Dynamic Consensus for Secured Vehicular Ad hoc Networks

Jonathan Petit
Distributed and Embedded Security Group
University of Twente, The Netherlands
Email: j.petit@utwente.nl

Zoubir Mammeri
IRIT - Paul Sabatier University
Toulouse, France
Email: Zoubir.Mammeri@irit.fr

***Abstract*—Vehicular ad hoc networks provide vehicle-to-vehicle communications and safety-related applications to enhance the road safety. However, safety-related applications, like Local Danger Warning, need a high trust level in received messages. Indeed, decisions are made depending on these messages. To increase the trustworthiness, a consensus mechanism is used. With consensus, vehicles need to receive at least *X* times the same warning before making a decision. Because the consensus should meet real-time constraints of safety applications, a main issue is to set parameter *X*. In this paper, we investigate the problem of consensus and propose a generic model to define decision method involved in consensus. Then, we propose to dynamically set the consensus parameter according to the neighborhood density and the warning criticalness. The proposed mechanism enhances the "*majority of freshest X with threshold*" decision method [1] and is analytically modeled. This context-aware and data-centric security mechanism ensures a quick and correct decision. We present some simulation results that validate our model.**

***Index Terms*—dynamic consensus, security overhead, VANET.**

I. Introduction

Safety related applications such as cooperative collision avoidance, local danger warning and road hazard notification could save lives. In fact, alerts from these applications enable the drivers to react to dangerous situations such as obstacles or bad road conditions, hence reducing the risk of an accident. It is crucial to make sure that the life critical information in these applications cannot be forged or modified by an attacker. Vehicular networks are especially vulnerable to *fake attacks* where misbehaving vehicles inject erroneous information into the network to affect the behavior of the other drivers for their selfish objectives. For example, in traffic congestion optimization, honest drivers may be misled and driven to congested area by falsely injected information, while the attacker vehicle can enjoy less traffic on its own path. More dangerously, the drivers may be misled into potential accidents.

From a security point of view, the decision whether or not such an application should rely on reported hazard, is a crucial issue, which cannot be completely protected by conventional security mechanisms. Conventional solutions, such as digital signatures, focus on securing the communication network. In this way, attackers are prevented from manipulating the network. But cryptographic protection mechanisms cannot verify information itself. In other words, manipulating sensor readings to simulate a false message may still result in a perfectly signed and certified message. Therefore, an additional application-level approach is required. A technique is to evaluate the plausibility of information received during the decision process. Thus, hardening the decision process against attacks.

To provide trust into these warnings and avoid inappropriate reactions, a simple way is redundancy. The consensus mechanism provides such property. Indeed, a vehicle—that implements the consensus mechanism—needs to receive *X* times the same warning from its neighborhood before making a decision—react or warn the driver [2]. A main issue is to define the decision method that sets *X*. This could be done in two ways: static or dynamic. In the static case, *X* is set at the manufacturing of the vehicle and could be changed only with human intervention—during annual vehicle inspection for example. In the dynamic case, *X* will change sporadically. We focus on a dynamic *threshold-based* scheme which sets the minimum messages needed before reaction.

It is worth noticing that, depending on the decision method used to set *X*, the technique has an impact on the guarantees of real-time constraints of the application because of:

- the number of messages generated on the network.
- the delay to transmit one message.
- the processing time (because each message needs to be verified [3]).
- the delay to make a decision.

So the choice of the decision method should be done carefully. Moreover, each technique should be deeply investigated to assess its performance. We propose a generic model to characterize decision methods in VANET.

The remainder of the paper is organized as follows. In section II, we introduce the related work on consensus and plausibility check. Section III details the assumptions and the model used. Section IV presents the generic model to define decision methods. The main idea of this paper is to propose a dynamic decision method in order to be closer to the network state and the criticalness of the danger. Thanks to our mechanism, the decision method evaluates the warning and becomes *criticalness-based*. We present and analyze this dynamic decision method in section V.

978-1-4577-2014-7/11/$26.00

II. Related Work

Setting parameter X is a trade-off between detection power and overhead. In [2], two criteria for the choice of X are presented. The first one is based on the importance of the report. The second is based on regular inspection of the authority that could provide information on how to determine the threshold. As no available infrastructure is assumed, we only consider the first proposition. However, the mechanism proposed does not avoid false data injection attacks because the information is not verified and it does not take into account the network density.

In the context of analyzing the data and deciding whether the data is true or not, authors in [4] focused on distributed reputation systems to decide whether or not to use data. But implementing a reputation system into the infrastructure of autonomous agents in transportation networks degrades the performance. This is mainly due to the dynamic nature of the list of neighbors. Thus, not only does it require maintaining the neighbors list, since the neighbors change frequently, it is also harder to build a good reputation system [5] [6].

Ostermaier *et al.* proposed and compared four decision methods [1]:

- *Freshest message*: considers the most recent warning received.
- *Majority wins*: considers all received messages regarding the same warning and takes the majority (duplicates are not considered).
- *Majority of freshest X*: considers the majority of the recent X distinct messages (regarding the same warning).
- *Majority of freshest X with Threshold*: when the number of warnings received is greater than a *Threshold*, then the vehicles uses the "*Majority of freshest X*" method.

Their simulations showed that the "*majority of freshest X with Threshold*" is the best suited to provide protection against fake attacks in VANETs. However they did not consider how to set the *Threshold* and X, and let this issue open.

Likewise, Raya analyzed a decision method based on the *Bayesian Inference* (BI) [7]. BI uses a prior probability to compute the posterior probability of an event. Because of the dynamic topology, it is difficult to derive the prior probability in vehicular networks.

In [8], authors proposed a local verification of each message received. Their model Plausibility Validation Network (PVN) checks five rules before considering the message as trustworthy: the message duplication, the broadcast range in function of the event (by the number of hops since the initial sender), the event location, the message timestamp, and the speed of the sender. Unfortunately PVN does not avoid fake attacks.

Our proposal is three-fold. First, we propose a generic model to define decision methods. Secondly, we propose an optimization of the "*majority of freshest X with Threshold*". Thirdly, we propose to define the consensus parameters dynamically (based on the network density and the content of the warning) to ensure an adaptive and robust plausibility check.

III. System model

A. Assumptions

We assume vehicles on a multi-lanes highway which use the Local Danger Warning application (LDW). In LDW [9], vehicles exchange information about dangerous traffic situations based on local sensor readings to realize a collaborative and predictive situation-awareness. As mentioned in [1], cooperative local danger warning comprises three steps: detection, dissemination, and decision. In the detection process, vehicles detect hazards whilst driving with their on-board sensors. Whenever a critical condition is detected, the vehicle triggers the dissemination process and broadcasts a warning message—sent in a WAVE Short Message (WSM) every 100 ms [10]. Vehicles receiving such messages, trigger the decision process. If there is sufficient evidence for a critical road condition on the route ahead, the system notifies the driver to undertake appropriate reactions.

We are interested in the decision process, where the LDW application has to decide whether or not to make action or notify the driver, because leading the system into a wrong decision is one of the major threats.

In this paper, we denote as *event* the detection of a hazard (fake or not), and as *source* the first originator of the warning (whatever it is a fake warning or not). We assume a one-hop broadcast communication. We denote as *network density* or *neighborhood density*, the current number of neighbor ahead of the vehicle. We divide the set of decisions for a vehicle into two subsets:

- *Vehicle action*: brake, change lane, change path, turn, accelerate, warning light, do nothing.
- *Network action*: broadcast a message (warning or revocation), do nothing.

Thanks to the geographical coordinates of the event included in the WSM, a vehicle could detect a false warning when it overtakes the warning location. We assume that the global navigation satellite system embedded in the vehicle provides a sufficient accuracy for detecting on which lane of the highway is the vehicle. The *beaconing* mechanism provides to vehicle a local view of its neighborhood [11]. Hence, we assume that the vehicle has a spatial representation and could define what is *ahead of* and *behind* it (thanks to geo-spatial coordinates and a road map for example). We also assume that each warning has a global unique identifier and vehicles are synchronized, so a vehicle could differentiate the warnings received. We assume that vehicles have always a packet to receive.

Fig. 1 shows the different areas considered in this paper.

- *Decision area*: area allowed for collecting WSMs and making a decision.
- *Information and reaction area*: area for warning the driver. The area depends on the driver reaction time.
- *Braking distance*: the braking distance is computed from the current speed of the vehicle, the road condition (dry, wet, snowy), and the vehicle characteristics (tires pressure, brake capacity).

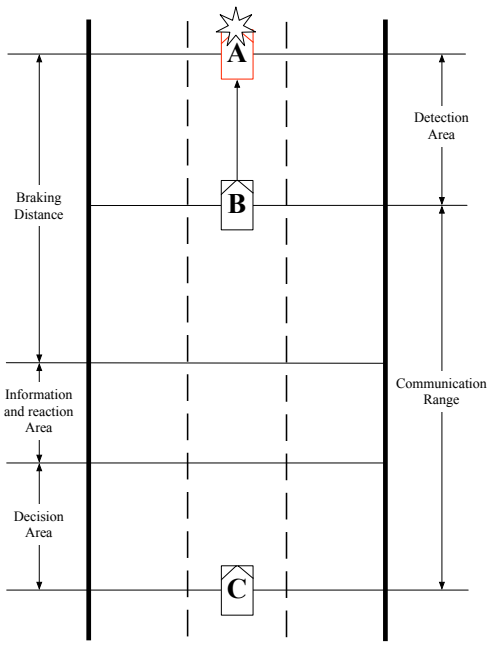

Fig. 1. Definition of areas

- *Detection area*: area where a vehicle could detect the warning with its on-board sensors. The area depends on the maximum sensor range.

From Fig. 1, we define the following notations:

- $T_{collision}$: the expected collision time computed by the speed and the distance.
- $T_{braking}$: the time of braking computed by $T_{braking} = \frac{v_k}{a}$, where a is the deceleration rate and v_k the speed of the vehicle V_k.
- $T_{reaction}$: the reaction time of the driver (0.7-1.5 second).
- T_{safety}: the time to travel the safety distance which is computed by $T_{safety} = T_{braking} + T_{reaction}$.

From these areas, we compute the maximum time allowed to make a decision before entering the braking distance. One of the goal of this paper is to provide a method to assess the consensus parameters (*X*, *Threshold*) to respect the maximum decision delay allowed. In the following, we define the *decision delay* as the elapsed time between the generation of the first warning by the source and the decision.

B. State transition diagram of a vehicle

Fig. 2 shows the state transition diagram of a vehicle. A vehicle goes from *idle* to *sending alert* when it detects a hazard. The target of the hazard could be itself (the vehicle stops because of an emergency reason), another vehicle or the environment (ice, hole, obstacle). While the hazard is still detected, the vehicle generates an alert. A vehicle goes from *idle* to *receiving alert* when it receives an alert. It keeps collecting WSMs until one of the following conditions is reached. It goes from *receiving alert* to *idle* when the hazard location is overpassed, or when the hazard has disappeared. The vehicle goes from *receiving alert* to *decision* when it

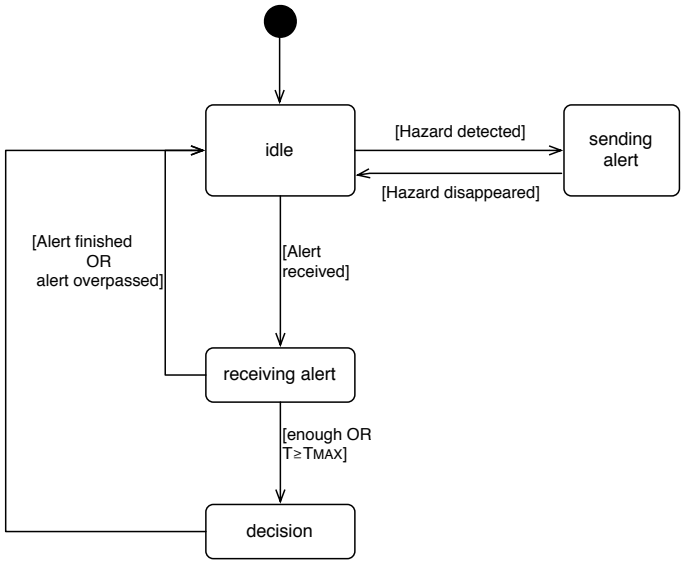

Fig. 2. State transition diagram of the OBU

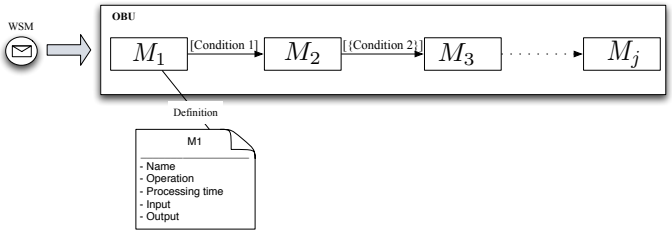

Fig. 3. Generic model

receives enough WSMs (i.e. consensus parameter). Another case of transition is when the maximum delay allowed before making a decision is exceeded. Indeed, safety-related applications have real-time constraints, and mandate to react before a specified delay (T_{MAX}). T_{MAX} is the maximum time allowed by the application before having critical impact (e.g. an accident). T_{MAX} is computed with the speed, the distance from the danger and the application design. T_{MAX} is less than 500 ms for highly time-critical applications, and equal to three seconds for time-relevant applications [12]. In Fig. 2, T is the elapsed time between the first reception and the current time.

IV. Generic model for decision method

To clearly identify the process involved in decision method, we propose a generic model. Fig. 3 shows that the model is composed by j modules (M_1 to M_j). Each module is represented by a box which has the following characteristics: name, textual operation, input and output parameters, processing time.
To define the logic and take into account the time, we add transitions. Transitions between modules are denoted by arrows. Each arrow could have one (or many) conditions to influence the model.

From the generic model is derived an analytical model.

When a vehicle has to make a decision, it starts the decision method. The vehicle has to wait for a certain number of messages before its final decision. The analytical model aims at computing this *decision delay* which corresponds to the consensus overhead. The decision delay impacts the braking distance and the authentication delay. Indeed, if a vehicle has to wait for *X* messages before making a decision (which induces a transfer delay of $X \times T_{tx}$), it has to verify X digital signatures (which induces a processing delay of $X \times T_{verify}$) [13].
The decision delay $Delay_{decision}$ of a system composed by j modules corresponds to the sum of the processing time D_i of each module i.

$$Delay_{decision} = \sum_{k=1}^{j} D_k \tag{1}$$

To mathematically model the decision method, we use the following notation. e_k^i denotes an event i of type $\lambda(i)$ from vehicle V_k. The event-specific trust is: $f(\tau(V_k), \lambda(i))$ where $\tau(V_k)$ is the default trustworthiness of the vehicle V_k (a public-safety vehicle could have a higher trust level). But the vehicle could be revoked. So, to capture this, a security status function $s : \Upsilon \in [0, 1]$ is defined (Υ is the set of vehicles). $s(V_k) = 0$ implies vehicle V_k is revoked, and $s(V_k) = 1$ means that the vehicle is legitimate. When a vehicle detects a hazard, it generates a warning message. If a vehicle travels nearby the location of a previous warning received and does not detect a hazard (it may have disappeared), then the vehicle generates a revocation message. Here, the revocation message is to revoke the warning, not the vehicle. A warning message has a weight of $b_{\lambda(i)} = +1$, and a revocation message $b_{\lambda(i)} = -1$. Each event report has a trust level:

$$F(e_k^i) = G(s(V_k), f(\tau(V_k), b_{\lambda(i)})) \tag{2}$$

G is the trust level function that returns values in the $[0, 1]$ interval. $F(e_k^i) = 0$ if V_k does not report the event *i*.
Let d_i denote the trust level computed by evaluating evidence corresponding to event *i*. The OBU assesses the trust level by applying:

$$d_i = \sum_{k=1}^{N_{TX}} F(e_k^i) \tag{3}$$

where N_{TX} is the expected number of vehicles equipped with the DSRC system, which are in transmission range *R* [3]. If the score is positive (resp. negative), then the OBU makes a positive (resp. negative) decision.

A. Analysis

1) Impact on the decision delay: The decision delay depends on the processing and communication delays and *X*. It is denoted as:

$$Delay_{decision} = X \times (T_{sign} + T_{tx} + T_{verify}) \tag{4}$$

According to [13], T_{sign}, T_{tx}, T_{verify} are the times to sign, to transmit and to verify a message. To simplify the notation, we denote the time overhead as:

$$T_{ov} = T_{sign} + T_{tx} + T_{verify} \tag{5}$$

Without the consensus mechanism, the vehicle decides with only one message. So, the time overhead of the decision method is:

$$T_{decision\_ov} = (X - 1) \times T_{ov} \tag{6}$$

The *Threshold* parameter sets the minimum *X* before making a decision and so induces the decision delay. If the *Threshold* is high, then *X* is high and the decision process is delayed. Before starting the decision process, the vehicle has to wait for $Threshold \times T_{ov}$. If *X* is high, then there is a high delay before having the *X* messages and a high processing overhead (for signature verification). So, a high *X* leads in a higher robustness but a lower reactivity.

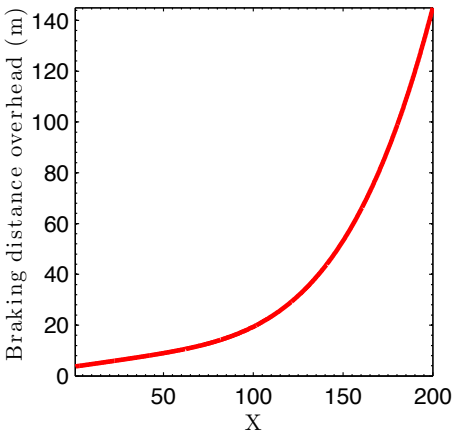

Fig. 4. Impact of X on the braking distance

2) Impact on the braking distance: In the context of a LDW application, the distance is a critical metric. Indeed, depending on the distance between vehicle and hazard, the action will not be the same. The braking distance overhead is defined by:

$$\Delta D_B = D_B^1 - D_B^0 = X \times (v_k T_{ov}) = v_k \times Delay_{decision} \tag{7}$$

where ΔD_B is the braking distance overhead resulting from the authentication process (in meters), D_B^0 is the initial braking distance according to [13], D_B^1 is the total braking distance with security mechanisms.
Fig. 4 shows the impact of *X* on the braking distance for $v_k = 130$ km/h. The exponential shape is due to the fact that higher the number of vehicles (in the communication range) is, higher the probability of message collision is, and higher the transfer delay is. It should be noted that from $X = 180$ (i.e. density $\beta = 100$ veh/km/lane), the braking distance is doubled.

B. Discussion

1) Effect of Threshold: The goal of *Threshold* is to avoid fake attacks by dropping individual attackers and sensor faults. But a limit appears in low-density scenarios. Indeed,

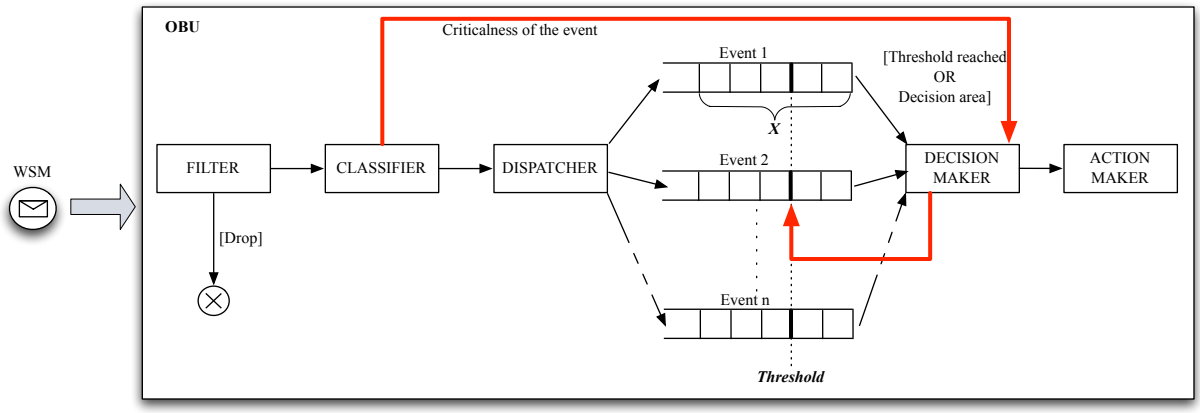

Fig. 5. System architecture for dynamic decision

while the vehicle has not enough messages, it will make a negative decision whereas there is a real danger. This phenomenon is called the *first adaptation phase*.
With a low threshold, a low percentage of collaborative attackers could raise a false decision. Conversely, a high threshold decreases the reactivity of the vehicle. So, the threshold should be adapted to the network density.

2) Effect of X: As the vehicle has to store warnings until the decision process, a high *X* leads in a high storage of messages. So, *X* impacts the size of the event queue. Moreover, we noticed that *X* has an impact on the braking distance. So, *X* should be adapted to the criticalness of the warning and the network density.

V. Dynamic consensus

A. Basic scheme

We propose a dynamic criticalness-based consensus where consensus parameters are based on the current neighborhood density and the criticalness of the event. For example, a vehicle stopped on a lane is more dangerous than a soft brake of a vehicle. Criticalness depends on the danger location, vehicle speed, environmental conditions, traffic density, and type of driver. There are two strategies to make a decision:

i) The more critical the warning is, the less the number of warnings needed is.
ii) Inversely, the more critical the warning is, the more number of warnings needed is.

This choice depends on the property of the target application. If *precaution* is important, the first technique is used because it will make a decision faster. But, if *robustness* is required, then the second technique should be used because as the warning is critical, it should make the right decision.

Fig. 5 shows the components that form our model. There are five components: *filter*, *classifier*, *dispatcher*, *decision maker* and *action maker*. When a vehicle receives a message, the message first goes through the *filter*. To drop useless warning received, the *filter* could check:

- *Distance*:
  $\Delta D = D_{Source} - D_{Receiver} < D^i_{MAX}$. Depending on the type of warning, the maximal distance between source and receiver for event i (D^i_{MAX}) could be different.
- *Lane*: In function of the type and the scope of the warning, the vehicle could drop the alert if the hazard is not on its current lane (in highway scenario). For example, if the hazard is "small" (only concerns one lane of a highway) and $lane_{Source} \neq lane_{Receiver}$ then there is no *vehicle action* (but *network action* is allowed).

The *filter* reduces the number of messages to store and to verify in the same queue.
Then, the WSM goes through the *classifier*, which computes the criticalness $C_{\lambda(i)}$ of the hazard *i* in function of:

- *Distance* ΔD.
- *Speed* of the current vehicle.
- *Heading*: is the vehicle heading to the danger area?
- *Path*: is the vehicle route on the danger area? A vehicle could have a current "safe" heading but the path is expected to cross the danger area soon.

Then, the message goes through the *dispatcher*. The *dispatcher* analyzes the content of the message (event identifier) and sends it to the corresponding queue. Then, a *decision maker* checks every queue to process the decision as soon as the *Threshold* is reached. This module is responsible of setting *X* and *Threshold* according to the trust level and the criticalness computed by the *classifier*. When the decision process is started, the OBU takes from the queue corresponding to the current warning the *X* last messages and compute d_i. The decision could be: negative (do nothing), positive (other action). The decision is sent to the *action maker*, which realizes the action.

B. Analytical modeling

In order to fulfill real-time constraints of the application, we introduce a precaution parameter $\omega \in \{0; 1\}$ (boolean). The precaution parameter depends on the design of the application and follows the precaution principle "an ounce of precaution is worth a pound of cure". It could be dynamically defined in function of $\lambda(i)$. If $\omega = 1$, then the *decision maker* will make

a decision even if X messages are not stored in queue yet. In this case, the decision maker takes all warnings available in the queue. The length of the queue for the event e_k^i is denoted $|Q_{e_k^i}|$. With this parameter, the decision delay becomes:

$$Delay_{decision} = (Threshold + W(X)) \times T_{ov}$$
$$W(X) = \begin{cases} X - Threshold, \text{ if } |Q_{e_k^i}| \geq X \text{ before } T_{MAX} \\ \omega \times (|Q_{e_k^i}| - Threshold), \text{ if } T_{MAX} \text{ elapsed} \end{cases} \quad (8)$$

1) Dynamically change Threshold: In function of the network density, the *decision maker* reduces or increases the *Threshold* parameter. As we mentioned, the goal of the *Threshold* is to avoid decisions from only one message and to allow making a decision quickly. In order to prevent small percentage of attackers to reach false positive decisions, we propose to set the threshold to a certain percentage of the current majority of neighbors ahead. Of course, this parameter could be changed according to the application constraints for example. The dynamic threshold is defined by:

$$Threshold = p \times (Ahead(N_{TX}(t), R))/2 \quad (9)$$

$Ahead(N_{TX}(t), R)$ is a function that returns the number of neighbors, which are moving ahead of the current vehicle at time t in the transmission range R. $p \in [0, 1]$ is the percentage of neighbor ahead.

2) Dynamically change X: The *decision maker* reduces or increases X in function of the network density. X should be proportional to the number of vehicles ahead of the current vehicle. We define X by:

$$X = \frac{(2 \times C_{\lambda(i)})^{pc}}{C_{\lambda(i)}} \times \frac{(Ahead(N_{TX}(t), R))}{2} \quad (10)$$

with $C_{\lambda(i)} \in [1, C_{MAX}]$ the criticalness of the event i. $pc \in [0, 1]$ is an integer which represents a more flexible precaution parameter than ω (which is of type "all or nothing").

To define the criticalness we need to define ΔT which is the time remaining before the collision. When a vehicle receives a warning, it computes ΔT with the following formula:

$$\Delta T_i = T_{collision} - t \quad (11)$$

where t is the current time.
To compute the criticalness $C_{\lambda(i)}$, we use the following formula:

$$C_{\lambda(i)} = \begin{cases} 1, \text{ if } \Delta T_i > T_{safety} \\ 1 + \frac{1}{\Delta T_i}, \text{ if } T_{safety} > \Delta T_i > T_{collision} \end{cases} \quad (12)$$

As long as the vehicle is in the *decision area*, the criticalness is equal to 1. Indeed, the vehicle still has time to collect WSMs. But, as soon as the vehicle enters the *information and reaction area* (i.e. the safety distance), the criticalness is increased to reduce the number of messages needed, and thus, speed up the decision process (to keep a margin between the decision time and the braking time to allow driver notification).

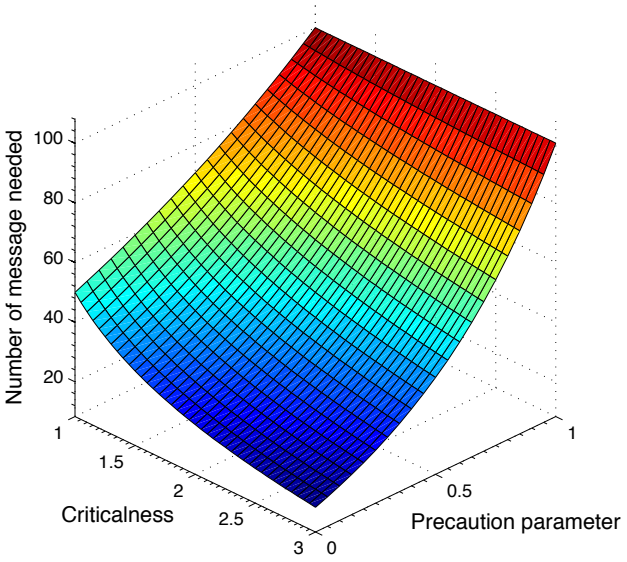

Fig. 6. Impact of criticalness and precaution parameter on X ($N_{TX}(t) = 100$)

C. Analysis and discussion

Fig. 6 shows the impact of the criticalness and the precaution parameter on X when $N_{TX}(t) = 100$. When the criticalness increases and $pc = 0$ (precaution), X decreases. When $pc = (\ln(C_{\lambda(i)}))/(\ln(2 * C_{\lambda(i)}))$, the lower bound of X is $(Ahead(N_{TX}(t), R))/2$ to ensure robustness in scenario with collaborative attackers.

The assessment of X could take into account the real-time application constraint given by x_{MAX} which is the maximum number of messages that could be received before jeopardizing the application (according to T_{MAX}). It could be computed by [13] [3]:

$$\begin{aligned} T_{ov}(M) &= T_{sign}(M) + T_{tx}(Sign_{PrK_V}[M]) + T_{verify}(M) \\ &= (6n+2)T_{MUL} + T_{INV} + 5nT_{SQR} + T_{HASH} \\ &+ \frac{W-1}{2}[\sigma P_e + (T_h + \frac{S_{ov} \times 8}{D_R} + DIFS + \delta)P_S \\ &+ (T_h + \frac{S_{pu} + 2 \times (S_{sigcert} + S_{sigmess}) + 8}{D_R} + EIFS + \delta)P_C] \\ &+ (1-\pi)^{n-1}(1-e) \\ &\times (T_h + \frac{S_{pu} + 2 \times (S_{sigcert} + S_{sigmess}) + 8}{D_R} + DIFS + \delta) \\ &+ (1 - (1-\pi)^{n-1}(1-e)) \\ &\times (T_h + \frac{S_{pu} + 2 \times (S_{sigcert} + S_{sigmess}) + 8}{D_R} + EIFS + \delta) \\ &+ (12n+2)T_{MUL} + T_{INV} + 10nT_{SQR} + T_{HASH} \end{aligned} \quad (13)$$

This assumption introduces x_{MAX} into (10):

$$X = min((\frac{(2 \times C_{\lambda(i)})^{pc}}{C_{\lambda(i)})} \times \frac{(Ahead(N_{TX}(t), R))}{2}, x_{MAX}) \quad (14)$$

But the real-time constraint given by x_{MAX} reduces the security level by decreasing the number of messages needed in high-density scenarios. This constraint cancels the need of $Ahead(N_{TX}(t), R))/2$ messages. Thus, malicious vehicles could inject false warnings. It is well established that a compromise between security and performance needs to be done.

To verify if a consensus is always reached, we analyze the termination. From the computation of the limit of X when N_{TX} goes to $+\infty$, we remark that without the bound in (14), the consensus is never reached. Indeed, as the vehicle approaches the danger location, its neighborhood will grow. So, the vehicle will wait for more and more messages. That is why a decision should be made before the end of the decision area or when the maximum delay allowed by the application is almost exceeded. Thanks to the precaution parameter pc, the time between the first reception and the decision is bounded because the distance is involved in the formula. Compared to an immediate reaction on the first reception, the consensus delays the braking start in order to collect as much messages as possible about the event, and this, maintaining the necessary distance for braking and avoiding the collision.

In terms of performance optimization, a vehicle could verify the message (the signature) as soon as it receives it (*proactive scheme*). Thus, if the threshold is reached, then it has saved $Threshold \times T_{verify}$ processing time to make a decision faster. But if the threshold is never reached, then the vehicle has lost $Threshold \times T_{verify}$ processing time. In function of the precaution parameter the vehicle could verify a message as soon as it receives it, or waits for the threshold to start the verification (*reactive scheme*). The reactive scheme saves *Threshold* verifications but could slow down the decision process. The analysis of this trade-off is an open issue.

VI. Simulation Parameters

We simulate a highway with three lanes in one direction. A percentage of vehicles (20% here) will randomly stop to generate an event. The network simulator ns2.34 generates traces which are used in Matlab to analyze the impact of the consensus parameters. We trace the vehicle that goes through the most important number of events, and analyze the number of messages received for each event. Table I details the simulation parameters.

Parameter	Value
Communication range R (m)	300
Density of vehicle (veh/km/lane)	20
WSM frequency (Hz)	10
Simulation time (sec)	300
Propagation model	Nakagami (m=3)
Propagation delay (ms)	1
Data rate (Mbps)	6
Packet size (bytes)	254
Vehicle speed (m/s)	27.7, 30.5, 36.1
Percentage of attacker	20%
Area	highway 5 km
Number of lanes	3
Percentage of accident	20%
Precaution parameter	0.5

TABLE I
Simulation Parameters

VII. Simulation Results and Analysis

Fig. 7 (resp. Fig. 8 and Fig. 9) shows a trace for a vehicle that receives warnings for the event 406 (resp. 229 and 59). As the vehicle approaches the event location, the number of messages received increases. The bold line (first starting from the right) represents the expected collision time ($T_{collision}$) if the vehicle does not change its state (speed, heading, path). The dashed line (second starting from the right) represents the braking time ($T_{braking}$) before which a decision should be made. The dash-dotted line denotes the safety time (T_{safety}). The dashed line represents the decision time when the vehicle uses (10) to compute X. The dotted line shows the decision time when the majority method is used ($X = \frac{Ahead(N_{TX},R)}{2}$). To compute the dynamic X, $p = 0.2$ and $pc = 0.5$ are used.

Fig. 7 shows a gap of 2 seconds between the majority and the dynamic methods. Therefore, as the vehicle receives more warnings reporting the event, the trust level in the event is increases. The vehicle does not wait for the safety time because (14) introduces an upper bound to respect the real-time constraints of the LDW application of 2.5 seconds.

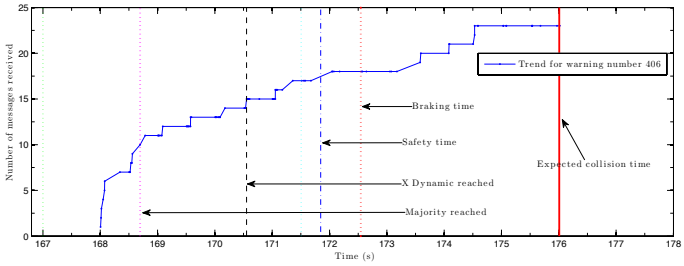

Fig. 7. Warning 406

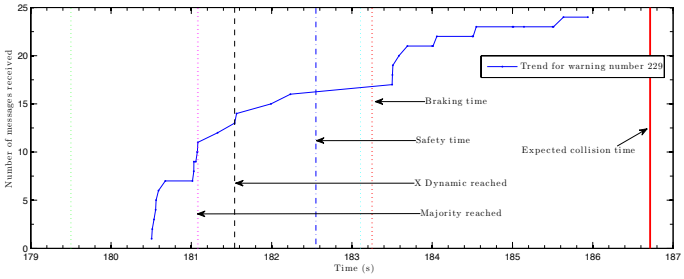

Fig. 8. Warning 229

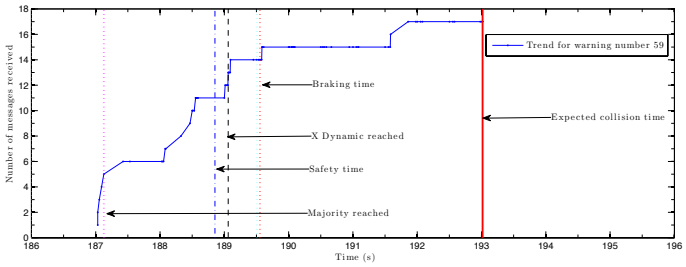

Fig. 9. Warning 59

Fig. 8 shows the case when an event disappears before the vehicle overtakes it (expected collision time line). Fig. 9 illustrates the case when the decision is made after the safety time. Indeed, the vehicle does not have enough messages at the safety time. So, thanks to (12), X is decreased to speed up the decision, and made it before the braking time.
We remark that the dynamic method has a higher decision delay than in the majority method. But the decision is still made before the braking time. We conclude that the dynamic decision method permits to increase the trust into the warning by collecting more messages than in the majority method without jeopardizing the braking distance.

VIII. Conclusion

In this paper, we investigate the consensus mechanism to increase trust in local danger warning application. More especially, we focus on the decision method because it sets the consensus parameter and has an impact on the vehicle reaction. First, we propose a generic model that defines decision methods. As the vehicular network topology changes quickly, we aim at setting the consensus parameter dynamically. So, we propose a decision method that sets X and *Threshold* in function of the network density and the criticalness of the warning. We analyze the impact of these parameters on the decision delay and the braking distance. As of future work, we first intend to optimize the decision delay formula. Indeed, it is assumed that the X warnings are received continuously without considering the background traffic, the competition between warnings (for different events), or the queuing time. Our model should take into account these considerations.

Acknowledgment

The research leading to these results has received funding from the European Union's Seventh Framework Programme project PRESERVE under grant agreement n°269994.

References

[1] B. Ostermaier, F. Dötzer, and M. Strassberger, "Enhancing the security of local danger warnings in vanets - a simulative analysis of voting schemes," *2nd International Conference on Availability, Reliability and Security (ARES'07)*, pp. 422–431, Apr. 2007.

[2] Z. Cao, J. Kong, U. Lee, M. Gerla, and Z. Chen, "Proof-of-relevance: Filtering false data via authentic consensus in vehicle ad-hoc networks," *INFOCOM Workshops 2008, IEEE*, pp. 1–6, 2008.

[3] J. Petit, "Analysis of ecdsa authentication processing in vanets," *3rd international conference on New technologies, mobility and security*, pp. 388–392, 2009.

[4] F. Dötzer, L. Fischer, and P. Magiera, "Vars: a vehicle ad-hoc network reputation system," *6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05)*, pp. 454–456, 2005.

[5] L. Raz, K. Sarit, and S. Yuval, "On the benefits of cheating by self-interested agents in vehicular networks," *6th international joint conference on Autonomous agents and multiagent systems*, pp. 1–8, 2007.

[6] B. Ostermaier, "Analysis and improvement of inter-vehicle communication security by simulation of attacks," *Master's thesis, Technische Universität München*, October 2005.

[7] M. Raya, P. Papadimitratos, V. D. Gligor, and J.-P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks," *27th Conference on Computer Communications (INFOCOM'08)*, pp. 1238–1246, 2008.

[8] L. Nai-Wei and T. Hsiao-Chien, "Illusion attack on vanet applications - a message plausibility problem," *IEEE Globecom Workshops, 2007*, pp. 1–8, 2007.

[9] T. Kosch, "Local danger warning based on vehicle ad-hoc networks: Prototype and simulation," *1st International Workshop on Intelligent Transportation*, pp. 43–47, 2004.

[10] IEEE, "Trial-use standard for wireless access in vehicular environments - security services for applications and management messages," *IEEE Standard 1609.2-2006*, 2006.

[11] J. Mittag, F. Thomas, J. Härri, and H. Hartenstein, "A comparison of single- and multi-hop beaconing in vanets," *6th ACM international workshop on VehiculAr InterNETworking*, pp. 69–78, 2009.

[12] F. Kargl, M. Zhendong, and E. Schoch, "Security engineering for vanets," *4th Workshop on Embedded Security in Cars (ESCAR'06)*, 2006.

[13] J. Petit and Z. Mammeri, "Analysis of authentication overhead in vehicular networks," *3rd Joint IFIP Wireless and Mobile Networking Conference (WMNC'10)*, pp. 1–6, 2010.