VANET distributed data fusion for traffic management

Romain Guyard, Véronique Cherfaoui

VANET distributed data fusion for traffic management

Romain Guyard[1], Véronique Cherfaoui[1]

***Abstract*—In this article, we propose a distributed fusion algorithm to detect traffic congestion through the exchange of messages in vehicle network. This algorithm is based on the Dempster-Shafer theory that manages the uncertainties on data and sources of information. Each vehicle updates its database with local measurements (speed and interdistance) and information received from other vehicles and can calculate its route. Thanks to the collaboration, smart cars can avoid congested roads and take a better path to their destination. Several variants of the algorithm are studied and compared to a centralized approach through experiments carried out on the SUMO simulator using real urban road networks.**

I. Introduction

In the domain of traffic management, most algorithms use a centralized server (eg. Waze, Google map...) that collects data (position, speed...) sent by connected vehicles. The server computes a map of congested roads. This map is then sent back to the vehicles that can use it to find the fastest path. In some cases the path is directly computed by the central server.

In this paper we study an other method of collaboration based on direct connection between cars in the Vehicular Ad-Hoc Network (VANET) context. The VANET has some characteristics radically different from regular computer networks. Nodes are dynamical and connections between nodes are only temporary since their relative positions change all the time. A WiFi norm called 802.11p has been proposed for VANET. Messages can only be sent in a broadcast mode since there is no possible durable connection between nodes.

Traffic management has been heavily studied and different methods has been used to detect traffic jams in VANET. A common solution is to analyze a local map of cars like proposed in [1] with the application Trafficview. Trafficview generates an instantaneous snapshot of the traffic and not an average over time. Moreover it is sensitive to Sybil attacks[2] and needs trust between nodes which can not be accepted in real conditions. [2] proposes an efficient traffic congestion detection protocol (ECODE) for intelligent transportation systems. This method, as in [3], sends the position of cars in the network which can be a privacy concern. [4] proposed to count the number of messages exchanged to determine the traffic level. This mechanism can be easily implemented in complement of other VANET applications but it can be inaccurate and can be modified artificially by a dysfunction or by an attacker. In [5], an adaptation of K-mean to V2V is used as well as a distributed hierarchical agglomerative clustering system. The method can detect the status of roads by fusing position of cars in the network. Traffic patterns recognition can be used to detect traffic congestion as shown in [6]. The author of [7] introduces the concept of vehicle to vehicle to infrastructure (V2V2I). With this method average speed are computed using V2V and then the results are centralized to an infrastructure.

In this paper, we propose an adaptation of a distributed algorithm originally proposed in [8] to perform traffic management using V2V communications. This auto-stabilizing algorithm use distributed data fusion to improve the knowledge of each vehicle. Since this method doesn't send data to a third part organization, it keeps personal data private. A centralized algorithm becomes unusable when the central server fail; whereas a decentralized algorithm is more robust in this case. Finally, our method doesn't require internet connection. In a crowdy road, mobile network can be saturated and thus unusable. This algorithm is auto-stabilizing and is resistant to transient errors due to broken or dirty sensors as proven in [9]. The algorithm is also resilient to deliberately modification of local data.

In the first section, we present the distributed data fusion algorithm and its variants based on Dempster Shafer theory (or belief functions theory) to model uncertainties in traffic management scenarii. The next section describes the implementation of the algorithm in the SUMO simulator. Finally, results of the simulations are presented and discussed.

II. Distributed data fusion algorithm

The method described in this paper is based on the algorithm presented in [8]. The objective is to fuse data coming from other vehicles with local data to determine the level of congestion of roads. Using this information, each car can find the fastest path to its destination.This algorithm has multiple steps:

1) Modeling of local data: each vehicle observes its environment, generates local data and models a representation using belief functions. Local values are independent of each others.

[1] The authors are with *Sorbonne Universités, Université de Technologie de Compiègne, CNRS Heudiasyc UMR 7253, France*

[2] The Sybil attack is an attack wherein a reputation system is subverted by forging identities in peer-to-peer networks.

1) Data reception: each vehicle receives data coming from all other vehicles in neighborhood.
2) Data fusion: information from other vehicles and the local representation are fused.
3) Data sending: each vehicle sends the fused data to other vehicles.

The algorithm updates estimations of lane average speed on a road map. Each lane of each road has a unique ID that every cars knows. The average speed can be used to determine the shortest path using for example a Disjktra algorithm.

A. Representation of road congestion status

Car sensors are imperfect. Data generated with imperfect sensors give imperfect results. Moreover in VANET, it is not possible to fully trust other vehicles. Cars can have broken or dirty sensors or an attacker can try to benefit from sending false information in order to have a better driving experience. Thus, we propose to use Dempster Shafer theory that is able to deal with uncertainties.

1) Belief functions: Dempster-Shafer theory introduced in [10] is often used to model both imprecision and uncertainties of an agent. The Dempster-Shafer framework models confidence in events using belief functions. Belief functions can be expressed by different representation. The most common representation is the mass function. If we consider a frame of discernment Ω the finite set of all possible events, then the mass function are values attached to the power set of Ω following the equation 1. Let be a variable $\omega \in \Omega$ and $A \subseteq \Omega$ a element of 2^{Ω}. The quantity $m(A)$ is interpreted as the part of the belief allocated to the hypothesis "the answer ω is in the subset A of Ω".

$$0 \leq m(A) \leq 1, \sum_{A \subseteq \Omega} m(A) = 1 \tag{1}$$

The mass $m(\Omega)$ represents the ignorance and $m(\emptyset)$ the inner conflict. A "focal element" of m is an element A of Ω such that $m(A) > 0$. A "simple mass function" has one or two focal element including Ω. If Ω is the only focal element then the mass function is called "vacuous" and represents total ignorance. If Ω is not a focal element, then the mass function is called "dogmatic". With imperfect sensors, most of mass functions are non-dogmatic and we will thus assume it is the case in this paper. Mass functions can be combined to improve the knowledge and reduce uncertainties. The well-known Dempster conjunctive rule is shown in equation 2.

$$\begin{cases} & (m_1 \oplus m_2)(\emptyset) = 0 \\ A \neq \emptyset & (m_1 \oplus m_2)(A) = \frac{1}{1-K} \sum_{B \bigcap C = A} m_1(B) m_2(C) \\ where & K = \sum_{B \bigcap C = \emptyset} m_1(B) m_2(C) \end{cases} \tag{2}$$

This rule requires the independence of sources. When a source uses multiple times the same information to generate data, data incest can occur. To avoid data incest, the fusion operator has to be idempotent. An idempotent fusion operator called cautious has been proposed in [11]. This operator is applied on the conjunctive decomposition presented by [12] using the operator minimum. This operator is idempotent, associative and commutative and noted $\owedge$.

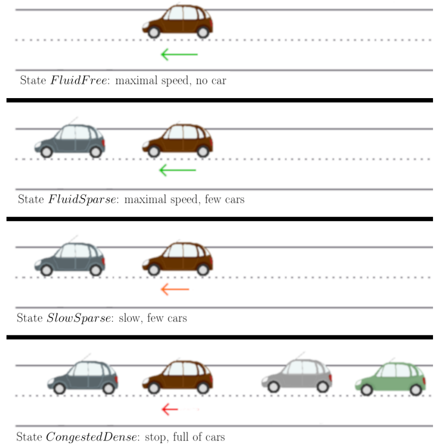

Figure 1. Road states in Ω

2) Modeling road congestion status: The congestion of a road is represented by the following frame of discernment: $\Omega = \{FluidFree, FluidSparse, SlowSparse, CongestedDense\}$. A $FluidFree$ state represents a road without any cars or with cars that can drive at the maximal speed without any cars near. The $FluidSparse$ state is also a state where cars can drive at maximal speed but the road contains cars close to the vehicle, which is a risk of traffic jam. Adding more cars to the road will make all the cars to slow down. In the $SlowSparse$ state cars are to many to keep maximum speed. Finally, the $CongestedDense$ state is the state where cars are too many and thus a traffic jam is formed. The states of Ω is summarized on the figure 1.

Two sensors are used to evaluate the congestion of the road : the speed sensor LIDARs that detect the distance between cars. A basic belief assignment function is used to transform the measurement to a belief function m^{local}. Each sensors measurement is converted to belief function separately and are then fused together.

3) Representation of the average speed on a road: The car average speed is represented by the frame of discernment $\Omega_s = \{Fluid, Slow, Congested\}$The speed value has a range of $[0, V_{max}]$, with V_{max} the speed limit of the road. Let V the proportion of V_{max}. Sigmoids are used to represent the average speed of a road. These functions change the speed value to a belief function m_s^{local}. The sigmoid function $S_s()$ is shown on the figure 2. The sensors are imperfect thus the sigmoid function does not generate a dogmatic belief function. The uncertainty is represented by mass $m(\Omega_s) = 0.3$. The parameters of the sigmoids are chosen to make the curves smooth with empiric values. In a real scenario, they should be chosen by experts of the domain or by learning methods approach.

4) Representation of the congestion of the road: The road congestion is represented by an other frame of discernment $\Omega_d = \{Free, Sparse, Dense\}$. The congestion

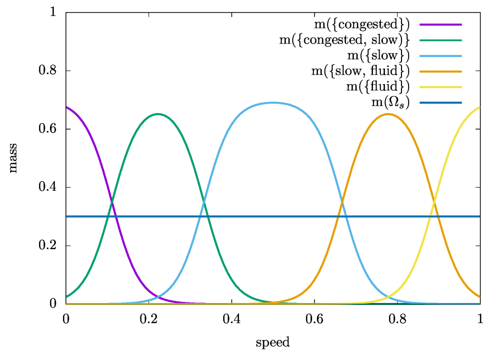

Figure 2. Speed basic belief assignment

of the road is detected using LIDARs that give the inter-distance between cars. Those inter-distances are values with a range of $[d_{safety}, d_{max}]$ with d_{safety} the safety distance that depends on the speed of the car and d_{max} the maximal distance detected by the LIDAR. d_{max} also represents the absence of detection. Let d be the rescaling of $[d_{safety}, d_{max}]$ to $[0, 1]$. Each inter-distance is changed to belief function m_d^{local}using the sigmoid functions $S_d()$, similarly to the speed representation. The uncertainty is represented by mass $m(\Omega_d) = 0.3$. The parameters of the sigmoids are chosen to make the curves smooth with empiric values. In a real scenario, they should be chosen by experts of the domain or by learning methods approach.

5) Generalization of speed and inter-distance sets of discernment: Both speed and inter-distance are represented by their own set of discernment Ω_s and Ω_d. Since we want to use them to describe the status of roads, we have to generalize them to the global set of discernment Ω. The generalization operation is noted $\uparrow$ as shown in equation 3.

$$m^{local} = (m_s^{local})^{\Omega_s \uparrow \Omega} \oplus (m_d^{local})^{\Omega_d \uparrow \Omega} \tag{3}$$

The speed sensor gives information about the states $\{DenseCongested\}$ and $\{SlowSparse\}$ and also the union of the states $\{FluidFree, FluidSparse\}$ since the speed data can not distinguish the difference between a road where cars drive at maximum speed with a "empty" road.

The inter-distance can give information about the states $\{FluidFree\}$ and $\{CongestedDense\}$ and the union of the state $\{FluidSparse, SlowSparse\}$ since cars inter-distance can not distinguish the difference between a road with some cars but driving at normal speed and a road with some cars but slowed. To have an information about the full frame of discernment, the two belief functions generated from speed sensor and the inter-distance are then combined using the Dempster's rule.

B. Data reception

The transmission of data is done using a WiFi norm called WiFi 802.11p for VANET. Data are broadcasted to other cars. Every car sends the fused data m^{public} to other cars in the neighborhood. A message is constituted of a collection of belief function representing an estimation of the status of roads. This collection, noted $[m^{public}]$, contains only non vacuous belief functions. The car collects all the messages received and saves them until the next step. When a new step starts, the saved data is fused and the database is cleared to be able to receive new messages for the following step.

We consider in the following that the messages are always received by every cars in range.

C. Data fusion

Fusion of data is done by an idempotent operator since every car uses information that have been already fused. Therefore, the algorithm uses the cautious operator.

Belief functions coming from other vehicles are representation of a distant measurement. in space and in time. Since some time has passed since the data has been generated, data received are less representative of the current situation and thus, they should be discounted before the fusion. The discounting operation reduces the importance of sources that are not precise or can not be trusted. The discounting proposed by Shafer in [10] is given by the equation 4. $m^{\alpha}(A)$ is the mass discounted with a factor of $\alpha \in [0, 1]$. The value of α depends on the reliability of the source.

$$\begin{cases} A \neq \Omega & {}^{\alpha}m(A) = \alpha m(A) \\ & {}^{\alpha}m(\Omega) = (1 - \alpha) + \alpha m(\Omega) \end{cases} \tag{4}$$

The discounting in the proposed algorithm has been studied in a previous paper [13]. The time between each step is considered as the same and thus we use a fixed discounting. The equation 5 shows the general fusion formula.

$$m_{public} = fuse(m^{local}, {}^{\alpha}[m^{public}][0] \oslash^{\alpha} [m^{public}][1] \oslash \ldots) \tag{5}$$

In [14] we proposed 4 different ways of data fusion, shown in figures 3, 4, 5 and 6 that are going to be tested in this scenario. The idea of these scenarios is to use Dempster's rule that increases the knowledge when data comes from two independent sources or the cautious operator when sources are dependent. Local value is always independent of values coming from the network. It is then possible to combine it with Dempster's rule the local value and the result of the cautious operator applied to all values sent by other cars (distributed values). Figure 4 (where $\oplus$ represents Dempster's rule and $\oslash$ the cautious rule) shows the fusion diagram of this proposition. As proposed in [15], since local data is timely independent, we can add a temporal fusion of the local value with Dempster's rule. This operation is done before the network data fusion as shown in Figure 5. Finally, we propose a fourth fusion diagram in Figure 6 that assumes that nodes send both distributed and local values. Dempster's rule can be used to combine neighbor local values since independent local data is fused only once.

D. Data sending

Fused data m^{public} are broadcasted to other vehicles. Since we fuse data for all roads of the map we have to send a belief

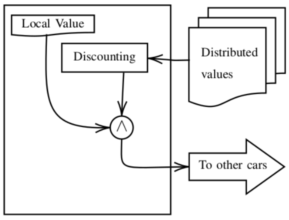

Figure 3. Cautious operator only

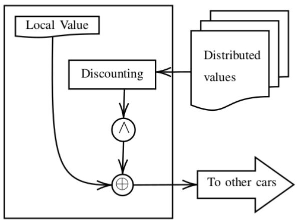

Figure 4. Dempster's rule before sending

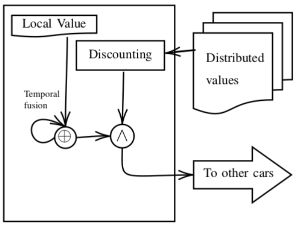

Figure 5. Dempster local value

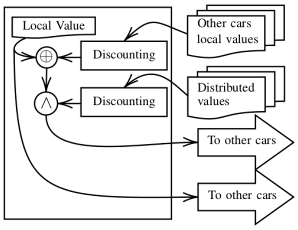

Figure 6. By sending local values to neighbors

function for each lanes of each roads. The WiFi 802.11p doesn't allow packaging and thus data has to be sent at once. Therefore, we must limit the size of message at 1500 bytes WiFi 802.11p packet. A mass can be encoded using only one byte since a precision of $\frac{1}{256}$ is enough. If we consider the conflict normalized and the sum of all masses equal to 1, a belief function can be represented using 14 bytes. We also need to add a lane identifier which can be represented using 4 bytes, assuming each car having the same map. Thus, we can send 107 belief functions. In order to choose which road information should be sent, the algorithm sorts the belief function based on the mass on the ignorance. The more commit the belief function are the more likely they are going to be sent.

III. Simulation

We have developed a simulated scenario with SUMO simulator [16], [17] to test each variation of the algorithm.

A. Scenario

The scenario must test the different characteristics of the variants of the algorithm. The traffic is simulated in multiple real cities in order to have realistic results. A constant heavy traffic is kept in order to cause traffic congestion. To do this, we set a fixed number of cars that drive in the city. Each car starts at a random location and travel to another random location in the city. Each time a car arrives at destination, it is removed from the simulation and a new car with a new path is generated.

The algorithms are compared by counting the number of cars that have reached destination after a fixed time.

To obtain realistic results, we have tested on two scenarios: one in a district of Paris and in the city of Compiègne (average french city) The two cities have a very different topology.

B. Pathfinding algorithm

The generated distributed belief function should be used to found the fastest path to a destination. The decision of the shortest path can be done using Disjktra's algorithm. The weight on each edge is the travel time computed using the maximum speed and the length of a lane. The result of the data fusion algorithm should be used instead of using the maximum speed. In order to use a belief function in Disjktra's algorithm, the average speed s_e should be estimated. We use the plausibility functions as proposed by [18].

The plausibility of a set of a mass function is an alternative representations of mass functions. The plausibility is noted Pl and it is described by the equation 6. The set of plausibility on the bayesian elements is called the contour function and it is noted pl.

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B) \tag{6}$$

A transformation function is used to change the plausibilities back to values and thus estimate the original speed modelized by the belief function. To determine our transformation function we have used the special case where the local speed values s are transformed into belief function m_s^{local} using the sigmoid functions. Our goal is to inverse the sigmoid function to find an estimation of the speed. The transformation function should return the original value s when applied on the particular case $m_s^{local} = S_s(s)$. The belief functions has to be transformed from Ω to the original speed set set of discernment Ω_s using a method called minimization and noted $\downarrow$ as shown in equation 7.

$$m_s = m^{\Omega \downarrow \Omega_s} \tag{7}$$

The plausibilities of the belief function generated with the speed sensor and discounted is used to make the estimation. In order to have a bijective function, the function $Pl_{Sum}(m) = Pl(m(\{Slow, Fluid\})) - Pl(m(\{Congested\})) + Pl(m(\{Fluid\})) - Pl(m(\{Congested, Slow\}))$ is considered. The function PlD is then defined by $PlD(s) = Pl_{Sum}(S_s(s))$ with $S_s(s) = m_s^{local}$ the speed sigmoid function. To find the speed using the plausibilities functions, we must invert the function PlD. This inverse is plotted in figure 7. It is then possible to find the estimate speed s_e with the equation 8.

$$s_e = PlD^{-1}(Pl_{Sum}(m^{public})^{\Omega \downarrow \Omega_s})) \tag{8}$$

C. Implementation

A C++ program has been implemented using SUMO simulator to test the scenario. To simulate the WiFi-P connection we have made our own implementation of a communication

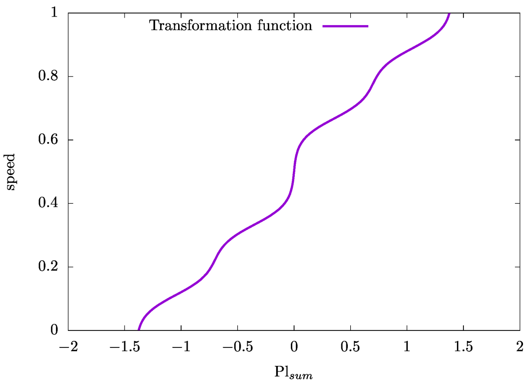

Figure 7. Transformation function PlD^{-1}

Figure 8. Map of a portion of Paris (Montmartre) in SUMO simulator

system. The theoretical range of the WiFi 802.11p is 2 km which is unrealistic in urban driving. The study in [19] showed that in ideal conditions the range can be up to 500m thus we have chosen to limit the range to 200m. The message length is limited to 1500 bytes, headers included. We suppose that our application will not be the only one that need car communications capabilities thus the data size has been limited to 20% of the maximum defined by the norm. The cities road networks are imported to SUMO simulator from OpenStreetMap.

IV. Results

In the figure 8 we can see a part of the city of Paris simulated in SUMO and the figure 9 is Compiegne. Roads are displayed in red when its occupancy is unknown, green is the road is free and yellow when it's congested.

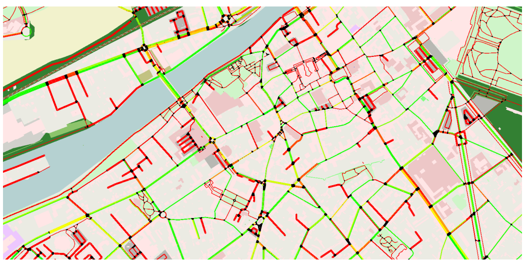

Figure 9. Crop of Compiègne's map in SUMO simulator

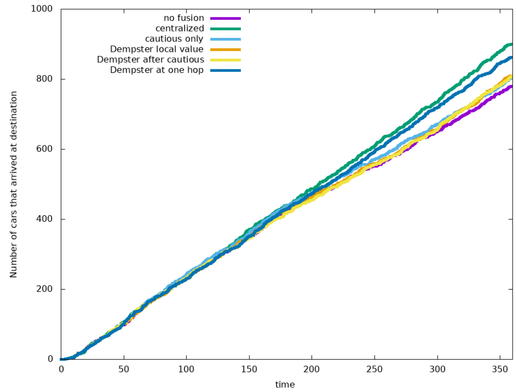

Figure 10. Number of cars that have reach their destination after a given simulation time with 50 vehicles in a part of Paris

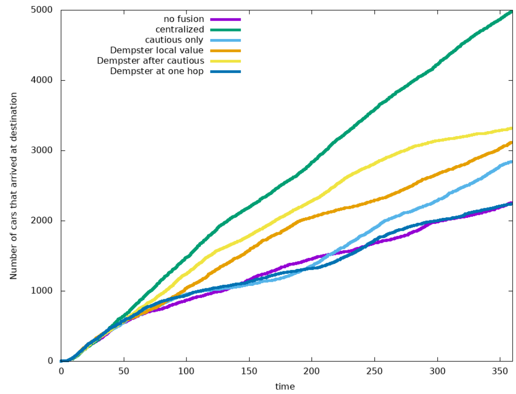

Figure 11. Number of cars that have reach their destination after a given simulation time with 500 vehicles in Compiègne

A. *Performance of path finding*

The figures 10 and 11 show the number of cars that have reached their destination after a certain simulation time for each algorithm. During the first seconds of simulation, no cars have arrived to its destination thus the curve stays at 0. After few minutes, all algorithms have the same efficacy because the estimations of occupancy are not yet known. Then, the performance of each algorithm grows at their own rates. In Compiègne, there is a significant difference between scenarios when about 500 cars are simulated at a given time. In Paris, only 50 cars are enough but the deviation is smaller. After an hour, the centralized average out performs the other algorithms. The second best is the Dempster at one hop, followed by the two other algorithms that mix cautious rule and Dempster's rule. Finally the "cautious only" algorithm is the worse, but still better than no fusion. We can see in the Compiègne scenario, with this particular random seed, that the "Dempster at one hop" has the same performance than the "no fusion" scenario. The "Dempster at one hop" algorithm have a bigger standard deviation than the other algorithms. It stays the better distributed algorithm in average but in some cases it can be the worst.

	Performance	Privacy	Pollution	Point of failure
No fusion	100%	Full	100%	None
Cautious only	117%	Full	84%	None
Dempster local value	119%	Full	77%	None
Dempster after cautious	120%	Full	75%	None
Dempster at one hop	122%	Limited	73%	None
Centralized	139%	Low	52%	Centralized server

Table I
COMPARISON OF THE ALGORITHMS ON MULTIPLE ITERATIONS OF THE SCENARIOS

B. Summary

The SUMO simulator also generates pollution emission information. The car average CO_2 pollution emission is computed for each algorithm. We can observe the link between pollution and time time spent in traffic jams. The performance of each algorithm are summarized in the table I. We added a "no fusion" algorithm where every cars choose the shortest path without taking in account the occupancy of the roads. The performance metric M is the percentage of cars that have arrived to their destinations ($nbArrived$) in comparison with method without fusion ($nbNoFusion$): $M = \frac{nbArrived}{nbNoFusion} \times 100$. In this table, the result is an average value of 10 iterations of the scenario.

The centralized method has the best performance but there are drawbacks. If a central server fails, the whole system stops working. It also needs internet connectivity to operate since the private data of the users are sent to a third party. The "cautious only" algorithm gives better results than doing no fusion but both "Dempster local value" and "Dempster after cautious" are better. Finally, the best distributed algorithm is the "Dempster at one hop" but it needs to send car's local values to neighbors. The privacy of this last algorithm is better than the centralized version since only neighbors have access to private data and not a central server that aggregate everyone's positions in the network.

V. CONCLUSION

In the paper we have shown an new application for a distributed data fusion algorithm based on the Dempster Shafer theory. We have demonstrated the utility of distributed data fusion to regulate traffic. We have shown the difference between the different variants in different realistic scenario. Despite slightly worse results than with centralized fusion, distributed data fusion shows promising results since it preserves users' private data by not sending personal information to a third party and are more resilient to malfunctions. in the absence of a single point of failure. We have also shown that data fusion reduces the amount of emissions. Future work concerns the improvement of the mass building model and the validation from real data acquired on the platforms of the Heudiasyc laboratory.

VI. ACKNOWLEDGMENTS

This work was carried out in the framework of the challenge DAPAD (Distributed and Augmented vehicle Perception to support Autonomous Driving) funded by the Labex MS2T, supported by the French Government, through the program "Investments for the future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02)

REFERENCES

[1] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "Trafficview: traffic data dissemination using car-to-car communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 3, pp. 6–19, 2004.

[2] M. B. Younes and A. Boukerche, "A performance evaluation of an efficient traffic congestion detection protocol (ecode) for intelligent transportation systems," *Ad Hoc Networks*, vol. 24, pp. 317–336, 2015.

[3] L. Lin and T. Osafune, "Road congestion detection by distributed vehicle-to-vehicle communication systems," Jan. 25 2011. US Patent 7,877,196.

[4] J. Fukumoto, N. Sirokane, Y. Ishikawa, T. Wada, K. Ohtsuki, and H. Okada, "Analytic method for real-time traffic problems by using contents oriented communications in vanet," in *2007 7th International Conference on ITS Telecommunications*, pp. 1–6, June 2007.

[5] S. Dornbush and A. Joshi, "Streetsmart traffic: Discovering and disseminating automobile congestion using vanet's," in *2007 IEEE 65th Vehicular Technology Conference-VTC2007-Spring*, pp. 11–15, IEEE, 2007.

[6] S. Vaqar and O. Basir, "Traffic pattern detection in a partially deployed vehicular ad hoc network of vehicles," *Wireless Communications, IEEE*, vol. 16, pp. 40 – 46, 01 2010.

[7] J. Miller, "Vehicle-to-vehicle-to-infrastructure (v2v2i) intelligent transportation system architecture," in *2008 IEEE intelligent vehicles symposium*, pp. 715–720, IEEE, 2008.

[8] B. Ducourthial and V. Cherfaoui, "Experiments with Self-stabilizing Distributed Data Fusion," *2016 IEEE 35th Symposium on Reliable Distributed Systems (SRDS)*, pp. 289–296, Sep 2016.

[9] B. Ducourthial, V. Cherfaoui, and T. Denoeux, "Self-stabilizing Distributed Data Fusion," *Richa A.W., Scheideler C. (eds) Stabilization, Safety, and Security of Distributed Systems. SSS 2012*, vol. 7596, pp. 148–162, Oct 2012.

[10] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[11] T. Denœux, "Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence," *Artificial Intelligence*, vol. 172, pp. 234–264, Feb 2008.

[12] P. Smets, *The canonical decomposition of a weighted belief*. Morgan Kaufmann Publishers Inc., Aug 1995.

[13] R. Guyard and V. Cherfaoui, "Study of discounting methods applied to canonical decomposition of belief functions," *21st International Conference on Information Fusion (FUSION 2018)*, pp. 2509–2516, July 2018.

[14] R. Guyard and V. Cherfaoui, "Study of distributed data fusion using dempster's rule and cautious operator," in *BELIEF 2018*, (Compiegne, France), Sept. 2018.

[15] N. El Zoghby, *Distributed data fusion in VANETs*. Phd thesis, Université de Technologie de Compiègne, Feb. 2014.

[16] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo–simulation of urban mobility: an overview," in *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*, ThinkMind, 2011.

[17] *https://sumo.dlr.de*.

[18] E. Lefevre, O. Colot, and P. Vannoorenberghe, "Belief function combination and conflict management," *Information Fusion*, vol. 3, no. 2, pp. 149 – 162, 2002.

[19] S. Demmel, A. Lambert, D. Gruyer, G. S. Larue, and A. Rakotonirainy, "Ieee 802.11 p empirical performance model from evaluations on test tracks," *Journal of Networks*, vol. 9, no. 6, pp. 1485–1496, 2014.