An Enhanced Voting Algorithm for Hybrid Jamming Attacks in VANET

Mohamed S. Mohamed
Department of Computer Science
University of Idaho
Moscow, ID 83843
Email: moha3425@vandals.uidaho.edu

Sherif Hussein
Department of Computer Science
University of Idaho
Moscow, ID 83843
Email: huss3426@vandals.uidaho.edu

Axel Krings
Department of Computer Science
University of Idaho
Moscow, ID 83843
Email: krings@uidaho.edu

***Abstract*—DSRC Safety Applications, as part of Intelligent Transportation Systems, are an effective means to reduce road accidents. These applications rely on connected vehicle technologies using wireless communication and thus inherit their security problems. Reliability of safety applications is paramount. Voting-based approaches have been proposed to increase reliability. This research addresses safety application reliability in the presence of jamming, specifically hybrid jamming, which is an unavoidable attack on such technologies. The impact of hybrid jamming on voting-based approaches is investigated, and an Enhanced Voting-based Algorithm is presented, capable of overcoming deficiencies of previous algorithms under this jammer model. Field experiments using commercially available equipment were conducted to measure the performance of the proposed algorithm. Experimental results show that the new algorithm is superior in terms of time required to make decisions and reliability compared to previous work reported in the literature.**

I. Introduction

Intelligent Transportation Systems (ITS) provide technologies, services, and applications that allow Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication. This wireless communication is based on Dedicated Short Range Communications (DSRC) [1]. V2V and V2I communication require that a vehicle be equipped with an On Board Unit (OBU) and the infrastructure, such as a traffic intersection, with a Road Side Unit (RSU). A collection of these devices can form a Vehicular Ad Hoc Network (VANET), which is similar to a Mobile Ad Hoc Network (MANET); however it is designed for quick message exchanges where connections may change rapidly. To access the medium VANET uses the IEEE 802.11p standard [2].

One of the most important applications in VANET are DSRC Safety Applications, which aim to enhance safety by notifying drivers of potential hazards or impending accidents. Examples of such safety applications are Forward Collision Warning (FCW) and Emergency Electronic Brake Lights (EEBL), which alter a driver that a vehicle is braking hard ahead in order to prevent potential collisions.

As safety applications utilize wireless communication, they inherit the full spectrum of security problems associated with the underlying technology. Since ITS is a critical infrastructure where safety application failure could result in injury or death, security is a major concern, as it directly affects application reliability and safety. Any compromise, whether due to a benign fault or malicious attacks attempting to cause safety application to make wrong decisions, may result in the public's loss of confidence in these technologies.

An example of an attack is described in the following scenario. Imagine an attacker launching an object into traffic. The driver seeing the hazard would react, e.g., by braking hard. At the same time assume that the attacker disrupts V2V communication in the region around the hazard by jamming. This inability to communicate would result in the EEBL to fail to alert drivers not having direct visual contact to the hazard, thus potentially leading to rear-end collisions. Whereas such attack may seem contrived, it can be easily executed with all its possible consequences.

This research investigates voting-based agreement as the solution for DSRC safety applications under the effect of jamming. The remainder of this paper will be organized as follows. Section II will provide background information. Related work will be discussed in Section III. The attack model assumed in this research and an Enhanced Voting-based Algorithm (EVA) capable of dealing with the attacks, are presented in Section IV and Section V respectively. The analysis of the performance of the EVA is presented and put into perspective in Section VI. Finally, Section VII concludes the paper.

II. Background

Background information of the key technologies used in this research will be presented next.

A. DSRC and Basic Safety Message (BSM)

DSRC provides V2V and V2I communication. The Federal Communications Commission (FCC) has licensed the use of 75 MHz of bandwidth at 5.9 GHz (5.850-5.925 GHz) for DSRC services. This bandwidth is divided into seven channels, each having 10 MHz of bandwidth [3]. The seven channels are composed of one Control Channel (CCH) (denoted by CH 178), and six Service Channel (SCH) (denoted by CH 172, 174, 176, 180, 182, and 184). The most important channel is Safety Channel CH172, dedicated to DSRC safety applications. The remaining 5 MHz is for future use.

The Basic Safety Message (BSM) is the most important message for safety applications. It is periodically broadcast every 100 ms on Safety Channel CH 172 to exchange information about the status of the vehicle. A BSM consists of two parts. The first part is mandatory and contains data included in every BSM, such as speed, GPS location, heading, acceleration and brake status information [4]. The second part is optional and includes additional information for certain applications.

B. DSRC Safety Applications

Various types of DSRC Safety Applications, such as FCW and EEBL, have been presented in [5]. Alerts from these safety applications enable the driver to react to dangerous situations. The safety applications use information contained in BSMs received from surrounding vehicles. In this research we considered the EEBL application, which alerts the driver of the Host Vehicle (HV) of an impending rear-end collision with a Remote Vehicle (RV) driving ahead in the same lane and direction. EEBL is useful in situations of low visibility, e.g., due to fog or drifting snow, or when other vehicles block the view of the HV. The timing model related to EEBL is shown in Figure 1. Upon recognition of a hazard the driver

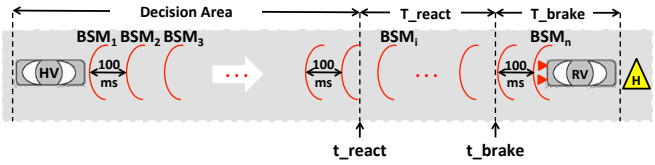

Fig. 1. EEBL timing model

of the RV is assumed to brake hard at time t_{brake}. During the time interval T_{brake} the RV broadcasts this braking event e in its BSMs to the surrounding vehicles. Note that we use lower case t to denote instances of time and upper case T for time intervals. The EEBL application running on the HV uses the received BSMs indicating event e to alert the driver of the HV. This alert has to be issued early enough to allow the driver to react, i.e., no later than t_{react}. Typical reaction times T_{react} have been recorded within 0.9 to 1.2 seconds [6].

The reliability of EEBL is conditioned on the reception of these BSMs and making the correct decision in the proper time. The application fails if no BSM was received, thereby leaving the HV unaware of the occurrence of the event. To verify that a reported event e exists indeed, and is not the result of an attack, BSMs from vehicles close to the event may be checked for consistency in event reporting. Such voting-based applications fail if not enough BSMs are received to come to a correct decision, as will be discussed in Section III.

C. CSMA/CA protocol of the 802.11p

The access rules to the medium are defined in standard IEEE 802.11p [2], which was proposed to be used in rapidly changing environments where very short communication durations are required. The standard employs the Enhanced Distributed Channel Access (EDCA) contention-based channel access as the MAC layer protocol. EDCA utilizes Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). When a node wants to send a packet, it senses the medium first, and if it is free for an Arbitration Interframe Space (AIFS), the node selects a random backoff time to delay the transmission.

The backoff procedure functions as follows:

(i) The node selects a random backoff time uniformly from the Contention Window CW defined as $[0, CW + 1]$, where the initial CW is equal to a predetermined CW_{min}.

(ii) The CW value increases, i.e., it doubles, if the subsequent transmission attempt fails, until it reaches a predetermined CW_{max}.

(iii) The backoff value decreases only when the node senses the medium as free.

(iv) When the backoff value reaches 0, the node will send the packet immediately.

EDCA utilizes four different access categories (ACs) with different priorities in order to ensure that high priority messages will be exchanged timely and reliably, even in dense traffic scenarios. These four traffic categories are: Background traffic (BK or AC0), Best Effort traffic (BE or AC1), Video traffic (VI or AC2) and Voice traffic (VO or AC3). Table I shows different AIFS numbers and CW values chosen for each of these four categories. The first column contains the four ACs ordered from lowest to highest priority. The second and the third columns show the corresponding minimum and maximum contention windows. Finally, the last column holds the Arbitration Interframe Space Number (AIFSN) used.

TABLE I
PARAMETER SETTINGS FOR DIFFERENT ACCESS CLASSES IN IEEE 802.11P [2]

AC	CW_{min}	CW_{max}	AIFSN
BK (AC0)	CW_{min}	CW_{max}	9
BE (AC1)	CW_{min}	CW_{max}	6
VI (AC2)	(CW_{min})/2-1	CW_{min}	3
VO (AC3)	(CW_{min})/4-1	(CW_{min})/2-1	2

Any node violating the policies of the inter-frame spacing could gain disproportional access to the medium. Such node is considered to be *misbehaving*. The impact of misbehaving nodes has been addressed in research such as [7], [8], [9], [10] and the misbehavior detection of [7] will be used later in Section V.

D. Jammer Types

The main attack model for this research is jamming, which can be defined as an act of transmitting radio signals in order to interfere with communication and/or block legitimate nodes from accessing the medium. The goal of jammers may thus be twofold, 1) to interfere with wireless communication to decrease the Signal to Noise Ratio (SNR), thus making reception unreliable or impossible and potentially destroy network packets, or 2) prevent nodes from gaining access to the medium. Jammers may have different properties, including ease of detectability, power usage, sophistication with respect

to protocol awareness and level of Denial of Service (DoS) [11], [12]. Jamming models can be divided into two main categories: 1) simple jamming models and 2) intelligent jamming models.

1) Simple Jamming Models: Simple jammers were discussed in detail in [11]. The first jamming model in this category is the *Constant Jammer*, which emits a constant stream of random data that does not follow the MAC layer protocol. As a result, the medium appears to be busy, thus blocking legitimate nodes from access. However, it may also result in corruption of ongoing packets. The *Deceptive Jammer* is the second jamming model. This type of jammer does not follow the channel access protocol by continually injecting a stream of what appears to be valid packets without any gaps between them. The third jamming model, *Random Jammer*, switches randomly between periods of jamming and sleeping. During the jamming period its behavior resembles that of a constant jammer. *Reactive Jammer* is the last jamming model in this category. It senses the medium for ongoing communication, and when it senses a packet transmission it emits a radio signal that collides with the packet, thus corrupting it.

2) Intelligent Jamming Models: This type of jammer interferes with communication between nodes, thereby corrupting packets sent by legitimate nodes. It is also called protocol-aware jammer, and has the capability of corrupting specific packets. It may target control packets, such as *RTS, CTS* or *ACK*, but could also target *DATA* packets, as described in [12].

From this research point of view, all jamming models presented above can be classified into two different models: destructive and non-destructive jamming models. Destructive jammers interfere with communication between nodes, thus they may destroy messages sent by legitimate nodes during the jamming period. A non-destructive jammer blocks legitimate nodes from accessing the medium, thereby forcing these nodes to queue messages in their transmission queues. After jamming ends, the nodes will be able to flush their queued messages to the medium following the CSMA protocol. It should be noted that transmission queues may overflow, causing either the newest or oldest packets to be discarded [13].

An example of a truly non-destructive jammer is the *Hybrid Jamming* model described in [7]. It combines properties of constant, deceptive, and random jammers. The hybrid jammer sends continuous random bits like a constant jammer, however they appear as bursts of regular packets, without following the channel access protocol, like a deceptive jammer. The hybrid jammer is the specific jamming model selected for this research.

Even a non-destructive jammer may cause destruction of packets. For example, it may corrupt an ongoing transmission when it starts up. Furthermore, when its signal diminishes, for example for vehicles further away from the jammer, its impact on these nodes may degrade to simple reduction of the SNR.

III. Related Work

The reliability of safety applications is paramount. For safety applications subjected to malicious act, such as attacks aiming to confuse applications in order to derive incorrect decisions, special mechanisms are needed. Research has addressed this by using redundant BSMs received from nearby vehicles capable of witnessing an event. These vehicles are said to be located in the *detection zone* [15]. Upon detection of an event, each HV starts collecting the BSMs received from vehicles in the detection zone to construct a voting set. Voting-based solutions have been presented in the literature [14], [15], [16]. These approaches differ in the way a voting set is constructed, e.g., based on the freshness of messages, and the size of the voting set. The latter is called *voting threshold*. The final decision is based on this threshold, e.g., by applying majority voting.

The challenge is how to select the correct threshold in a tradeoff space between speed and robustness of the voting decision. Selecting a low threshold allows decisions to be made fast, however, it may increase the probability of making wrong decisions. On the other hand, selecting a high threshold makes robust decisions, but results in higher latency. Different strategies have been used to define the threshold as *static* or *dynamic* [17], [15]. A static threshold is set a priori, e.g., during the manufacturing of the vehicle, while dynamic thresholds change based on neighborhood density and criticality of the event.

Algorithms based on voting can be classified into two categories. The first category consists of voting algorithms using new message architectures based on authentic consensus, namely authentication and verification, of each vehicle. The second category consists of voting algorithms relying on configuring the voting set based on factors like message freshness and thresholds.

An example of the first category described in [17] considers a Proof-of-Relevance (PoR), which is generated by vehicles collecting digital endorsements from other witnesses of an event. Its scheme consists of three phases: 1) Report generation that includes location, type and time of an event, 2) Signature collection, which is the key procedure in this scheme. In this phase all vehicles that detected the event will participate in the signature collection protocol until enough signatures are collected. 3) Report verification, where each vehicle that received the event report will examine whether there are enough signatures or not. If there are enough signatures, each vehicle will start validating signatures to check for incorrect signatures. Once enough correct signatures are observed, a decision is made. However, such an approach requires additional communication, thereby adding overhead, but more importantly, they require a modification of the standards.

For the aforementioned reasons we will consider approaches of the second category using voting set configuration as discussed in [14], [15], [18], [16]. In [14] the authors proposed four static decision methods, which are based on voting algorithms that use plausibility checks in order to take the

correct decision in the presence of value faults. These decision methods are: *Freshest Message*, which only consider the most recent messages, *Majority Wins*, which execute local voting over all distinct messages, *Majority of Freshest X*, which combines the previous two methods considering only the recent X distinct messages, and *Majority of Freshest X with Threshold*, which is simply an extension of the previous method in addition to checking if the distinct messages received so far exceed a certain threshold or not. However, their work did not specifically state the *time to live* for messages and does not explicitly state a way for the calculation of the thresholds.

In [15] the authors proposed a dynamic criticality threshold based on the *Majority of Freshest X with Threshold* scheme of [14], where consensus parameters and threshold are depending on neighborhood vehicle density and criticality of the event. There are two strategies for making a decision. As mentioned before, for critical events a compromise space exists between fast and robust decisions. The more critical the event, the fewer messages should be needed for fast decisions. However robust decisions require more messages.

The authors in [18] proposed an adaptive decision making method in order to improve the accuracy and time efficiency of decision-making. It aims to take a decision as soon as possible once the amount of received opinions are greater than a threshold or when the time delay between the first received message and the current received message exceed a maximum delay.

An adaptive threshold algorithm was proposed in [16] that considered the Majority of Freshest X with dynamic Threshold [15] for an adaptive threshold algorithm that provided higher resilience against certain types of jamming.

None of the previous research is suitable for dealing with the hybrid jammer used in this research, as message queuing affects the freshness of messages.

IV. Attack Model

A scenario involving vehicles reacting to a hazard in the absence of an attack is shown in Figure 2. Vehicles RV1 and RV2 observing a hazard react, causing their BSMs to indicate a braking event. Vehicle HV is assumed to not have visual contact and thus its safety application relies on messages from the RVs, which both consistently indicate the event.

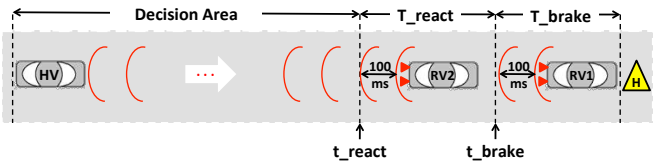

Fig. 2. No attack

Next, consider the scenario depicted in Figure 3, where a hybrid jammer is positioned on the roadside next to RV1 and RV2. This jammer A1 jams for a period of T_{jam} in coordination with the creation of the hazard in front of the RVs. At the same time a collaborating attacker vehicle A2

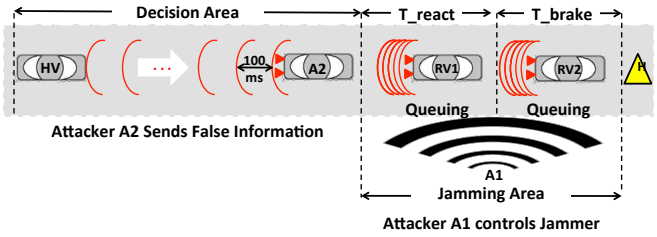

Fig. 3. Attack causing message queuing

starts sending false BSMs indicating no event to the HV. In the absence of jamming, the false values of A2 would be outvoted by the correct values of RV1 and RV2 in the voting algorithm executing in the HV.

In the presence of jamming the above behavior changes. Specifically, the hybrid jammer will force the RVs to queue their BSMs during jamming period T_{jam}. The attacker A2, positioned outside of the jamming area, will be able to stack the voting set of the HV with false values. The HV will have to make a decision to notify the driver of the event before it is too late to react. This time is approximated by

$$t_{safety} = t_{now} + T_{safety}$$

where t_{now} is the current time and

$$T_{safety} = \frac{loc_{HV} - origloc_{RV1}}{speed_{HV}} - T_{react}$$

where $speed_{HV}$ is the current speed of the HV, and loc_{HV} and $origloc_{RV1}$ are the current and original locations of the HV and RV1 respectively.

In line with the standard definition of reliability, i.e., $R(t)$ is the probability that the system is working to specifications during the entire time interval [0, t] [19], we can define the EEBL application reliability as the probability of the algorithm taking a correct decision at or before t_{safety}.

V. Enhanced Voting Algorithm (EVA)

The EVA is built on an architecture consisting of multiple components. A core component is the *dispatcher*, which is similar to the dispatcher used in the system model of [15]. It starts a separate thread for each distinct event e_j observed, i.e., upon the first occurrence of e_j reported in a BSM by some vehicle. The dispatcher forwards BSMs to the threads corresponding to events e_j if the RV sending the BSM is located in the event detection zone. In [15] this zone is determined by a so-called filter, which uses metrics such as distance from, and lane of an event.

The EVA, which is running in the thread associated with each e_j, is shown in Figure 4. The algorithm consists of two stages: an investigation and a voting stage.

Investigation: This stage deals with hybrid jamming attack and/or misbehavior detection. The EVA calls the Hybrid Jammer Detection Algorithm (HJDA) described in [7], which identifies if an RV is a victim of a hybrid jammer attack or a misbehaving node. For each vehicle the HJDA saves

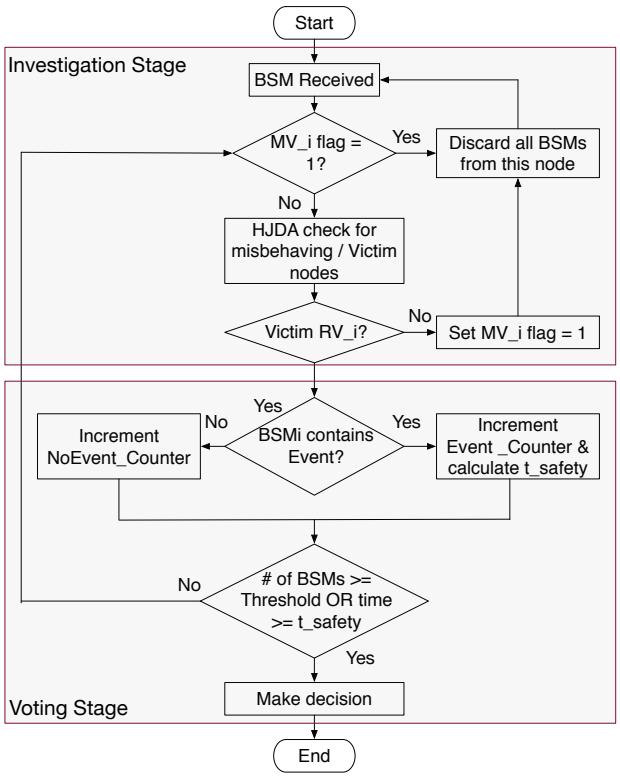

Fig. 4. Enhanced Voting Algorithm for Event e_j

the last BSM received and keeps track of the number of BSM omissions. This can be done using watchdog timers, since a BSM is expected from each vehicle approximately every 100ms. The HJDA uses two metics. The first is the difference between time stamps of the creation of the saved BSM and the currently received BSM. Such time stamp is included in a BSM field called *Dsecond*. If this difference is significantly less than 100ms, this vehicle is considers to be misbehaving. The second metric is the difference between the time stamps of the HV and the recently received BSM. This metric allows allows identifying the number of missing BSMs. If this number is approximately equal to the number of BSM omissions identified then this vehicle is a victim. This procedure is explained in detail in [7].

The HJDA controls Misbehaving Vehicle flags, MV_i, one entry for each surrounding RV. The MV_i flags are initially cleared when EVA is invoked and are used to differentiate between misbehaving and legitimate RVs. When a BSM is received and its corresponding $MV_i = 1$, indicating misbehaving, the BSM is discarded. Otherwise, the HJDA is called to check whether RV_i is misbehaving or if it is a victim of hybrid jamming.

Voting: If RV_i passes the investigation stage, its value is added to the voting set. It should be noted that, due to the timing checks in the HJDA of the investigation stage, this not only includes BSMs with regular 100ms transmission rates, but also those from victim RVs flushing their messages queued during jamming. These latter BSMs will be discarded in the research identified in Section III, if they are considered to be outdated.

Recall that t_{safety} is the critical time by which the HV needs to take action. The value for t_{safety} is updated for each BSM indicating the event. A final decision using voting is taken if 1) t_{safety} is reached or 2) if the threshold on the cardinality of the voting set is reached. Due to the investigation stage, the EVA bases its voting decisions only on values coming from non-misbehaving and victim vehicles.

VI. Performance Analysis

In this section we will investigate the performance of previous voting algorithms subjected to the hybrid jammer. Then the performance enhancements of the EVA will be presented, but first the field test assumptions are introduced.

A. Experiment set-up

Whereas the theoretical impact of the hybrid jammer on the vehicles is clear, as it is defined directly by the attack model and its associated queuing behavior described in Section IV, the impact of the jammer in the field needed to be investigated. Therefore, in order to validate the EVA, field experiments were conducted and the results were collected and analyzed. The experimental setup consisted of the scenario compatible with that shown in Figure 3. Specifically, vehicles representing one HV, two RVs and attacker A2, were equipped with LocoMate Classic OBUs from Arada Systems [20]. All OBUs of vehicles used the standard transmission rate of 10 BSMs per second and a transmission power of 23 dBm using Safety Channel CH172. The RVs were configured to send BSMs announcing that an event occurred. On the other hand, the attacker A2, located in the detection zone, sent BSMs falsely indicating no event. The OBU in the HV executed the EVA. An additional OBU was configured to act as a hybrid jammer capable of operating with different transmission powers, data rates and jamming periods.

The experiments were conducted in a controlled configuration, where the vehicles were positioned as in Figure 3, however they were stationary. This configuration mimics the worst case real scenario, in which the jammer can control the RVs precisely, without affecting attacker A2. The jammer was placed directly next to the RVs and produced 1 dBm of jamming power for a duration of 1.5, 2 and 4.5 seconds. The attacker A2 and the HV were placed outside of the jamming area, 80 meters from the RVs. The reason for conducting the experiment stationary rather than on moving vehicles was multifold. During many hours of field testing it proved to be very difficult to maintain constant speeds and thus distances between all vehicles. Furthermore, we needed to shield the experiment from the impact of the road geometry, such as curves and elevation changes, as well as that of unrelated road traffic, in the absence of a dedicated test site. A summary of the experiment parameters is given in Table II.

B. Previous voting approaches

Figures 5 and 6 show several graphs, each of which represents certain contributions to the voting set with respect

TABLE II
FIELD EXPERIMENT CONFIGURATION PARAMETERS

OBU Model	Arada Systems LocoMate Classic
Number of OBUs	5 (4 OBUs in four vehicles, 1 OBU as stationary jammer)
Test range	straight one-lane road
Jammer position	2 m from the RVs
Distance: HV to RV1	80 m
Vehicles speed	0 m/s (Fixed)
BSM generation	10 BSM/s
Channel	Safety Channel 172
Transmitter power	23 dBm
Data rate	3Mbps
Jammer power & data rate	1 dBm, 3Mbps

to different jamming periods. In the absence of attacks, the

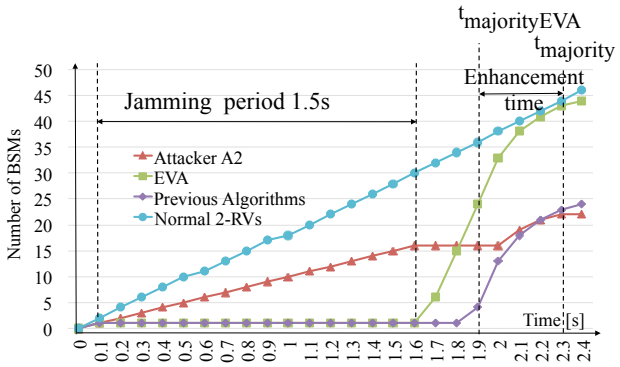

Fig. 5. Performance: 1.5 second jamming with Arada OBU

messages contributed to the voting set by RV1 and RV2 are shown in graph *Normal 2 RVs* in the figures. With no message losses this graph would be linear. However, during the specific field test represented in the graph, three BSMs were lost.

Next we consider the impact of attacks on the voting set using the attack scenario shown in Figure 3. Note that attacker A2 is not affected by jamming, whereas RV1 and RV2 are in the jamming zone. The measured contribution of attacker A2, who is sending false data, is shown in graph *Attacker A2*.

During different jamming periods no BSMs from RV1 and RV2 were received by the HV and only BSMs from A2 are visible in the voting set. Recall that the voting algorithm is running on the HV. However, due to the omission of BSMs from RV1 and RV2 as the result of message queuing, the threshold of the voting set of previous algorithms, e.g., [15], [16], cannot be reached yet to make a decision.

Once the jamming period ends, BSMs queued in the RVs are flushed, as described in Subsection II-D. Recall that with a BSM transmission rate of 10Hz and jamming periods of 1.5 and 2 seconds, approximately 15 and 20 BSMs were queued by each RV respectively. According to [21] the time-to-live of a BSM should be no more than 500ms. Older messages are considered to be outdated. This implies that the voting algorithm running on the HV will consider only the most recent 5 BSMs of the queued BSMs in each RV. The contributions of RV1 and RV2 to the voting set of algorithms using this real-time constraint is depicted in graph *Previous Algorithms* in both figures.

As stated before, a voting decision needs to be made either once the voting set cardinality threshold is reached, or no later than time t_{safety}. A voting algorithm can make a correct decision only once the voting set contains a majority of correct BSMs. This point is reached in Figure 5 and 6 at time $t_{majority}$. If $t_{safety} < t_{majority}$ the voting algorithm will come to the wrong decision, otherwise a correct decision is made.

C. Performance evaluation of the Enhanced Voting Algorithm

One of the key issues of voting is the fact that messages may arrive in bursts due to jamming and that outdated messages could be discarded (as argued in [21]). Recall that if algorithms consider time, then RVs will be considered as misbehaving nodes during bursts. Consequently their BSMs would be discarded, even if they indicate an event. If time is not considered, then the voting set could be highly affected by misbehaving nodes, as they would disproportionally stack the set. The EVA has the capability to resolve these conflicts.

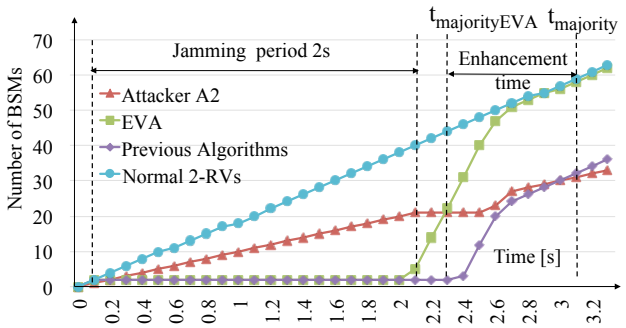

Fig. 6. Performance: 2 second jamming with Arada OBU

The graph *EVA* in Figures 5 and 6 shows the performance of the new algorithm. During the jamming periods no BSMs from the RVs are added to the voting set. However, once jamming stops, the bursts of queued BSMs from RVs are added to the voting set, as they arrive. The EVA can make a correct decision once majority is reached at time $t_{majorityEVA}$. The figures show the difference in time the EVA and the previous algorithms reach their corresponding $t_{majority}$, indicated by *Enhancement time*. As can be seen, the EVA outperformed other voting algorithms by 0.4s and 0.9s for the 1.5s and 2s jamming periods respectively. For safety applications such improvement could have significant impact, for example the 0.9s enhancement is approximately equal to typical reaction times. From a safety application reliability point of view a reliable decision can be made by the EVA at time $t_{majorityEVA}$, whereas the previous algorithms will make a wrong decision during the half-closed time interval $[t_{majorityEVA}, t_{majority})$.

The performance of the EVA in Figures 5 and 6 was derived with data captured from Arada LocoMate OBUs, which were

evaluated to have a queue size of 40 [7]. An attacker knowing the OBU queue size could force a worst case behavior by causing the queue to overflow. Such scenario is shown in Figure 7. Here the jamming period of 4.5 seconds is sufficiently long to queue 40 messages and drop 5. Recall that during 4 seconds 40 BSMs are sent. The previous algorithms will discard all queued messages. Together with the 5 messages dropped due to the *newest packet dropped* behavior [13], it will have no messages to consider. The EVA on the other hand will consider all queued messages, leading to the largest enhancement time of 3.3 seconds, as can be seen in the figure.

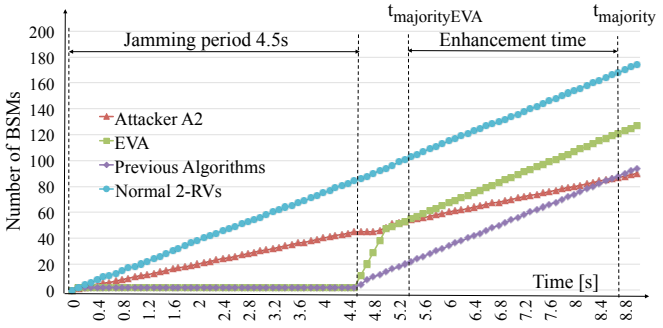

Fig. 7. Performance: 4.5 second jamming with Arada OBU

Running the EVA, which implies also the executions of the HJDA algorithm, imposes computation overhead on the OBUs. This overhead is constant and negligible for each BSM in HJDA and the EVA, with respect to updating data structures. The delay associated with achieving the threshold in Figure 4 is dependent on messages received from vehicles in the detection zone. However, this time is bound by the T_{safety}, as described in Section IV.

VII. Conclusions

This paper presented a new Enhanced Voting-based Algorithm to improve reliability of DSRC Safety Applications operating in hostile environments. A hybrid jammer, potentially causing safety applications to fail, was considered. This jammer type is not only capable of forcing nodes to queue messages, but also making these legitimate nodes appear as misbehaving. Field experiments based on an attack model were conducted to demonstrate the impact of the hybrid jammer on forced queueing in specific commercially available OBUs. The EEBL safety application was used as an example during experiments. The results observed showed that the EVA was capable of significantly reducing the application's decision times, thereby improving application reliability. In worst case scenarios, which an intelligent attacker could provoke, the improvements of the EVA were significant. During experiments enhancements of up to 3.3 seconds were observed. In the context of safety critical applications, such improvements could have significant impact on avoiding accidents and saving lives.

References

[1] Kenney, J. B., *Dedicated short-range communications (DSRC) standards in the United States*, Proc. of the IEEE, vol. 99, no. 7, pp. 1162-1182, 2011.

[2] *IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments*, IEEE Std 802.11p, 2010.

[3] *Amendment of the Commission's Rules Regarding Dedicated Short-Range Communication Services in the 5.850-5.925 GHz Band (5.9 GHz Band)*, Federal Communications Commission FCC 03-324, 2004.

[4] *Dedicated Short Range Communications (DSRC) Message Set Dictionary*, Society of Automotive Engineers, SAE J2735, November 2009.

[5] *Vehicle safety communications-applications (VSC-A) final report*, DOT HS 811 492 A. U.S. Department of Transportation, NHTSA. September 2011.

[6] G. Johansson, K. Rumar, *Drivers brake reaction times - Human Factors*, The Journal of the Human Factors and Ergonomics Society 13, no. 1, pp. 23-27, 1971.

[7] S. Hussein, M. S. Mohamed and Axel Kring, *A New Hybrid Jammer and its Impact on DSRC Safety Application Reliability*, The 7th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, pp. 1-7, 2016.

[8] M. Raya, J. P. Hubaux, and I. Aad, *DOMINO: Detecting MAC Layer Greedy Behavior in IEEE 802.11 Hotspots*, IEEE Trans. Mobile Computing, 2006.

[9] M. N. Mejri and J. Ben-Othman, *Detecting greedy behavior by linear regression and watchdog in vehicular ad hoc networks*, Global Communications Conference (GLOBECOM), pp. 5032-5037, 2014.

[10] L. Toledo, and X. Wang, *Robust detection of MAC layer denial-of-service attacks in CSMA/CA wireless networks*, IEEE Trans. Inf. Forensics Security, vol. 3, no. 3, pp. 347-358, Sep. 2008.

[11] Xu W, Trappe W, Zhang Y, Wood T, *The feasibility of launching and detecting jamming attacks in wireless networks* In: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp 46-57, 2005.

[12] K. Pelechrinis, M. Iliofotou, S.V. Krishnamurthy, *Denial of Service Attacks in Wireless Networks: The Case of Jammers*, Communications Surveys & Tutorials, IEEE , Vol.13, No.2, pp.245-257, 2011.

[13] L. Hendriks, *Effects of Transmission Queue Size, Buffer and Scheduling Mechanisms on the IEEE 802.11p Beaconing Performance*, 15th Twente Student Conference on IT, June 20th, Enschede, The Netherlands, 2011.

[14] B. Ostermaier, F. Dotzer, and M.Strassberger, *Enhancing the security of local danger warnings in vanets - a simulative analysis of voting schemes*, in Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on. IEEE, pp. 422431, 2007.

[15] J. Petit and Z. Mammeri, *Dynamic consensus for secured vehicular ad hoc networks*, in Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE 7th International Conference on. IEEE, 2011, pp. 18, 2011.

[16] Hani Alturkostani and Axel Krings, *The Impact of Jamming on Threshold-Based Agreement in VANET*, International Conference on Connected Vehicles and Expo (ICCVE), 2014.

[17] 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*, in INFOCOM Workshops, IEEE. IEEE, 2008, pp. 16, 2008.

[18] Yu-Chih Wei and Yi-Ming Chen, *Adaptive Decision Making for Improving Trust Establishment in VANET*, IEICE - Asia-Pacific Network Operation and Management Symposium (APNOMS), 2014.

[19] W. B. Johnson, *Design and Analysis of Fault-Tolerant Digital Systems*, Addison-Wesley Publishing Company, New York, 1989.

[20] Arada Systems, *www.aradasystems.com*

[21] X. Ma, X. Yin, and K.S. Trivedi, *On the Reliability of Safety Applications in VANETs*, Invited paper, International Journal of Performability Engineering Special Issue on Dependability of Wireless Systems and Networks, 8(2), March 2012.