem to automatically find optimal configurations of the routing protocol. A cost function optimized a weighted average of the packet delivery ratio (PDR), the routing traffic load (RTL), and the end-to-end delay (E2E). But, the evaluation was only processed on a 10-to-60-node network for an urban area of 1,200×1,200 m^2. Furthermore, focusing on the obtained median value of PDR, E2E and RTL, the authors showed that the default values of OLSR parameters, defined in Clausen et al. (2003), gave a better packet delivery ratio than those computed and minimized the number of hops. We therefore uses two cases for the evaluation shown in this paper: the default parameters and half their value. Another study, Mehra (2016), proposed to add clustering to the MPR concept of OLSR as we propose. Their OLSR-C protocol creates 2-hop clusters, which cluster-head election depends on the node density. CBL clustering uses only 1-hop clusters with other metrics. Mehra (2016) and Toutouh et al (2012) modelized realistic vehicle trajectories using SUMO software as we do.

The rest of this paper is organized as follows. Section 2 presents OLSR and its native MPR clustering scheme. The CBL clustering algorithm is presented in section 3. The comparative analysis of OLSR and CBL performances is presented in section 4, before the conclusion.

2. Hierarchical network organization with multipoint relaying method

OLSR, Clausen et al. (2003), is a proactive ad-hoc routing protocol that computes periodically the shortest paths from each source node to all known destination nodes in its local network topology graph. In order to optimize the flooding of broadcast traffic, the clustering scheme of OLSR introduces the concept of MPR nodes. These latter are the only nodes allowed to generate and to broadcast through the network the link state information used to build and update routing tables. Also, only MPR nodes are able to relay messages from a source node to its destination. During MPR selection, each node in the network selects the smallest subset of its symmetric 1-hop neighbours that allows it to reach every node in its 2-hop neighbourhood. A detailed study of the MPR selection algorithm is available in Qayyum et al (2002).

Fig. 1 illustrates a broadcast message transmission. At the beginning, a source node (drawn in red) sends a broadcast message to its 1-hop neighbours. In Fig. 1 (a), the classic flooding strategy is used: each node retransmits the message that it received which creates a significant number of redundant messages. In Fig. 1 (b), the MPR method is used: only the MPR nodes (drawn in green) retransmit the message, which limits the congestion of the communication network.

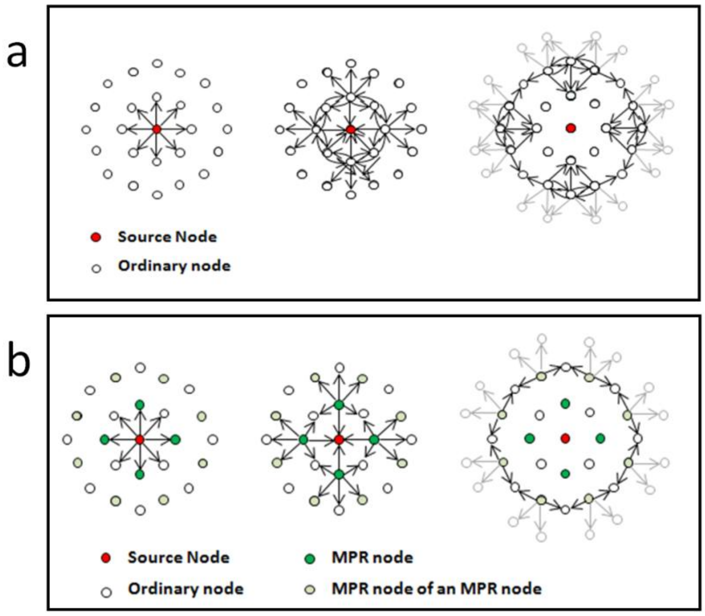

Fig. 1 Illustration of the sending of a broadcast message in the case of: (a) the forwarding strategy of the classical flooding method; (b) the forwarding strategy of multipoint relays (MPRs).

3. Context and description of CBL

The automation of the driving activity requires the enhancement of the communication links between the systems which contribute to the driving process: cooperative vehicles and smart highways. However, current works in the literature suppose the presence of an infrastructure allowing communication between vehicles and the infrastructure (V2I). But, assuming that the infrastructure failed, one challenge is to design an ad hoc V2V communication system which allows both close and remote communication, Da Cunha (2014). Communication between close vehicles will be necessary to share variables periodically (such as speed, acceleration, positioning information) and messages between near vehicles. Such exchanges will be useful to coordinate the relative movements of vehicles in future autonomous systems. Communication over long distance, upstream or downstream the traffic flow, will be necessary to forward warning messages to remote vehicles, also to manage the road traffic (for instance, upstream messages to prevent risks of bottleneck, downstream message to inform of the approach of a priority vehicle, such as a police car or a fire truck).

Thus, assuming that the infrastructure can fail, we designed a cooperative shared clustering algorithm, called Chain-Branch-Leaf (CBL), which builds a backbone allowing both the communications in close neighbourhood and remote communications between distant vehicles according to the specific needs of each road system application. CBL algorithm does not need information from the infrastructure. However, it requires that each vehicle is equipped with a GPS that enables self-localization and determines its speed. Also, it needs an on-board V2V wireless ad hoc communication card enabling communication with the other vehicles.

CBL is a completely distributed algorithm described in detail in Rivoirard et al (2017): 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 defines two kinds of nodes (see Fig. 2): branch nodes and leaf nodes. Both kinds of nodes emits periodic HELLO messages to build the CBL hierarchical structure. In that structure, connected branch nodes are links of a chain used to relay messages from branch to branch over long distance, up to their destination(s), upstream or downstream the traffic flow. Typically, they are used to remote communication. Each branch node has its set of leaf nodes attached to it where close communication occurs. A set of leaf nodes and its attached branch node defines a 1-hop cluster. Two branch nodes linked are 1-hop neighbours. But, their sets of leaf nodes are disjoint sets.

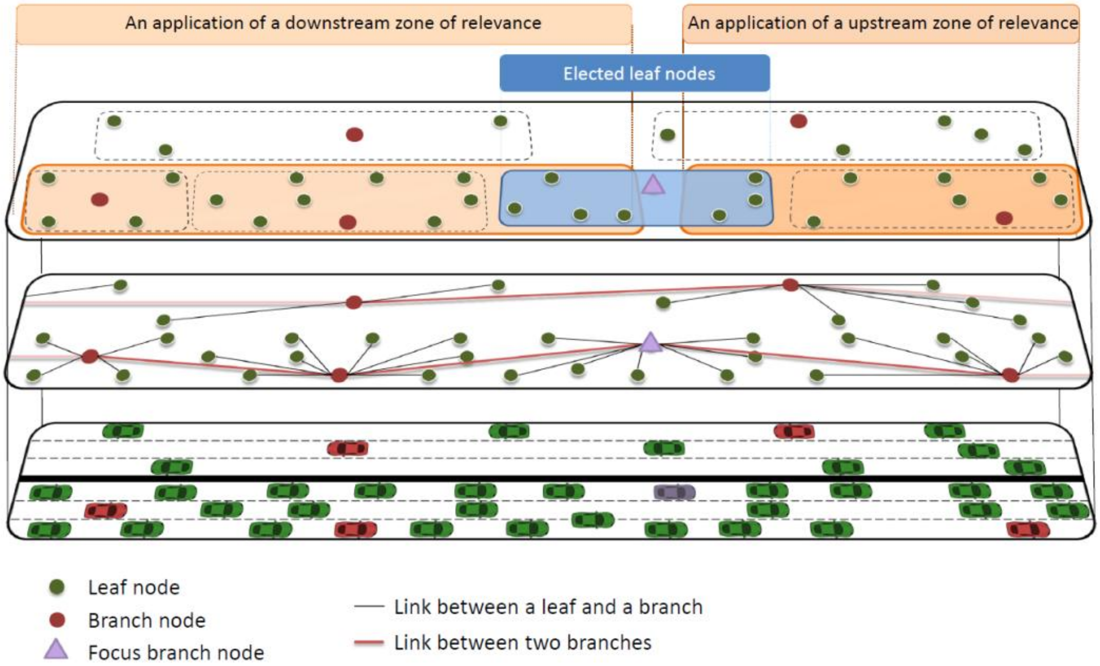

Fig. 2: CBL view of a three-lane one-way highway: at the bottom, the road traffic; at the middle, the CBL nodes and links; at the top, upstream and downstream zones on both sides a branch node.

Each branch node is elected by other nodes (branch or leaf) in its 1-hop neighborhood. To build stable chains, CBL uses a metric called "connection time" (*CT*) or "contact time". That metric which evaluates the duration of the connection between the network nodes are used by nodes to manage the election of their branch node.

Precisely, branch nodes, leaf nodes, chain and connection time are defined as follows.

- *A branch node* is a cluster-head node which is elected by other nodes (branch or leaf). It emits HELLO messages like every node, but it is the only 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:
  – its leaf nodes;
  – upstream branch node;
  – downstream branch node;
  – all branch nodes (including branch nodes of another traffic direction).

- *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.
- *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 by Y. Li (2013). CT is approximated using the equation (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 and their direction $[V_i, \theta_i]$ for the N_i and $[V_j, \theta_j]$ for the node N_j; and the maximum radio range (P). The Fig. 3 shows the value of the connection time between two vehicles in the same direction or opposite directions according to their relative speed and the transmission range.

$$CT = \frac{-(ab+cd)+\sqrt{(a^2+c^2)*\mathrm{P}^2-(ab-bc)^2}}{a^2+c^2} \quad (1) \qquad \begin{cases} a = V_i\cos(\theta_i) - V_j\cos(\theta_j) \\ b = X_i - X_j \\ c = V_i\sin(\theta_i) - V_j\sin(\theta_j) \\ d = Y_i - Y_j \end{cases}$$

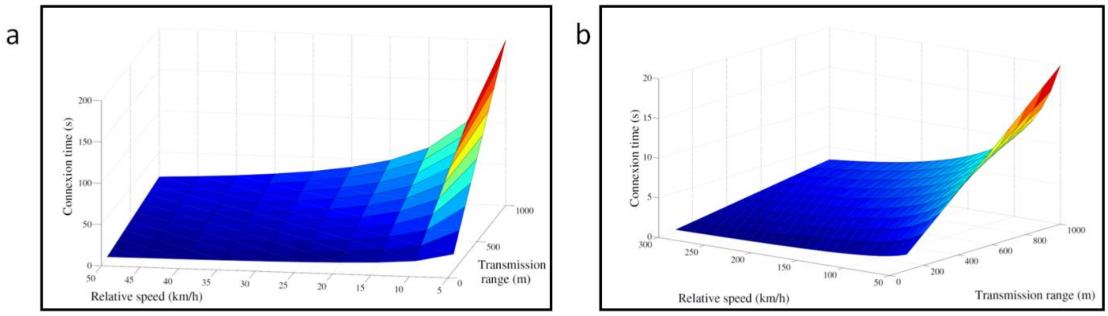

Fig. 3: Connection time between two vehicles driving: (a) in opposite directions; (b) in the same direction. The maximum CT value of two vehicles driving in opposite directions is about 20 seconds; CT can be upper than 200 seconds for vehicles in the same direction.

4. Performance evaluation

4.1. CBL implementation in OPNET Riverbed Modeler

The OPNET Riverbed Modeler is a simulation platform to evaluate network performance. The platform is split up into three layer models, namely network layer model, node layer model and process layer model as illustrated Fig. 4.

CBL has been implemented inside a copy of the OLSR process defined in OPNET Riverbed Modeler**.** However, CBL can be implemented inside any other ad hoc routing protocol in adding beacon messages to it if such messages are not already provided. In particular, the destination options previously mentioned in section 3 (leaf nodes, upstream or downstream branch node, all the branch nodes) are coded into the link code of the original format of the HELLO packets defined in OLSR protocol. Furthermore, the current position, the type (leaf or branch) of the sender node and its linked nodes are added to the HELLO message header. Therefore, CBL HELLO messages are longer than the OLSR ones.

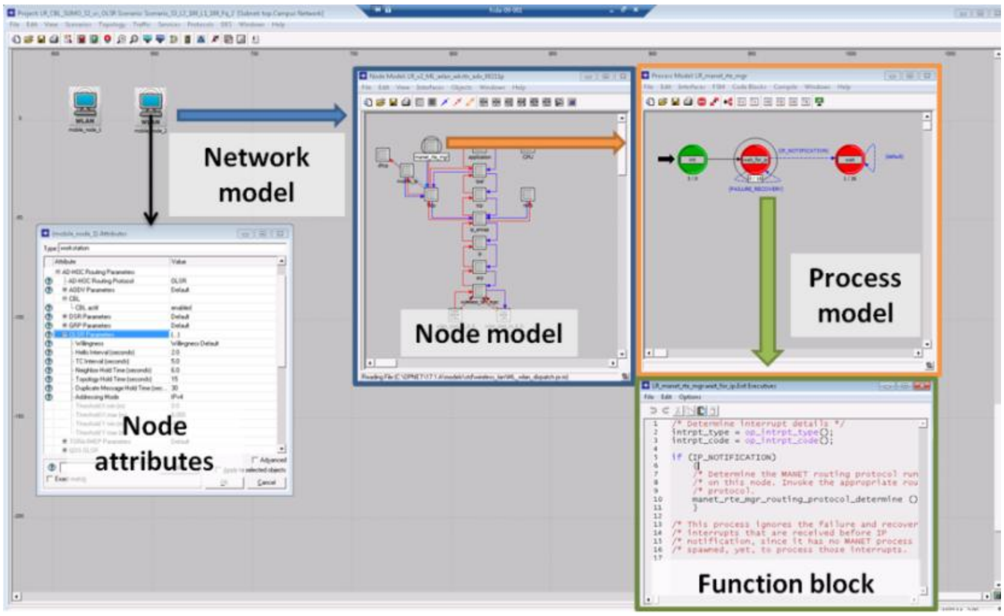

Fig. 4 The three layers in OPNET Riverbed Modeler: the network model, the node model and process model. CBL implementation was carried out within the OLSR process model

4.2. Trajectory file generation with SUMO software

The Simulation of Urban MObility (SUMO) software, presented in Santana et al. (2015), is an open source, microscopic road traffic simulator. We used SUMO in order to generate the mobility traces of the vehicles over the road network. Fig. 5 shows the steps of conception of a trajectory in the Opnet ".TRJ" format from data entered in the SUMO simulator. At first, the road network has to be cut into small linear pieces before performing a SUMO simulation. A file in a ".nod.xml" format has the nodes of these small pieces in; another file in a ".edg.xml" format contains the set of junctions between those nodes. Secondly, the NETCONVERT tool generates the road network into a file in a ".net.xml" format. Furthermore, the type (cars or trucks) and the number of vehicles driving on the network are specified in a ".rou.xml" format. Then, the SUMO simulation runs and provides trajectory traces that are converted into the OPNET ".TRJ" format. Finally, each ".TRJ" trajectory obtained (one trajectory per vehicle) can be attached to an OPNET node in the OPNET Riverbed Modeler tool.

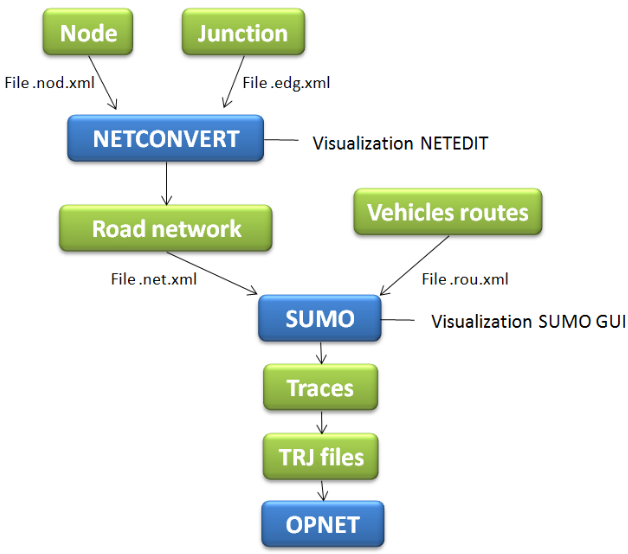

Fig. 5 Flow chart showing the generation of an OPNET trajectory per vehicle after the production of SUMO traces

In that work, the road network studied is a 5 km-long three-lane one-way highway. We define two types of vehicles: cars and trucks. A ratio of 1/6 trucks and 5/6 cars is considered. The traffic densities used 2000 vehicles per hour for the car traffic and 400 vehicles per hour for the truck traffic. The default car following model included in SUMO simulator is a variant of the Krauβ model: each vehicle drives up to its “desired speed”, while maintaining a perfect safety distance with the leader vehicle (the front vehicle). The speed limit is set at 130 km/h, which corresponds to the legal speed limit on highways in France.

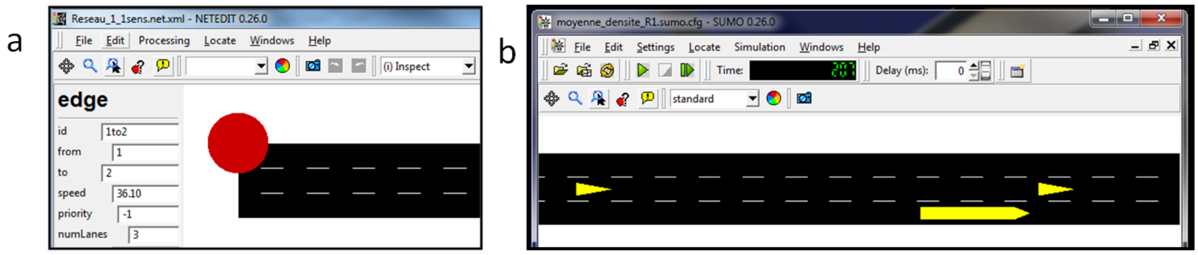

Fig. 6 Display tools provided with SUMO. (a) display of the road network under NETEDIT, where it is possible to stretch and parameterize the edges and junctions of the road network; (b) display of the simulation result with the presence of vehicles (two cars and a truck) under SUMO GUI.

Vehicles are modeled by a set of parameters which values are given in Table 1. The “desired speeds” are modeled with speed distributions to achieve realistic car following behaviour. Otherwise, vehicles would have the same “desired speed” and they would be unable to catch up with their leader vehicle, thus causing unrealistic situation. Speed distribution functions under SUMO use two parameters: *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.

4.3. Node configuration attributes in OPNET Riverbed Modeler

The simulated system is an ad hoc network of vehicles equipped with IEEE 802.11p cards with a data rate fixed to 13 Mbit/s and at a frequency band to 5.0 GHz. The transmission power is set at 0.005 W and the receiver sensitivity at -95 dBm in order to obtain a communication range of 500 m and to match the IEEE 802.11p dedicated short range communication.

Table 1. Kinematic parameters for cars and trucks used in the SUMO file “rou.xml”

	Unit	Value for Car	Value for Truck
The acceleration capability of vehicle	m/s^2	2	1
The deceleration capability of vehicles	m/s^2	3	2
The vehicle length	m	5	15
The Krauβ driver imperfection (between 0 and 1)	-	0.5	0.5
The driver-desired minimum time headway	s	1	1
The offset to the leading vehicle when standing in a jam	m	2.5	5
The maximum velocity of the vehicle	km/h	150	130
The vehicles expected multiplicator for lane speed limits	-	1	0.84
The standard deviation of the speed-Factor	-	0.1	0.1

The OLSR parameter values recommended by the RFC 3626 are reported in Table 2 (column “Case A”). Furthermore, we consider the “Case B”, where OLSR parameter values are half of those recommended, to also take into account the velocity of VANET nodes in the comparison of CBL and MPR. The HELLO interval concerns the periodic HELLO messages sent by every node to discover its 1-hop neighbours. The Topology Control (TC) interval concerns the TC messages sent by every MPR. TC messages are used by each node in order to build and update its routing table. Implementations of 802.11p and OLSR used are those provided by the Riverbed OPNET Modeler. The CBL scheme has been implemented in a copy of the OLSR model code.

Table 2. Parameter values used to tuned OLSR and CBL process model in *Riverbed OPNET modeler*

Attribute	HELLO interval	TC interval	Neighbor hold time	Topology hold time	Duplicate message hold time
Case A	2 sec	5 sec	6 sec	15 sec	30 sec
Case B	1 sec	2.5sec	3 sec	7.5 sec	15 sec

4.5. Comparative analysis of simulation results

In the following, we refer to the native OLSR using MPR method as "OLSR", and we mention the version that uses CBL as the clustering scheme as "CBL".
It can be noticed that the multipoint relaying technique leads to a higher number of MPRs than that of the branch nodes selected by CBL (Fig. 7). After the network stabilization period (after 2 minutes of simulation), the number of MPRs in the case of OLSR varies between 60 and 80 nodes in a 95-node network at the same time. On the other hand, the number of MPRs in the case of CBL varies between 15 and 25 nodes. This is explained in particular because CBL forces each node to select a single branch node while the MPR strategy does not impose any limit. There is not much difference between the two cases A and B for OLSR and for CBL concerning the number of MPR and branch nodes.

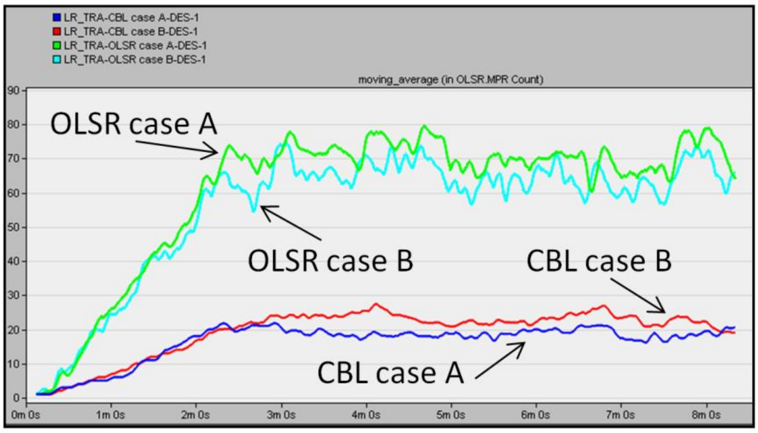

Fig. 7 The number of MPRs versus time in the network with OLSR and CBL protocols (cases A and B).

Fig. 8.a shows a higher HELLO routing traffic sent in bits/s for the CBL than for the OLSR. However, the number of HELLO messages is directly related to the number of nodes in the network (249 nodes in total and 90 at the same time on the highway section). This means that the HELLO traffic is approximately the same for OLSR and CBL. The difference is due to the larger size of the HELLO messages defined in the CBL protocol which contain in addition the position of the transmitter node (see section 4.1). Comparing the results obtained with cases A and B once the network simulation is stable, Fig. 8.a reveals that the HELLO traffic is double with parameter values of case B (90 kbits/s). Indeed, the HELLO frequency is also double in case B.
Concerning the TC messages, the traffic is directly related to the number of MPRs or branch nodes which are the only ones sending and transmitting this kind of messages. Fig. 8.b shows that OLSR generates in average 0.25 Mbits/s with parameter values A and 0.38 Mbits/s with parameter values B whereas CBL generates in average 10 times less. The TC routing traffic that is much more limited with CBL than with OLSR allows freeing up radio resources for the application traffic.

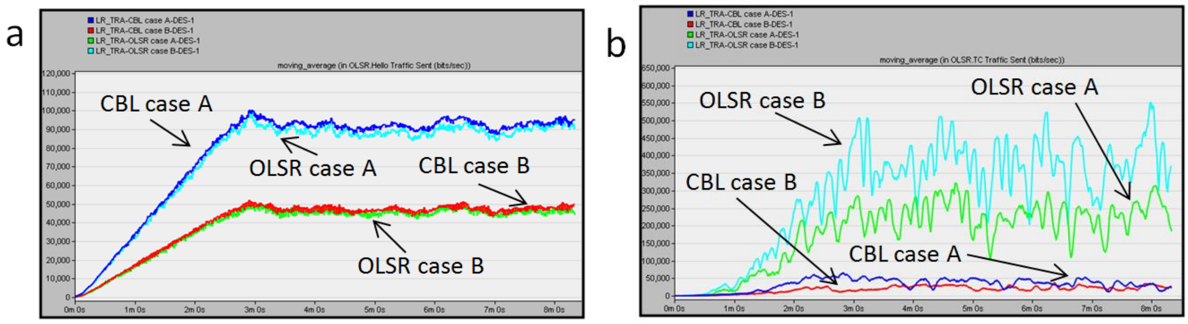

Fig. 8 Routing traffic sent in bit/sec versus time: (a) HELLO messages; (b) TC messages.

Despite the advantage of CBL regarding the number of relays, it should be noticed that the multiplicity of relays of the OLSR scheme allows a redundancy that may favour robustness. Therefore, it is also necessary to compare how both OLSR and CBL clustering schemes impact the routing of application traffics in VANET. This work focuses on future real-time VANET applications such as those related to the autonomous vehicle. In such safety applications, on-board functions will have to share variables periodically for their inner process. The throughput of the application chosen for evaluations complies with the IEEE data traffic settings defined in E.T.C.I.T (2012): a 300-byte packet length is sent with a 10-Hertz update rate. That traffic is evaluated for the case B as a unicast flow during 20 seconds between node 42 and each of the three nodes 63, 83, and 105, respectively located at 1 km, 2 km, and 3 km away.

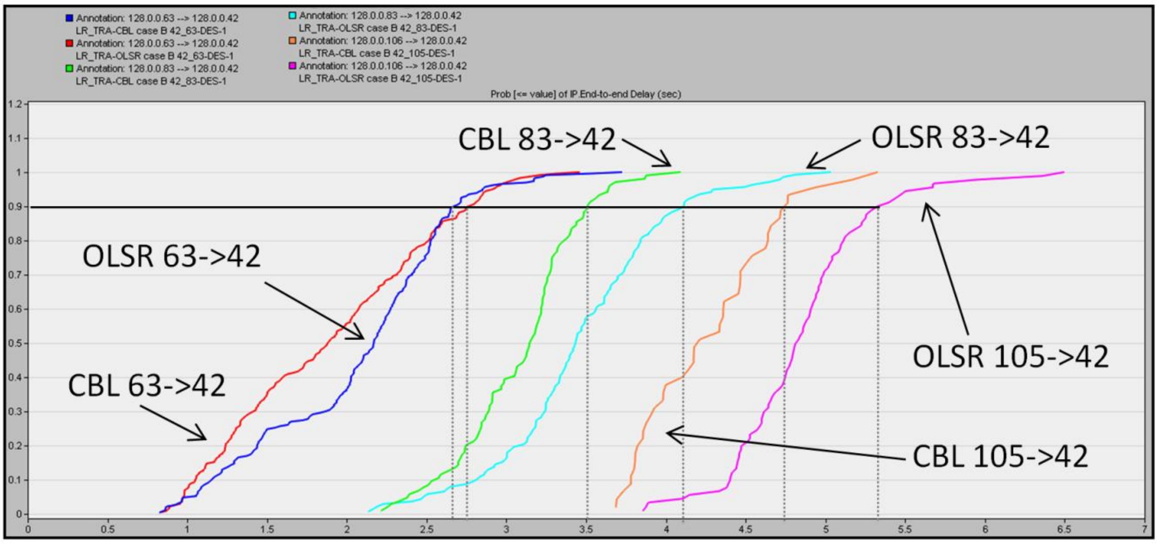

Fig. 9 Cumulative distribution of IP delays (in millisecond) between nodes (63,42), (83,42), and (105,42) at a relative distant of respectively 1, 2, and 3 km.

The “end-to-end delay” metric concerns the time taken by a packet to reach its destination. It is measured as the difference between the arrival time of a packet at its destination and the creation time of this packet. This metric help to evaluate the routing protocols in relation to the application requirements. Fig. 9 shows the cumulative distribution of the end-to-end IP delays between the couple of nodes (63,42), (83,42), and (105,42).

IP delays between the nodes (63,42), 1 km away, are almost identical for OLSR and for CBL (red curve and dark blue). IP delays between the nodes (83,42), 2 km away, are lower for CBL than for OLSR (green curve and light blue). Looking at the ninth decile of IP delays (the horizontal black line drawn at the 90 % value of the cumulative distribution), CBL is at 3.5-ms delays and OLSR at 4.1-ms delays. This phenomenon is further accentuated for nodes (105,42), 3 km away: 90% of IP delays are below 4.6 ms for CBL and 5.4 ms for OLSR. Comparing values of IP delays obtained at 90% and 100% (the maximum values), we note that the gap between OLSR and CBL is rising. Therefore, IP delays grow with the distance between the source and destination nodes. That is due to the increase of the number of hops required to forward messages up to the destination. That number is in average two hops for (63,42), four hops for (83,42) and seven hops for (105,42). We observe that the maximal IP delays after seven hops, obtained in the case of two vehicles at a distance of 3 km, is 6.5 ms. This time is smaller than 10 ms which is a current time in the real-time process automation area.

Conclusion

We presented a new clustering scheme for VANETs which optimizes communication links and limits the routing traffic. Contrary to the multipoint relaying clustering scheme designed for open areas MANETs, the CBL scheme takes benefit from the spatial limitations of the roads and achieves an efficient clustering which leads to reduced number of relays in comparison with native OLSR. As a result, though the routing traffic related to TC messages (both sent and retransmitted) is drastically reduced. The whole routing traffic (TC plus HELLO) is lower with CBL than with OLSR despite of a slow increase of the CBL HELLO routing traffic due to a longer header in the message. The simulation-based evaluations performed on realistic road traffic scenarios based on real-world data show that CBL leads to a reduced utilization of the bandwidth for routing traffic, which benefits to the applications traffic. The results show that the IP delays over peer-to-peer application traffic shows that CBL has better performance than native OLSR. Even for up to seven-hop communications, the IP delays observed remain under 10 ms, this allows imagining the deployment of real-time control applications over significantly long road sections. Future work will focus on the study of specific road safety applications in various road traffic scenarios, including highways, subways and street centers.

Acknowledgment

The authors thank the “Direction interdépartementale des routes du Nord” for the traffic data on the A27 road. They 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

Chauhan, R., Goyal, N., & Kumar, R. (2016). “A Review on Tuning Of OLSR Routing Protocol IN VANET”. *International Journal of Advanced Research and Innovative Ideas in Education*, *2*(2), 508-512.

Chebbi, E., Sondi, P., & Ramat, E. (2017). “Enhancing Formal Proofs of Network Protocols for Transport Systems using Discrete Event Simulation”, *SIMUL 2017, the 9th International Conference on Advances in System Simulation*.

Clausen, T. et al. (2003). “Optimized link state routing protocol (OLSR)”, IETF,no. RFC 3626.

Da Cunha, F. D., Boukerche, A., Villas, L., Viana, A. C., & Loureiro, A. A. (2014). “Data communication in VANETs: a survey, challenges and applications” (Doctoral dissertation, INRIA Saclay; INRIA).

E. T. C. I. T. System (2012). “Intelligent Transport Systems (ITS); STDMA recommended parameters and settings for cooperative ITS; Access Layer Part”, Technical Report ETSI TR 102 861 V1.1.1 (2012-01).

Li, Y., Jin, D., Wang, Z., Zeng, L., & Chen, S. (2013). “Exponential and power law distribution of contact duration in urban vehicular ad hoc networks”. *IEEE Signal Processing Letters*, *20*(1), 110-113.

Mehra, R., Bali, R. S., & Kaur, P. (2016, March). “Efficient clustering based OLSR routing protocol for VANET”. In *Colossal Data Analysis and Networking (CDAN), Symposium on* (pp. 1-7). IEEE.

Qayyum, A., Viennot, L., & Laouiti, A. (2002, January). Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on* (pp. 3866-3875). IEEE.3

Rivoirard, L., Wahl, M., Sondi, P., Berbineau, M., & Gruyer, D. (2017, July). CBL: A Clustering Scheme for VANETs. In *VEHICULAR 2017-The Sixth International Conference on Advances in Vehicular Systems, Technologies and Applications* (pp. pp-19).

Santana, S. R., Sanchez-Medina, J. J., & Rubio-Royo, E. (2015, February). How to simulate traffic with sumo. In *International Conference on Computer Aided Systems Theory* (pp. 773-778). Springer, Cham. .

Toutouh, J., García-Nieto, J., & Alba, E. (2012). Intelligent OLSR routing protocol optimization for VANETs. *IEEE transactions on vehicular technology*, *61*(4), 1884-1894.