A FRAMEWORK FOR INCIDENT DETECTION AND NOTIFICATION IN VEHICULAR AD-HOC NETWORKS

by

Mahmoud Abuelela
B.S. June 1999, Alexandria University, Egypt
M.S. June 2005, Alexandria University, Egypt

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
May 2011

Approved by:

_____
Stephan Olariu (Director)

_____
Hussien Abdel-Wahab

_____
Ravi Mukkamala

_____
Michele C. Weigle

_____
Mecit Cetin

ABSTRACT

A FRAMEWORK FOR INCIDENT DETECTION AND NOTIFICATION IN VEHICULAR AD-HOC NETWORKS

Mahmoud Abuelela
Old Dominion University, 2011
Director: Dr. Stephan Olariu

The US Department of Transportation (US-DOT) estimates that over half of all congestion events are caused by highway incidents rather than by rush-hour traffic in big cities. The US-DOT also notes that in a single year, congested highways due to traffic incidents cost over $75 billion in lost worker productivity and over 8.4 billion gallons of fuel. Further, the National Highway Traffic Safety Administration (NHTSA) indicates that congested roads are one of the leading causes of traffic accidents, and in 2005 an average of 119 persons died each day in motor vehicle accidents.

Recently, Vehicular Ad-hoc Networks (VANET) employing a combination of Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) wireless communication have been proposed to alert drivers to traffic events including accidents, lane closures, slowdowns, and other traffic-safety issues.

In this thesis, we propose a novel framework for incident detection and notification dissemination in VANETs. This framework consists of three main components: a system architecture, a traffic incident detection engine and a notification dissemination mechanism. The basic idea of our framework is to collect and aggregate traffic-related data from passing cars and to use the aggregated information to detect traffic anomalies. Finally, the suitably filtered aggregated information is disseminated to alert drivers about traffic delays and incidents.

The first contribution of this thesis is an architecture for the notification of traffic incidents, NOTICE for short. In NOTICE, sensor belts are embedded in the road at regular intervals, every mile or so. Each belt consists of a collection of pressure sensors, a simple aggregation and fusion engine, and a few small transceivers. The pressure sensors in each belt allow every message to be associated with a physical vehicle passing over that belt. Thus, no one vehicle can pretend to be multiple vehicles and there is no need for an ID to be assigned to vehicles.

Vehicles in NOTICE are fitted with a tamper-resistant Event Data Recorder

(EDR), very much like the well-known black-boxes onboard commercial aircraft. EDRs are responsible for storing vehicles behavior between belts such as acceleration, deceleration and lane changes. Importantly, drivers can provide input to the EDR, using a simple menu, either through a dashboard console or through verbal input.

The second contribution of this thesis is to develop incident detection techniques that use the information provided by cars in detecting possible incidents and traffic anomalies using intelligent inference techniques. For this purpose, We developed deterministic and probabilistic techniques to detect both blocking incidents, accidents for examples, as well as non-blocking ones such as potholes. To the best of our knowledge, our probabilistic technique is the first VANET based automatic incident detection technique that is capable of detecting both blocking and non blocking incidents.

Our third contribution is to provide an analysis for vehicular traffic proving that VANETs tend to be disconnected in many highway scenarios, consisting of a collection of disjoint clusters. We also provide an analytical way to compute the expected cluster size and we show that clusters are quite stable over time. To the best of our knowledge, we are the first in the VANET community to prove analytically that disconnection is the norm rather than the exceptions in VANETs.

Our fourth contribution is to develop data dissemination techniques specifically adapted to VANETs. With VANETs disconnection in mind, we developed data dissemination approaches that efficiently propagate messages between cars and belts on the road. We proposed two data dissemination techniques, one for divided roads and another one for undivided roads. We also proposed a probabilistic technique used by belts to determine how far should an incident notification be sent to alert approaching drivers.

Our fifth contribution is to propose a security technique to avoid possible attacks from malicious drivers as well as preserving driver's privacy in data dissemination and notification delivery in NOTICE. We also proposed a belt clustering scheme to reduce the probability of having a black-hole in the message dissemination while reducing also the operational burden if a belt is compromised.

ACKNOWLEDGMENTS

This work could not be completed on its current state without the help of many individuals to whom I would like to express my appreciation. First and foremost, I would like to thank my advisor, Prof. Stephan Olariu who has taught me a lot in developing my research ability. I remember that I was meeting with Prof. Olariu almost every day for years and he always had his office open for me to help me resolve research, technical and even personal issues during the program. I cannot really find enough words to express my appreciation for his help and support during my PhD.

Next, I would like to convey my sincere thanks to Prof. Hussien Abdel-wahab who has helped me a lot getting started when I first joined the program. I would like also to thank other members of my PhD committee Prof. Ravi Mukkamala, Dr. Michele C. Weigle, and Dr. Mecit Cetin. Their expertise, thorough reviewing, continuous support, and valuable suggestions have led to a greatly improved dissertation.

Finally, I am grateful to my family for their encouragement and support.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The US Department of Transportation (US-DOT) estimates that over half of all congestion events are caused by highway incidents rather than by rush-hour traffic in big cities [1]. The US-DOT also notes that in a single year, congested highways due to traffic incidents cost over $75 billion in lost worker productivity and over 8.4 billion gallons of fuel. Further, the National Highway Traffic Safety Administration (NHTSA) indicates that congested roads are one of the leading causes of traffic accidents, and in 2005 an average of 119 persons died each day in motor vehicle accidents [2].

Given sufficient advance notification of traffic incidents, drivers could make educated decisions about taking alternate routes. This would improve overall traffic safety by reducing the severity of congestion while saving both time and fuel in the process. On most US highways, congestion is a daily event and advance notification of imminent congestion is unavailable [2].

Recently, Vehicular Ad-hoc Networks (VANETs) employing a combination of Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) wireless communication have been proposed to alert drivers to traffic events including accidents, lane closures, slowdowns, and other traffic-safety issues. In the US, VANETs are using 75MHz of spectrum in the 5.850 to 5.925 GHz band specially allocated by the US Federal Communications Commission (FCC) for Dedicated Short Range Communications (DSRC) [3].

In spite of their close resemblance to Mobile Ad Hoc Networks (MANETs), with which they share the same underlying philosophy, VANETs have a number of specific characteristics that set them apart from MANETs. First, while most MANETs are deployed in support of special-purpose operations including disaster relief, search-and-rescue, law-enforcement and multimedia classrooms, all of which are intrinsically short-lived and involve a small number of nodes, VANETs may involve thousands of fast-moving vehicles over hundreds of miles of roadways and streets. Second, and perhaps more importantly, while MANETs may experience *transient* periods of loss of connectivity, in VANETs, especially under sparse traffic conditions, extended periods of disconnection are the norm rather than the exception.

This dissertation follows the style of *The IEEE Transactions*

In most of the systems, developed for incident notifications in VANETs, individual vehicles are responsible for inferring the presence of an incident based on reports from other vehicles. This invites a host of serious and well-documented security attacks [4, 5] intended to cause vehicles to make incorrect inferences, possibly resulting in increased traffic congestion and a higher chance of severe accidents. Not surprisingly, the problem of providing security in VANETs is starting to attract well-deserved attention [4–6].

As a consequence, much of the recent work assume that VANETs will rely on a pervasive and costly roadside infrastructure that acts as encryption key distribution points or authentication authorities [4, 5]. Unfortunately, in addition to being prohibitively expensive to build and to maintain, this roadside infrastructure is very likely to be the target of vandalism that will hamper its intended functionality. Indeed, the way in which current systems are set up, the driver of a vehicle that participates in the traffic will not be able to preserve their privacy and may be subject to impersonation or Sybil attacks. It was argued that even if pseudonyms are used, detecting the true identity of the driver and, therefore, invading their privacy appears to be difficult to prevent [7].

I.1 OUR CONTRIBUTION

In this thesis, we propose a novel framework for incident detection and notification dissemination in VANETs. This framework consists mainly of three components: a system architecture, a traffic incident detection engine and a notification dissemination mechanism. The basic idea of our framework is to collect data from passing cars about their experience on the road, then use this data to detect traffic anomalies and finally notify drivers about traffic delays and incidents, if any. Figure 1 shows the main three components of the proposed framework and how they interact with each other.

The first component of our framework is an architecture for the notification of traffic incidents, NOTICE for short. In NOTICE, sensor belts are embedded in the road at regular intervals every mile or so. Each belt consists of a collection of pressure sensors, a simple aggregation and fusion engine, and a few small transceivers. The pressure sensors in each belt allow every message to be associated with a physical vehicle passing over that belt. Thus, no one vehicle can pretend to be multiple vehicles and there is no need for an ID to be assigned to vehicles. Vehicles in NOTICE are fitted with a tamper-resistant *Event Data Recorder* (EDR), like the well-known black-boxes on-board commercial aircraft.

EDRs are responsible for storing vehicles behavior between belts such as acceleration, decelerations and change lanes. Importantly, the driver can provide input to the EDR, using a simple menu, either through a dashboard console or through verbal input.

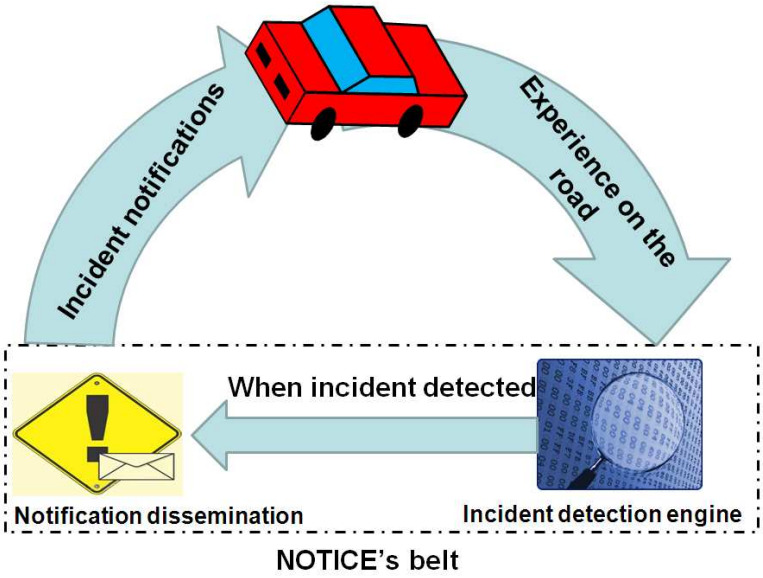

FIG. 1: The proposed framework interacting components

NOTICE's belts are collecting data from passing cars about their experience on the road such as lane changes, stoppages, accelerations and decelerations. This information in turn is fed to our incident detection engine.

Our second contribution is to develop incident detection techniques that use the information provided by cars and drivers in detecting possible incidents and road anomalies. Traditional automatic incident detection techniques have been integrated on the top of our framework, showing that our framework is generic and mixes the benefits of both traditional techniques and VANETs based techniques. To the best of our knowledge, we are the first to propose a VANET based automatic incident detection technique that is capable of detecting both non blocking incidents as well as blocking incidents.

Our third contribution is to provide an analysis for vehicular traffic proving that VANETs tend to be disconnected in many highway scenarios , consisting of a collection of disjoint clusters. We also provide an analytical way to compute the expected cluster size and we show that clusters are quite stable over time. To the best of our knowledge, we are the first in the VANET community to prove analytically that disconnection is the norm rather than the exceptions in VANETs.

Our fourth contribution is to develop data dissemination techniques specifically

adapted to VANETs. With VANETs disconnection in mind, we developed data dissemination approaches that efficiently propagate messages between cars and belts on the road. We proposed two data dissemination techniques for both divided and undivided roads. We also proposed a probabilistic technique used by belts to determine how far should an incident notification be sent to alert approaching drivers.

Finally, we propose a security technique to avoid possible attacks from malicious drivers as well as preserving driver's privacy in data dissemination and notification delivery in NOTICE. We also proposed a belt clustering scheme to reduce the probability of having a black-hole in the message dissemination while reducing the operational burden if a belt is compromised.

I.2 THESIS ORGANIZATION

The remainder of this dissertation is organized as follows, in Chapter II, we briefly describe related work in the areas of incident detection, information dissemination and security in VANET. Chapter III is devoted to present NOTICE, the main infrastructure used by our framework to receive, process, aggregate and analyze data from passing vehicles as well as sending notifications back to them when needed. In Chapter IV, we present our techniques for automatic incident detection implemented on the top of NOTICE. Our data dissemination techniques are described in Chapter VI. Chapter VII is devoted to present a security technique for information dissemination. Finally, Chapter VIII concludes our work and highlights the future research directions.

CHAPTER II

RELATED WORK

In this chapter, we discuss some approaches and techniques proposed in the literature to handle automatic incident detection, data dissemination and security techniques in VANET showing their limitations addressed by the proposed framework.

II.1 AUTOMATIC INCIDENT DETECTION

Roadway incidents refer to non-recurring events resulting in traffic congestion or disruption, including accidents, breakdowns, debris, spilled loads, inclement weather, unscheduled maintenance, construction activities, and other unusual or special events affecting roadways [8]. The goal of an incident detection technique is to automatically identify the occurrence of an unpredicted incident and its location as accurately and quickly as possible [9].

This section is devoted to discussing different automatic incident detection techniques proposed by other researchers. We divide incidents into two categories, temporary and permanent. By temporary incidents, we mean those incidents that block the way and force vehicles to change lanes or either wait until cleared. Vehicle accidents are a good example of this category. On the other hand, permanent incidents do not block the way and vehicles may avoid or may pass over them. Potholes are examples of this category.

II.1.1 Temporary Incident Detection

Most developed techniques, for temporary incident detection, rely mainly on traffic measurements acquired at inductive loop detectors (ILDs) or video detection cameras installed at regular spacing along the freeways. These detectors or cameras measure traffic parameters such as volume, average speed and occupancy, and transmit the measured statistics back to the traffic management center at fixed time intervals of 30 seconds or 1 minute.

Pattern-based algorithms are the most common algorithms in current operation. They work from occupancy, traffic volume and traffic flow information that are usually collected from inductive loops. By identifying patterns in the data that are not considered normal for a stretch of road, potential incidents are recognized. The most famous example in this category is the California Algorithm [10]. The basic idea of the California Algorithm is to compare traffic occupancy differences

between two adjacent detectors. An incident is detected if all of the following values exceeds some preestablished thresholds [10].

1. The absolute difference in the measured occupancy between the upstream and downstream detectors.

2. The difference in the measured occupancy between the upstream and downstream detector stations relative to the occupancy level at the upstream station.

3. The relative difference in the measured occupancy from two minutes ago as compared to the current occupancy level at the downstream detector.

Catastrophe Theory takes its name from the sudden discrete changes that occur in one variable of interest while other related variables are exhibiting smooth and continuous change. These variables are speed, flow and occupancy. When speed drops dramatically without a corresponding increase in occupancy and flow, the alarm sounds. In this regard, Catastrophe Theory based algorithms are able to differentiate between incidents and recurring congestion. The most recognized algorithm that fits into this classification is the McMaster algorithm [11].

Studies showed that relying on cellular phones can detect 38% of the incidents with an average of 5 minutes mean detection time. This is probably because severe incidents are likely to get immediate attention from other road users. Studies also show that cell-phone based detection could detect only 1% of non server incidents such as stall vehicles as they do not get much attention from passing drivers [12].

These are in addition to statistical methods as in [13], artificial intelligence [14], Fuzzy Set Logic [15] and vision detection techniques as in [16].

In [17], a DSRC-based approach is introduced for automatic incident detection where roadside infrastructures are installed along the highway every one mile or so. If vehicles average travel times is longer than the expected time between two roadsides, an alarm is raised.

II.1.2 Permanent Incident Detection

Pothole Patrol or P^2 [18] is a system to detect potholes on the road. In P^2, every vehicle is equipped with an embedded device that detects potholes when they are passed over using three-axis acceleration sensors, which are widely used today in detecting cell-phones orientation and protecting hard drives when they fall down. In P^2, the output of these sensors are being sampled at a high frequency to detect the shock of a vehicle when it passes over a pothole. When a vehicle has

a Wifi connectivity with an open access point or even through cellular networks, it uploads its detection information to a central server over the Internet. This central server in turn aggregates information from various vehicles and produces a list of detected potholes, after some manual filtering.

Vision-based techniques [19] have been proposed in the transportation research community to detect potholes by looking for some patterns in the field of view.

II.1.3 Disadvantages of Current Techniques

All of the techniques that use ILDs or video detection cameras to measure traffic parameters assign vehicles a passive role in the detection process. With the current development and research in VANETs, where vehicles have the ability to communicate with each other as well as with roadside infrastructures, vehicles are able to play a very active role in holding messages and giving inputs to the detection system. So, existing algorithms and techniques are losing much information by ignoring this fact.

Also, it is fundamentally difficult to detect non-blocking accidents and those that occur under light load, as the deviation from normal traffic patterns may be negligible [20].

The third limitation is that most of these algorithms cannot detect where exactly the incident occurred or even what exactly happened. All they can provide is that there is an incident between two ILDs. Thus, inter-spacing between ILDs has to be small enough to get better accuracy of the incident location.

The fourth shortcoming is the high failure rate and maintenance cost of ILDs. For example, studies show that up to 50% of ILDs can be defective at a given time and worse yet the maintenance cost can be up to $40,000$ annually in a single city [9]

The fifth limitations is that relying on cell-phone calls still has some problems because minor events (breakdowns which occur with greater frequency and do not present a hazard to other motorists or some obstacles that block only a single lane) are often not reported by other motorists. Also about 7% of all reported incidents by cellular phones are false alarms (moving violations and other events that could not be verified). The false alarm rate for other events is much higher (32 percent), reflecting cellular-phone callers' difficulties in judging whether vehicles are resting in the freeway shoulders or broken [12].

The sixth limitation is that even though vision detection techniques may be very helpful to detect different kinds of incidents, they would fail at many situations like fog, heavy rain and very bright sun at which most accidents usually

happen. Also, it is very expensive to install these devices along all highways.

Finally, false positive alarms have been unacceptably high for operational purposes [21]. In the case of freeway operations, where detection algorithms continuously verify the existence of incidents, apparently low false alarms may actually demand huge, if not unfeasible, emergency response deployment. For instance, considering that real-time data are fed into the system in 30 seconds time periods, and that the occurrence of an incident is checked at every time period; an AID algorithm with false alarm rate of 2% would yield, on average, 57.6 false alarms per day per pair of neighboring ILDs, which means that, for a single freeway segment containing 70 ILDs, approximately 4,000 (24hours/day x 3600sec/hour / 30sec x 0.02) false alarms would be triggered daily! This represents an average of one false alarm every 90 seconds. Not knowing if such incident alarms are in fact false without further investigation, the Traffic Management Center (TMC) personnel would respond to them diligently at first but soon grow weary of the constant false alarms and discredit the otherwise useful AID. Therefore, the system would eventually be rendered useless and abandoned.

On the other hand for permanent incident detection, P^2 showed good results when implemented over 6 taxis in Boston area. However, it suffers from many disadvantages. Firstly, relying on a central server is neither reliable nor scalable and acts as a single point of failure. Secondly, P^2 generates many false positive alarms because railroads and speed ramps would be detected by a vehicle as potholes. Thirdly, and may be most importantly, typical drivers (those who are not collecting training data) usually strive to avoid potholes by changing lanes or slowing down to save their vehicles. So, P^2 will not be able to detect a pothole when it is avoided.

II.2 DATA DISSEMINATION IN VANETS

Data dissemination is an essential component of VANETs for many applications especially safety related ones to alert drivers about local traffic incidents. Although VANETs shares many concepts with traditional Mobile Ad Hoc Networks (MANETs), VAENTS is characterized by its high mobility and frequent disconnection. This key differentiation causes traditional MANETs routing and data dissemination techniques such as AODV [22] and DSR [23] not suitable for VANETs [24].

Not surprisingly, a number of data dissemination techniques have been proposed for VANET. These techniques can be classified into different categories as being Unicast [25] [26], Multicast [27] or Broadcast [28] [29]. VANET data

dissemination techniques can also be classified as relying on the existence of an infrastructure [30] or they can work with zero infrastructure support [31].

For the sake of this thesis, we divide data dissemination techniques in VANET into two categories: (1) Techniques that assume the existence of end-to-end connectivity between vehicles. (2) Techniques that take lack of connectivity into consideration. Noticeable techniques from the first category are GVGrid [31], MURU [32] and PBR [33]. Noticeable techniques from the second category are CAR [26] for divided roads and DPP [34] for undivided roads.

GVGrid is an on-demand, position-based routing protocol that constructs a route from a static source node to vehicles that may exist in a destination region. GVGrid constructs a routing path from the source to the destination by grid-based approach, which divides the map into several grids. It also maintains the route when it breaks because of the vehicle mobility. GVGrid tries to discover, based on vehicle mobility characteristics, a route that is expected to provide the best stability.

MURU is a multi-hop routing protocol intended to find robust paths in urban VANETs. MURU tries to minimize the probability of path breakage by exploiting mobility information of each vehicle and by using a special parameter called expected disconnection degree factor to select the most robust path from source to destination. MURU implicitly assumes that there will be many paths between source and destination and it strives to select the most stable one.

Position Based Routing (PBR) protocol was presented where packet forwarding decisions are made based on power awareness. The basic routing strategy is a variant of greedy forwarding where the next hop is selected to be the vehicle closest to the destination. While this strategy is correct, it may lead to unnecessary forwarding and, ultimately, to wasting bandwidth.

Connectivity-Aware Routing (CAR) was developed taking into consideration the fact that end-to-end connectivity is not guaranteed in VANETs. The main idea of CAR is to try finding a connected path between the source and destination even if it is not the shortest one, this is being done using a route discovery process before the real data can be sent. This is because the longer fully connected path is better than shortest path that may experience lack of connectivity at some point.

Directional Propagation Protocol (DPP) [34] utilizes the directionality of data and vehicles for packet propagation. DPP considers real traffic scenarios in which vehicles form clusters on the road and these clusters may be disconnected from each other. DPP uses co-directional clusters that run in the same direction as the packet in the data delivery process. When disconnection occurs between two

co-directional clusters A and C, cluster B in the opposite direction will be used as bridges between A and C, if such oncoming cluster B exists. To guarantee packet delivery, DPP uses the idea of message custody, that is the current car a in cluster A that holds the packet will keep buffering that packet until it receives confirmation from some car c in cluster C. However, as we show later, DPP is likely to waste significant bandwidth because of uneven traffic density as well as imposing delays on packets propagation [35].

In [36], an analytical model for DPP is introduced in which the expected distance between clusters, the expected disconnection time and the effective propagation rate were computed. However, the model of [36] does not explain why traffic clustering is inherent to VANETs.

In [30], a realistic traffic scenario is considered in which vehicles may form clusters that are disconnected from each other. A hybrid routing protocol is introduced that can route a packet inside a cluster but relies on a pre-existing infrastructure to connect these clusters. Although, this protocol considers real traffic situations, its reliance on a pre-existing infrastructure is problematic. Indeed, the cost of installing roadside infrastructures along roadway is prohibitively expensive.

The main disadvantage of existing data dissemination techniques is that they either do not take frequent VANET disconnection into consideration such as GV-Grid, MURU and PBR. On the other hand, other techniques that take disconnections into consideration suffer from many disadvantages like routing loops and wasting a lot of limited resources by sending unnecessary messages in their data dissemination.

As we described before in Chapter I, one of our contributions is to provide an analytical proof that end-to-end connectivity is not guaranteed in VANET in may traffic conditions. We also proposed techniques for efficient data dissemination in disconnected VANET for both divided and undivided roads. Therefore, it is natural to compare our proposed data dissemination techniques with those that had same assumption such as DPP and CAR.

II.3 SECURITY IN VANETS

The main goal for VANET is to increase road safety. To achieve this, the vehicles act as sensors and exchange warnings that enables the drivers to react early to abnormal and potentially dangerous situations like accidents, traffic jams or any impacting incidents on the road. In addition, authorized entities like police or fire fighters should be able to send alarm signals and instructions [37]. Like any

communication system, security plays a vital role in VANET communication. As VANET consists of vehicular nodes which are moving at high speeds, efficient and secure routing protocols are highly desirable [38].

Safety applications shall use VANET to communicate; hence the warning messages should be authenticated [38]. Thus, Security is an essential component for VANET when notifying drivers about traffic incidents. Otherwise, an attacker may send notifications about fake incidents, which discredit the otherwise useful system. In order to secure VANETs, the following security requirements should be met [37] [39]:

1. Integrity: the security infrastructure has to provide mechanisms that prevent or at least detect message modification. This prevents malicious vehicles from modifying forwarded messages and protects message integrity for all application categories. Authentication is also needed to keep outsiders from injecting messages about incident that does not exist.

2. Confidentiality: a very dangerous and ignored fact about privacy is that innocent looking data from various sources can be accumulated over a long period and evaluated automatically revealing much information about these sources [40]. Even small correlations of the data may reveal useful information. For instance, the knowledge about specific sensor characteristics may give some hints about the make and the model of a car. This in turn may be related to other information to identify a specific car. Moreover, users are unlikely willing to participate in a system breaching their privacy. VANET's value would actually be very limited If not enough nodes exist [41]. As a matter of fact, privacy is one of the main challenge facing vehicular ad hoc networks. Whenever vehicular nodes attempt to access some services from roadside infrastructure, they want to maintain the necessary privacy without being tracked down for whoever they are, wherever they are and whatever they are doing. It is considered as one of the important security requirements that should be paid more attention for secure VANET schemes [42].

3. Availability: because most VANET messages are related to driving conditions and road safety, fast processing of these messages is important. Also, VANET consists of thousands of vehicles running on hundreds of miles of highways and streets. So, any proposed security approach should be fully distributed and don't rely on a single point of failure or bottleneck to be able to efficiently handle that large scale system.

VANET's security techniques may be divided into two categories namely position verification and securing incident notification. Position verification is required to verify if a vehicles is lying about its position and/or its speed. Several techniques have been proposed in literature to verify whether vehicle's broadcasted position is correct or not. For example, a technique for verifying the claimed location using a directional radar has been proposed in [43]. Another technique that uses radio signal direction strength to determine the transmitter location has been proposed in [44]. Also, a technique that uses received signal strength and laser range finders for position verification has been proposed in [45]

On the other hand, securing incident notification may be defined simply as authenticating the sender of the notification while preserving privacy of both the sender and the receiver. In this thesis, we are interested in securing incident notification in VANETs.

Proposing incident detection techniques with low false positive rate is one of our contribution in this thesis. However, if a malicious attacker could send a false notification to passing vehicles, these passing vehicles would not be able to differentiate whether this is a false alarm from our incident detection engine or a fake message sent by an attacker, which discredit the whole system as drivers will simply ignore even real notification sent by real belts in the future.

Most of the proposed security techniques for incident notification rely on the usage of public authority distributing keys between vehicles. For example, in [46], a Certificate Authority (CA) that provides and manages certificates for all vehicles on the road has been assumed to exist. Thus, a typical authentication approach between vehicles may be performed using the provided public/private keys along with public certificates. Actually, these solutions have many shortcomings. First, the CA is a single point of failure and perhaps very dangerous if compromised by a malicious user. Second, giving a vehicle a few pairs of public/private keys makes it easy to be tracked violating privacy of drivers. Third, and may be most importantly, a malicious driver can authenticate himself and send fake messages to other cars that causes the last problem which is the complexity of granting and revocation of certificates for such a huge community of vehicles [47]. It was also argued in [7] that even if pseudonyms are used, detecting the true identity of the driver and, therefore, invading their privacy appears to be difficult to prevent

In [4], a security protocol was introduced by creating a large number of anonymous certificates in vehicles. With a pool of around 43,800 certificates, each vehicle randomly chooses one of the available certificates for signing the message at one time in order to meet the drivers privacy requirement. Although this

technique can effectively meet the privacy requirement, it can hardly become a scalable and reliable approach, because the ID management authority has to keep all the anonymous certificates for each vehicle in the administrative region. Once a malicious message is detected, the authority has to exhaustedly search in a very huge database with size up to (43800 certificates * millions of cars) to find the ID related to the compromised anonymous public key.

A number of security mechanisms to complement the Public Key Infrastructure (PKI) in VANET havs been proposed in [47]. In that regard, the authors proposed mechanisms to enhance location privacy, efficient authentication and certificate revocation. The authors also proposed a mechanism for efficiently mitigating the effect of a Denial of Service (DoS) attack. However, like any other approach that gives vehicles a pair of public/private keys, distributing the revocation information to the vehicles is still a problem in VANET because of its large scale. Another problem with this approach is that building a global reputation-based system while supporting privacy preservation is hard to obtain because preserving the privacy of users requires frequent identity changes. Consequently, linking the reputation of a user to all its identities may contradict preserving the privacy of that user.

The main goal of our contribution to security in this thesis is to enhance NOTICE architecture to meet all of the security requirements listed above. (1) Vehicles should be able to authenticate belts and receive incident notification only from real ones. (2) A driver's privacy has to be preserved and a vehicle should not be associated with any unique identifier. (3) A malicious attacker should not be able to eavesdrop on the communications between a belt and a vehicle to gain any information about the vehicle's experience on the road, which may reveal its identity. (4) An attacker should not be able to change the meaning of a message or an incident notification sent in NOTICE. (5) The system should be highly available.

CHAPTER III

NOTICE: THE ARCHITECTURE

The main purpose of this chapter is to introduce the basic functionality of NOTICE, a secure and privacy-aware architecture for the *Notification Of Traffic InCidEnts* [48]. NOTICE is the infrastructure used by our framework to detect traffic incidents and to notify drivers about them when they exist.

One of the *underlying philosophies* of NOTICE, and our framework in turn, is that the decision about traffic-related information should rest with the infrastructure and not with individual vehicles that may have incorrect or incomplete knowledge.

Instead of relying on a vulnerable roadside infrastructure, we propose to embed *sensor belts* in the road at regular intervals (*e.g.*, every km or so), as illustrated in Figure 2.

Each belt consists of a collection of pressure sensors, a simple aggregation and fusion engine, and a few small transceivers. For robustness and fault tolerance, roadside solar panels of the type currently used on US highways can supplement the energy needs of the belts. We expect this configuration to be less expensive than a single ILD, even without the expensive optical fiber needed to interconnect the ILDs. The pressure sensors in each belt allow every message to be associated with a physical vehicle passing over the belt. Thus, no single vehicle can pretend to be multiple vehicles and there is no need for an ID to be assigned to each vehicle. There are three immediate benefits of using belts over roadside infrastructure. Firstly, the belts are far less prone to tampering. Secondly, they are better placed to detect passing vehicles and interact with them in a simple, secure and privacy-preservation fashion. Thirdly, a recent prototype [49] has confirmed that suitably encased belts are more robust, more reliable and longer-lived than ILDs.

III.1 BELT MODEL

Each belt is fitted with a few transceivers, at least one per lane of traffic, with a maximum communication range of 6 m. Consequently, the belts do not communicate with each other directly. Instead, adjacent belts rely on passing vehicles to communicate.

Referring back to Figure 2, featuring a two-lane roadway, each lane on the roadway has its own dedicated belt. For example, belt C consists of two logical sub-belts, each serving one lane. In the case of a divided highway, belts on opposite

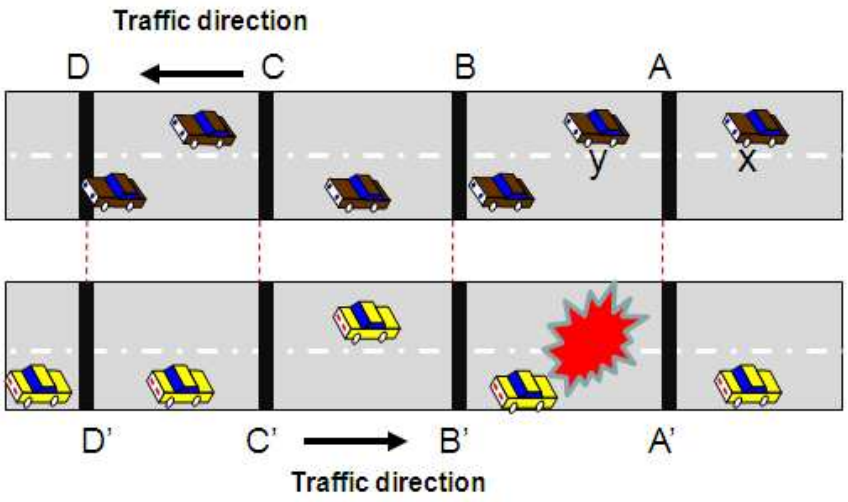

FIG. 2: Illustrating NOTICE architecture

sides of the median are connected by direct wired connection under the median. It is assumed, therefore, that the sub-belts can communicate directly in a secure way.

Referring again to Figure 2, consider the lane wherein the traffic is moving right-to-left. If belt C wants to communicate a message m to the next belt, D, it will encrypt m with a time-varying shared symmetric key $\mu(C, D, t)$ known only to belts C and D, with t representing the time parameter. We assume that the belts are roughly synchronized in time and that they switch from one key to the next in a pre-established key-chain based on their local time. Tight time synchronization between belts is not essential, given the inherent delays in communications.

III.2 VEHICLE MODEL

Here we discuss the basic vehicle's model while more assumptions will be stated later when needed. As has been suggested in [6,7,50], we assume that each vehicle will be fitted with a tamper-resistant Event Data Recorder (EDR), much like the well-known black-boxes onboard commercial aircraft. The EDR provides tamper-resistant storage of statistical and private data.

In its August 2006 ruling, the National Highway Traffic Safety Administration (NTHSA) has mandated that starting September 2010 an EDR will have to be installed in light cars (those vehicles with an unleaded weight of less than $5,000$ lbs). In NTHSA's terminology, an EDR is described as a device or function in a vehicle that records the vehicle dynamic, time-series data [51]. It is, perhaps, less well known that some car and truck manufacturers were offering EDR devices on

a voluntary basis. For example, it is not widely known that some GM vehicles as old as model year 1994 were equipped with an EDR-like device able to store retrievable data. Thus the use of an EDR is very much in line with the stipulations of NTHSA and should not be construed as a substantial change in vehicle design.

The EDR is also responsible for recording essential mobility attributes. For this purpose, all of the vehicles sub-assemblies, including the GPS unit, speedometer, gas tank reading, tire pressure sensors, and sensors for outside temperature, feed their own readings into the EDR. These sub-assemblies can report such attributes as the current geographic position, current speed, momentary acceleration or deceleration, lane changes, and swerving. As a consequence, given a time interval I of interest, the EDR can store information such as the highest and lowest speed during I, the position and time of the strongest deceleration during I, as well as location p, time t and target lane in a lane change.

The EDR is also fitted with a cell-phone programmed to call predefined numbers (including E-911) in the case of an emergency. For example, a driver may be incapacitated as a result of the accident and may be physically unable to place the call. This feature exists already on some vehicles and is a useful for reporting, upon the deployment of an airbag, that the vehicle was probably involved in a collision. This allows the authorities to be alerted in real-time to major traffic events and, ultimately, saves lives. Importantly, the driver can provide input to the EDR, using a simple menu, either through a dashboard console or through verbal input. This is useful feature that allows individual vehicles to alert NOTICE of traffic incidents that are otherwise hard to detect, such as roadway icing and the presence of stray animals on the roadway. Also, false driver input may be verified using other sensors data. For example, a belt can reject a report provided by a drivers about a broken car around a position that his car's EDR shows that the car has passed over that position.

Recently, much research and patents have been proposed for automatic lane detection [52–54]. Consequently, we assume that vehicles can detect lane changes. Moreover, the vehicle's EDR is responsible for maintaining a set of records about every lane change. Each record has the form $< FromLane, ToLane, Position, Time >$. For example, if an EDR contains the following records ($< 0, 0, 1000, t_1 >, < 0, 1, 400, t_2 >, < 1, 0, 200, t_3 >$). This means that the vehicle was originally at lane 0 then at time t_2 and position 400, that vehicle changed lane to lane 1 then it went back to lane 0 at time t_3 and position 200. Note that the first record does not record any lane change, instead it just shows the initial position of a vehicle after the previous belt. Whenever a vehicle

passes through a belt, it drops its EDR data then initializes it and starts recording again until next belt.

Thus, the EDR contains all information about lane changes made, accelerations, decelerations, driver inputs and other input fed by sensors installed at its vehicle.

III.3 BELT TO VEHICLE COMMUNICATION

We now give a succinct description of the communication between a belt and a passing vehicle. Consider vehicle c traveling at 100 km/h (approximately, 65 mph – the legal interstate speed in most US states) and approaching belt C. Once the pressure sensors in belt C have detected the front wheels of vehicle c, a radio transceiver in the belt will send, at a very low power (range of about 1 m), a "Hello" beacon on a standard control channel containing the ID, C, of the belt, as well as handshaking information. This information includes a frequency channel λ on which data is to be exchanged.

Once vehicle c receives this information, it will have roughly 36 ms (time to travel 1 m and thus becomes out of communication range) to respond. As the handshaking response will be very short and will not be encrypted, a NOTICE-equipped vehicle will have no problem responding in time. If belt C does not receive a reply to the handshake, it will not communicate further with vehicle c.

If vehicle c confirms the handshake before it leaves the radio range, belt C will send on channel λ a query that will be received by the vehicle's transceiver. Vehicle c will drop the following to belt C:

- the encrypted message uploaded by the previous belt, say B, if provided.
- the relevant data collected by its EDR in the time interval $I(B, C)$, which is the time spent traveling between belts B and C.

If there is traffic-related information that concerns vehicle c, belt C will upload this information to the vehicle. Belt C may also upload a message m destined for the next belt, say D. Message m is encrypted with the symmetric key $\mu(C, D, t)$, a time-varying shared key between belts D and C that we introduced in Section III.1. The message is stored in the EDR and will be dropped off with belt D at the appropriate time. The vehicle does not know the key $\mu(C, D, t)$ and, consequently, cannot decrypt the message destined for belt D.

For the data exchange between the belt and the vehicle, the belt uses a transceiver with slightly higher range than that of the handshaking transceiver,

about 3 m. Since the transceiver on the vehicle that will perform data exchange is placed at the rear of the vehicle, there will be a total range of 6 m (as the vehicle passes over the belt) for data exchange. This gives the belt and the vehicle about 216 ms to complete the communication.

Here we show that 216 ms is a feasible communication time period for the data exchange between the belt and the vehicle. Let s be the transmission time for a single message, d be the encryption/decryption time for a single message, and p be the processing time for the belt to incorporate new information. There are a total of 5 messages sent after handshaking (belt sends initial query, vehicle sends message from previous belt, vehicle sends EDR data, belt sends new information for vehicle, and belt sends message for next belt) and 2 encryption/decryption events (belt decrypts message from previous belt and encrypts message for next belt). This results in a total communication time $T = (5s + 2d + p)$ ms. If we set $p = 50$ ms, $d = 20$ ms, and $s = 1$ ms (corresponding to a 750-byte message at 6 Mbps, the lower end of DSRC [17]), then $T = 95$ ms. These are conservative estimates, as we anticipate messages to be much smaller than 750 bytes, at least for the first query sent by the belt. Even with these conservative estimates, for 95 ms to be too little time for communication, the vehicle would have to be traveling at 227 km/h (141 mph), an illegal, not to mention an unsafe, speed on US highways.

The very short-range radio transmission used in the vehicle to belt communication is deliberate. It renders the communication strictly *local* and, therefore, reduces the chances of eavesdropping by malicious entities positioned by the roadside. We note here that the belt to vehicle and vehicle to belt data exchanges discussed above are perfectly *anonymous* and do not interfere with vehicle or driver privacy. Indeed, the pressure sensors in the belts allow NOTICE to associate every message with a physical vehicle passing over the belt. We note also that a given vehicle cannot interact with a belt more than once in a reasonable time interval. So, impersonation and Sybil attacks are difficult to perpetrate. In addition, because messages carried by vehicles between belts are encrypted, these messages are secure.

III.4 INCIDENT DETECTION

A belt is collecting EDRs information from passing vehicles where each EDR stores information about the behavior of its vehicle since the previous belt. Thus, NOTICE relies on accumulated pieces of evidence deduced from the EDRs in conjunction with driver input and intelligent data mining to detect traffic-related

incidents. Details of the proposed incident detection techniques are presented in Chapter IV.

III.5 NOTIFICATION DISSEMINATION

Referring to Figure 2, assume that belt A' is aware of the accident and has informed belt A about that. Vehicles in the opposite direction of the accident may be used to carry the message in order to notify other vehicles approaching the accident. For example, when vehicle x passes over belt A, the belt will upload information to x about the incident destined for belt B. In order to propagate the message to belt B, the simplest for vehicle x is to continue traveling until it drops the message with belt B. Thus, message propagation time depends upon the speed of vehicle x. When the information about the traffic incident reaches belt B (and belt C and belt D), it will inform belt B' (and C' and D') to alert vehicles traveling towards the accident. These vehicles in turn may use their navigation system that may suggest an alternate route. Although having vehicles working as data mules, carrying messages, between belts is simple and easy to implement, this technique suffers from long propagation delay especially when we send the message several miles back and/or when traffic on the opposite direction is slow. So, we proposed efficient techniques to disseminate messages between belts. These techniques will be presented in Chapter VI.

III.6 ROLE BASED VEHICLE TO BELT COMMUNICATION

There are exceptional cases where the communication between belts and passing vehicles needs to be augmented to allow authorized vehicles to interact with the belts in a predetermined, role-based, fashion. This feature is essential to the interaction of NOTICE with first responders, ambulances, fire fighters, local police, and traffic management personnel in case of emergency operations.

III.7 CUSTOMIZED INTERFACE

As stated in Section III.2, each driver has an input console to provide input to the detection system. The same console can be used to alert drivers about different road incidents. Also, drivers may have an interface to customize their preference for notifications. For example, a driver may not be interested in notifications about delays less than 5 minuets. So, he would specify that in his console. Actually, this is like having filtering agent at each vehicle that filters traffic notifications based

on a driver's preference.

III.8 SUMMARY

In this chapter, we presented the first component of our framework which is an architecture for the notification of traffic incidents, NOTICE for short. In NOTICE, sensor belts are embedded in the road at regular intervals, every mile or so. Each belt consists of a collection of pressure sensors, a simple aggregation and fusion engine, and a few small transceivers. The pressure sensors in each belt allow every message to be associated with a physical vehicle passing over that belt. Thus, no one vehicle can pretend to be multiple vehicles and there is no need for an ID to be assigned to vehicles. Underlying philosophies of NOTICE, and our framework in turn, is that the decision about traffic-related information should rest with the infrastructure and not with individual vehicles that may have incorrect or incomplete knowledge.

Vehicles in NOTICE are fitted with a tamper-resistant Event Data Recorder (EDR), very much like the well-known black-boxes onboard commercial aircraft. EDRs are responsible for storing vehicles behavior between belts such as acceleration, deceleration and lane changes. Importantly, drivers can provide input to the EDR, using a simple menu, either through a dashboard console or through verbal input.

In NOTICE, belts in the same traffic direction communicate with each other by disseminating data with the help of passing vehicles. Thus, each two consecutive belts share a time variant symmetric key to secure data that is exchanged between them.

CHAPTER IV

AUTOMATIC INCIDENT DETECTION

After describing our infrastructures for both belts and vehicles, this chapter is devoted to presenting our techniques for automatic incident detection implemented on top of NOTICE.

It is important to mention that our techniques are not a replacement for any of the current AID techniques. Instead, they can provide a complementary support to them in order to recover their limitations mentioned in Chapter II.

IV.1 A DETERMINISTIC TECHNIQUE

This section is devoted to presenting our first attempt to develop automatic incident detection in VANETs using a deterministic approach [55]. Although this deterministic technique has many shortcomings, it is worthy to explain it first, before introducing our more generic probabilistic approach in the next section, for better illustration of the problem at hand.

IV.1.1 The Roles of Belts

A belt is responsible for collecting and managing EDR data from passing vehicles. For the sake of collecting traffic occupancy, each belt maintains a table called $RoadImage[m][n]$ where m is number of rows that matches the number of lanes and n is number of columns that matches the distance between two consecutive belts in some units , we can assume it simply to be in meters.

For example, $RoadImage[i][j] = x$ means that x vehicles have passed over the location ($lane = i$, $position = j$) in the previous time interval.

Figures 3, 4 and 5 shows the representation of the contents of this table in case of no incident, single lane blocking incident in a two-lane freeway and single lane blocking incident in a three-lane freeway respectively. The x-axis represents different points of the road and the y-axis represents the total number of passing vehicles through each position.

In the incident free situation, Figure 3, the two lanes have almost the same occupancy, as vehicles move freely between them. On the other hand, Figures 4 and 5 show that vehicles would change lanes at some position before the incident to avoid it and hence we expect to have positive peaks at incident-free lanes while we have negative peaks in the blocked lanes. To maintain the RoadImage table, a simple rule may be applied as follows: whenever a belt receives an EDR update

from a passing vehicle, it increments all the positions that the vehicle has passed over by one.

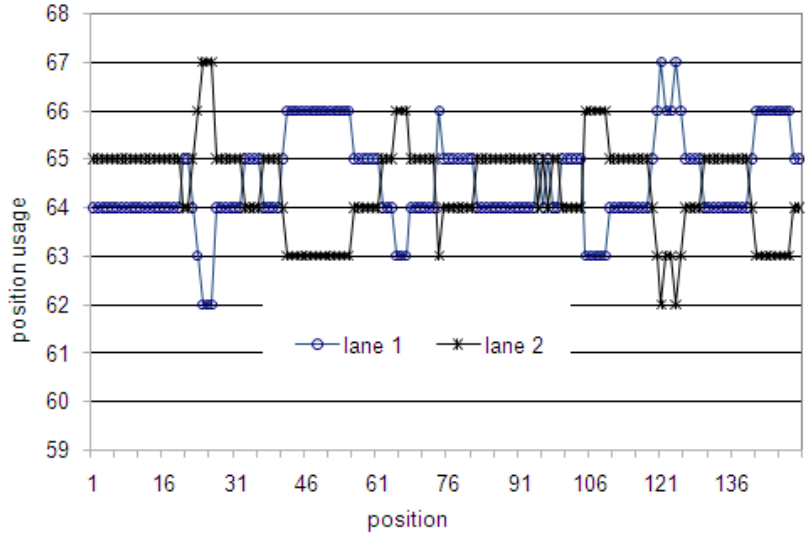

FIG. 3: Two lanes incident free situation

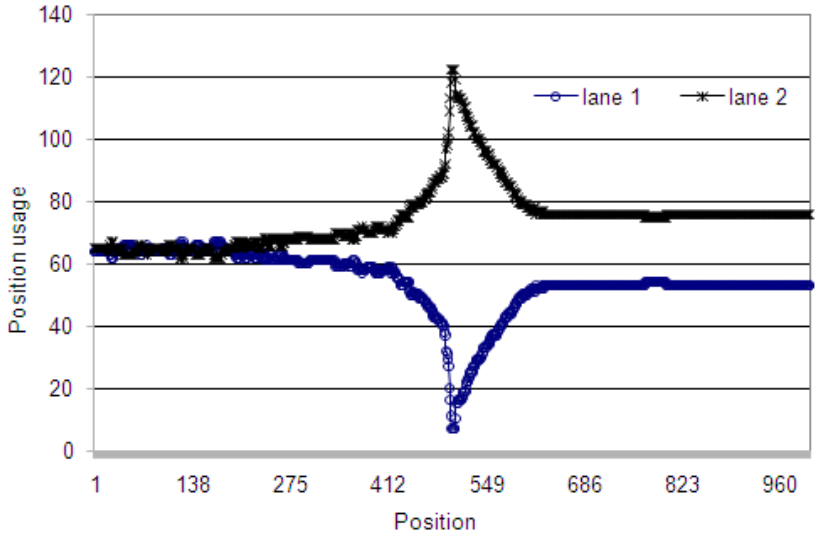

FIG. 4: Incident blocks single lane in two-lane highway

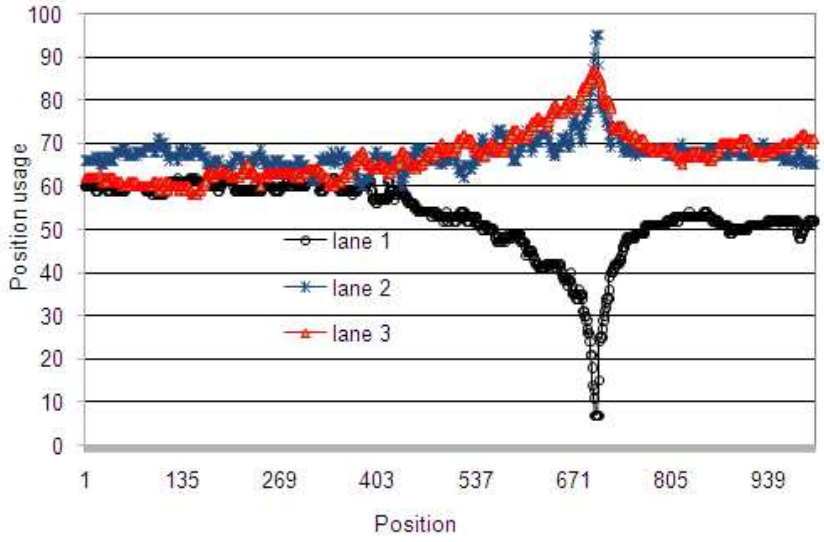

FIG. 5: Incident blocks a single lanes in three-lane highway

As shown in these figures, if an incident occurred, a belt would expect to have a negative peak in the row corresponding to the lane containing the incident in the RoadImage table while other lanes are still normal or have positive peaks especially for lanes adjacent to the incident's lane. So, one may argue that detecting an incident is simply to detect such a peak in the RoadImage table.

Although this idea is simple and easy to implement, it has many shortcomings that may be explained as follows. Consider the example shown in Figure 6 where the numbers in that table represents occupancies for the corresponding lanes and the shadowed area represents EDR data that has been just received by a belt, i.e the EDR showed that its vehicle has recently passed through the shadowed positions. Figure 6(a) shows the contents of a section of the RoadImage table before applying the new EDR. As shown in Figure 6(a), the middle positions of lane 1 have very low values meaning that very few vehicles have passed over these positions recently. That may be a reason to suspect an incident at these positions. If a belt applied the basic filling algorithm, which just counts how many vehicles have passed over each position, for the new EDR, then the belt would have the table shown in Figure 6(b).

Figure 6(b) shows that the suspected positions still have very low values relative to corresponding positions in the other two lanes that make it still be suspected. However, having a vehicle that has recently passed over these positions should override previous history for them and remove any suspicion accumulated over time about them. The above situation may result in many cases. For example, if an incident has occurred for a very short duration, because of a slow vehicle or

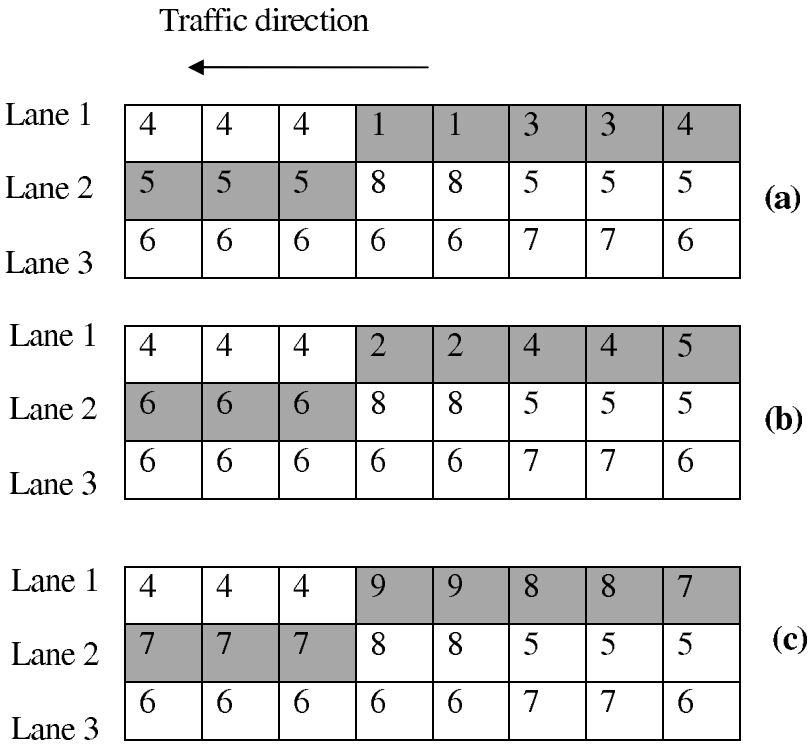

FIG. 6: Illustrating the need for the modified table filling

a temporary broken vehicle, then a history might be built against some positions as shown in Figure 6 until one vehicle passes over these positions. The problem is that it would take a very long time until the table becomes balanced again even if one vehicle was enough to remove any suspicion about these positions.

IV.1.2 Modified Table Filling

Here, we present our idea to overcome the problems depicted in Figure 6(a). The following rule is used to update the RoadImage table after receiving a new EDR from a passing vehicle: If a vehicle has passed over a certain position, this position is clear and must have value larger than corresponding positions in other lanes. The reason for this is that when a vehicle passes over position $(lane = i, position = j)$, it means that this position is clear and any history against this new fact must be forgotten.

Referring again to Figure 6(a), the history of this section of the road is against the middle positions of $lane_1$, because they have a very low values. But, once a vehicle passes through them, a belt must change this history by making them larger than corresponding positions on other lanes as in Figure 6(c).

The main advantage of the modified approach is that it has a rapid convergence, once a position is cleared, the table will show that immediately. Thus, any temporary problem on the road or any outdated history would not affect the belt's decision. Also, after detecting an incident, once a vehicle passes over the incident position, the table will show an incident-free status.

It is noteworthy here to mention that the values in the RoadImage table now do not reflect the number of vehicles that have passed though every position as before. Instead, they just reflect the status of the road.

IV.1.3 Time Dependent Modified Filling

In Subsection IV.1.2, if a belt receives EDR_1 from vehicle x at time t_1 and receives EDR_2 from vehicle y at time t_2 where $t_1 < t_2$, we had an implicit assumption that x was always ahead of y since the last belt. This is because EDR_2 is applied on the RoadImage table after EDR_1 has been totally applied.

Of course, that is not true as a general case as vehicles may accelerate and pass each other. Thus, if x arrived at any position before y, y may accelerate and arrive at next position before x.

The simplest example for this situation is when a slow vehicle passes over a certain position p then an accident occurs at p. Fast vehicles may arrive first to next belt and provide some information about the incident. However, according to our technique, when the slow vehicle arrives at the belt, that belt may, wrongly, clear that position and give it high value in the RoadImage table, which is not correct. Of course, many more sophisticated examples may be shown here to show the effect of time on deducing incidents.

So, we modify the proposed algorithm to catch these situations as follows. First, a belt modifies the RoadImage table to contain not only the counter for each cell but also the last time when that counter was changed. Thus, each cell in the table will be on the form $< Count, LTime >$.

Whenever an EDR reports that its vehicle has passed over any position, the belt checks the reported time with the last time stored in the table for that position, i.e the last time a vehicle passed over that position. If the current reported time is larger than the last time stored in the cell, or the reported time is smaller than the last time by certain threshold, then the belt changes it as described before.

Otherwise, the belt will simply ignore that report because it is outdated and should not override newer reports.

IV.1.4 Incident Detection

In order to detect whether an incident has occurred on the road or not, a belt needs to detect whether a peak with a certain value has occurred in any row/lane of the RoadImage table or not. One way to detect such a peak may be described as follows:

1. Compute the average (μ) and standard deviation (σ) for $Count$ values for each row in the RoadImage table. i.e. for each lane.
2. Find the minimum $Count$, $Count_{min}$
3. Use the idea of bandpass filter to take away regular oscillation and fluctuation from the values. Actually, this step is very important as we are interested in large negative peak at some positions *given that other positions have normal counts*.
4. If $\mu - \sigma - Count_{min} > K$ then raise an alarm for an incident, where K is a detection threshold that determines how conservative the detection should be. The larger the detection threshold, the more conservative the detection is, the more detection time is needed and less false alarms are generated.

IV.1.5 Table Cleaning

In order to prevent values in the RoadImage table from growing to infinity, whenever a belt has a small workload, it can clean the RoadImage table by simply subtracting the minimum value from all values in the table. Hence, the status of the table is preserved while decreasing its values.

IV.2 A PROBABILISTIC TECHNIQUE

Although the deterministic AID technique introduced in Section IV.1 was novel in attempting to solve the AID problem using VANETs, it still suffers from many shortcomings. Firstly, it produces many false positive alarms, specially if we are interested in short detection time. As we showed in Subsection II.1.3, even a 1% false positive alarm rate is not acceptable to traffic operators and drivers. Secondly, it does not consider the relationship between positions at which a vehicle has changed lanes and the location of the accident. That is whenever a vehicle changes lane at location $i > 0$, the deterministic approach assumed that an accident might have occurred at any location between 0 and i in that lane. In other words, the

effect of lane change is the same for all of the avoided positions. However, we will show shortly that this is not true.

Thirdly, both the deterministic technique and other AID models proposed in the literature present two major problems that are conducive to increasing levels of false alarms, namely calibration complexity and lack of universality (or transferability). Even the simpler algorithms require considerable calibration efforts (not to mention the development of an incident dataset, which is not always available) to determine the best algorithm threshold values for each individual, or pair of, ILDs/belts. Actually, it is very hard to determine what thresholds are reasonable for each section of the road thus making the operation of configuring NOTICE, and other ILD based techniques, very hard.

Finally, and may be most importantly, it is very difficult to incorporate other parameters, drivers inputs or deceleration, in the detection process.

In this subsection, we present a probabilistic AID technique based on Bayesian theory that avoids the disadvantages of the deterministic technique and other existing ILD based approaches [56–58].

IV.2.1 Basic Idea

Bayesian networks are known to be used for calculating new beliefs when new information (evidence) is available [59]. The basic task of the inference system is to compute the posterior probability upon arrival of an evidence. This is called belief updating or probabilistic inference.

For example, if we consider the effect of lane changes on the probability of having an incident (accident) on the road. Assume that a belt knows from historical data that the probability (its belief) of having an accident on a given section of the road is p. If that belt noticed that there are pieces of evidence about many lane changes that are correlated in time and position, then the belt may need to update its beliefs about having an incident that might have existed and caused these many correlated lane changes.

According to Bayesian theorem, the new belief or posteriori probability could be computed as

$$Bel(Incident = true) = \frac{P(incident = true) * P(change\ lane|Incident = true)}{P(change\ lane = true)}$$

$$= \alpha * P(incident = true) * P(changelane|Incident = true)$$

Where $P(incident = true)$ is the prior probability, $P(change\ lane|Incident =$

true) is the likelihood and α can be computed by the law of total probability as we will show shortly.

In general, let $\Pr[I]$ to be the *a priori* probability (or belief) of an incident I at a given position on the road. When pieces of *evidence* $E's$, correlated in both time and position, about an existing incident are collected, a belt will update its beliefs by using a Bayesian mechanism. A belt computes the *posteriori* probability of an incident I at the given location as

$$Bel(I) = \frac{\Pr[I] \cdot \Pr[E|I]}{\Pr[E]} = \alpha \cdot \Pr[I] \cdot \Pr[E|I] \tag{1}$$

where $\Pr[E|I]$ is the likelihood, E represents any evidence such as changing lanes or passing over a road anomaly and α is computed by the law of total probability as

$$\frac{1}{\Pr[I] \cdot \Pr[E|I] + \Pr[\overline{I}] \cdot \Pr[E|\overline{I}]}. \tag{2}$$

Where $Pr[E|\overline{I}]$ is the probability of false information. For example, if E is driver input, then $Pr[E|\overline{I}]$ is the probability that a driver inject information about non-existing incident.

The general idea of our technique is to start with some *beliefs* about having incidents on the road. If there are pieces of *evidence* about many lane changes, sudden deceleration and driver input that are correlated in time and position, then a belt updates its beliefs, using Bayesian theory, about having a road anomaly, that might exist and that triggered these many correlated pieces of evidence.

Referring back to Equation 1, two values should be computed to incorporate any evidence E in the detection process namely $Pr[E|I]$ and $Pr[E|\overline{I}]$.

As already mentioned before, detection thresholds, used by the deterministic technique and other ILD AID approaches, are very hard to determine and they are just some *magic* numbers that should be discovered somehow. On the other hand, our Bayesian based approach uses probabilities to determine the existence of incidents and threshold in that context represents how conservative a belt is about its detection process. Thus, making NOTICE easy to deploy in new environments with minimal configurations.

We will start by describing our proposed technique to detect blocking incidents, accidents for example. Then our approach will be extended to detect potholes as well.

IV.2.2 Blocking Incident Detection

In this section, we present our approach to detect blocking incidents by collecting pieces of evidence about lane changes and driver inputs from passing cars. In other words, two types of evidence, lane changes and driver inputs, are considered in the detection process.

Lane Change Model

Referring to Figure 7, assume that an accident has existed on the road at some position y. It is very natural that most vehicles would change lane to avoid the accident at some moderate distance away from y, around position p. On the other hand, very few vehicles would change lane at a large distance or at a very small distance shown as shadowed areas in Figure 7. Hence, the normal distribution for lane changes with respect to the incident location applies. It is noteworthy here to mention that non dense traffic is assumed where vehicles can easily change lanes on desire. On the other hand, the normal distribution may not be appropriate in dense traffic where lane change may not be an easy job. Assume that an incident

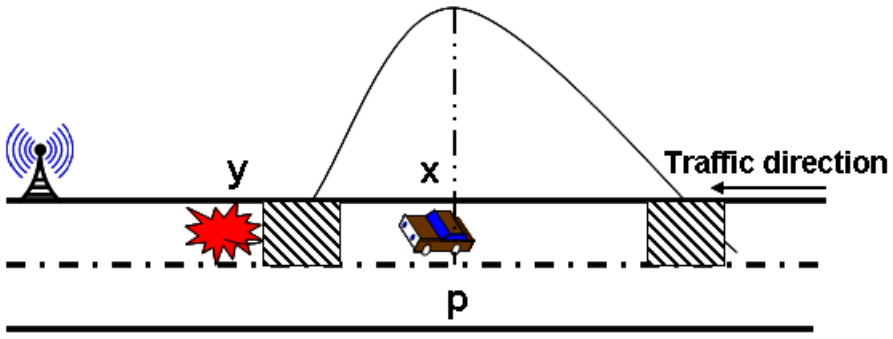

FIG. 7: An illustration of vehicles changing lane

has occurred at position y on the road and let X be the random variable that keeps track of the position at which vehicles change lanes. Since, as we postulated, X is normally distributed, we write

$$f_X(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-(x-p)^2}{2}}. \tag{3}$$

Driver Input Model

A number of cell phone-based incident detection mechanisms have been proposed in the literature [12] showing that that driver input would provide much help to

our incident detection engine. Refer to Figure 8, usually the driver avoids the incident first by changing lane from lane 0 to lane 1. Then somewhere after the incident, the driver may provide an input about it. Usually most drivers have the same behavior and would provide input around position y. Let Y be a random variable that keeps track of the position at which a driver would provide input after an incident. We can write

$$f_Y(y) = I(D) \cdot \frac{1}{\sqrt{2\pi}} e^{\frac{-(p-y)^2}{2}}. \tag{4}$$

where I(D) is an indicator function that returns 1 if the driver provided an input about the incident and returns 0 otherwise.

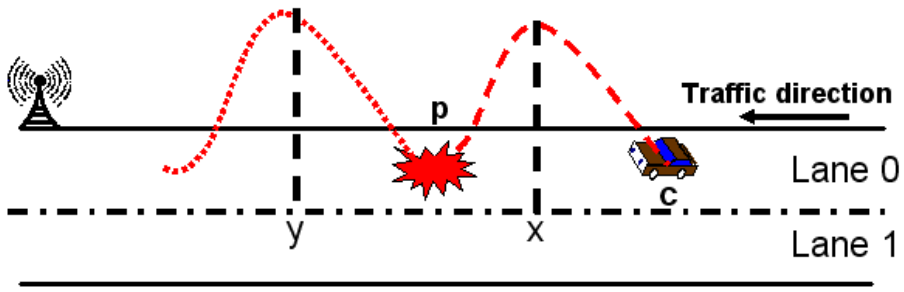

FIG. 8: Illustrating driver input

Belts Roles

As before, a belt is responsible for collecting and managing EDRs data from passing vehicles. Each belt maintains a table called $TempProb[m][n]$ where m is number of rows that matches number of lanes and n is number of columns that matches number of segments, distance, between two consecutive belts. The purpose of this table is to store the probability of having a blocking incident at each position, or section, of the road. The higher the value of $TempProb[i][j]$, the higher the expected chance of having an incident at location (lane $=i$ and position $= j$) where all probabilities were initialized to a very small value representing the original probability of having an incident on that road.

When a belt receives EDR data from a passing vehicle, it applies Bayes's theorem to update the posterior probabilities for different road positions based on the new EDR data.

We use Figure 9(a) to describe the operations performed when a belt receives an EDR from a passing vehicle where the transparent vehicles represents the old positions of the solid vehicle while moving. Assume that a vehicle has run over

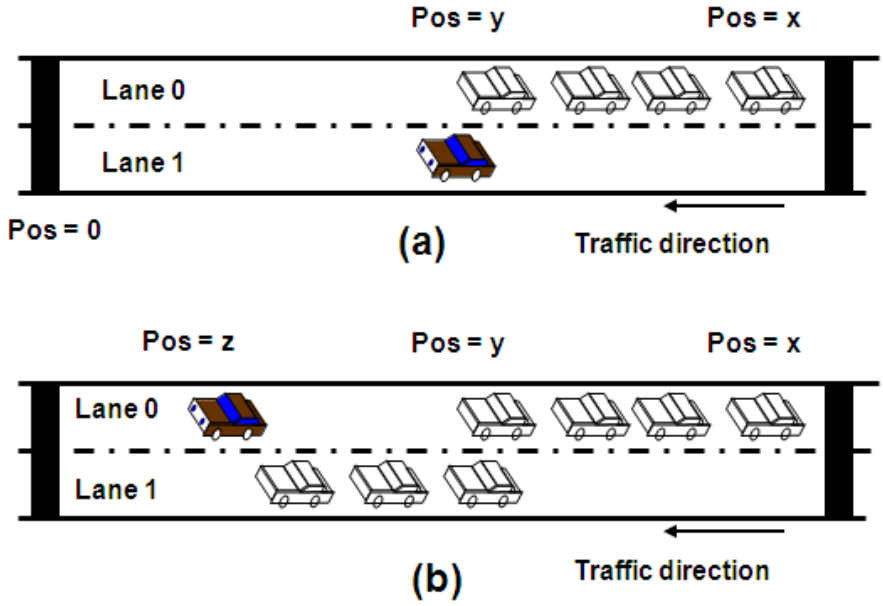

FIG. 9: Illustrating the probabilistic technique

lane 0 from position x to position y at which the vehicle moved to *lane* 1. Since the vehicle has passed over *lane* 0 from position x to position y, these positions between x and y must be clear of any accidents and hence must be assigned a very low probability of having a blocking accident there. In our example, we re-initialize $TempProb[0][j]$ for $y \leq j \leq x$. Thus, whatever the probability of a position was, it would be re-initialized once one vehicle passed over it.

Now, since the vehicle has changed lane at position y, it might have done that because of an accident (incident) that existed ahead of that vehicle at lane 0. Based on this new evidence, a belt would update the posteriori probability of having an accident at any of the locations from 0 to y at lane 0 as follow.

$$TempProb[0][i] = \frac{TempProb[0][i] * P_{y,i}}{P(changelane = true)} = \alpha * TempProb[0][i] * P_{y,i} \quad for\ 0 < i < y \tag{5}$$

where $P_{y,i}$ is the probability of changing lane at position y given that an accident had occurred at position i and can be computed using Equation 3 and α can be computed using Equation 2.

In general, the proposed automatic incident detection technique can be described as follow. Let us define those positions that a car has passed over as *passed over positions* and the positions that the car has not passed over as *avoided*

positions.

For each position $p = (lane = i, position = j)$ that a car x has passed over, p must be clear of any blocking incident and hence must be assigned a very low probability of having an incident there. So, we set

$$TempProb[i][j] = Initial\ probability$$

Thus, irrespective of what the probability of a position was, it would be re-initialized once a car passed over it.

On the other hand, for each position $p = (lane = i, position = j)$ that a car has avoided, the following is performed

- The belt computes the posteriori probability of having a blocking incident at p that forced x's driver to change lane before p as in Equation 1

$$TempProb[i][j] = \alpha \cdot TempProb[i][j] \cdot P_{y,j} \tag{6}$$

  where $P_{y,j}$ is the probability of changing lane at position y given that an accident occurred at position j and can be computed from Equation 3 and α can be computed by Equation 2.

- If x's EDR shows a driver input about position p, a belt also updates the posteriori probability as

$$TempProb[i][j] = \alpha \cdot TempProb[i][j] \cdot P_{z,j} \tag{7}$$

  where $P_{z,j}$ is the probability that the driver provided input at position z given that an incident did exist at position j and can be computed using Equation 4.

Thus, to apply the downloaded EDR data, a belt switches between two main operations. First, the belt re-initializes the probabilities for those positions at which the car has passed over. Second, the belt computes the posteriori probabilities for those positions that the driver has changed lane to avoid and/or provided an input about.

There are two main scenarios that may need to be clarified. Referring to Figure 9(b), if the vehicle changed lane back to lane 0 at some position $z < y$ and continued there until it met the belt. According to our probabilistic technique, positions from y to z at lane 1 and from 0 to z at lane 0 would be re-initialized. Thus, for lane 0, only positions from y to z have a new posteriori probability while all other positions for lane 0 were re-initialized.

The second scenario is when we have a slow vehicle on the road. At this case, vehicles may change lane to pass it and they may return after that to the original lane. Let us assume that one vehicle C_1 changed lane at position P_1, another vehicle, C_2, changed lane at position $p_2 < p_1$, *because the slow vehicle is moving*, and the i^{th} vehicle C_i changed lane at position $P_i < P_{i-1} < ... < P_1$. According to our probabilistic technique, after receiving C_2'EDR data, the probabilities of all positions between P_1 and P_2 would be re-initialized. After receiving C_3'EDR data, the probabilities of all positions between P_2 and P_3 would be re-initialized. In general, after receiving C_i'EDR data, probabilities of all positions between P_{i-1} and P_i would be re-initialized. Also, if any of these vehicles returned back to the original lane, or even when the slow vehicle arrives at the belt, the remaining positions from 0 to P_n would be re-initialized. It is noteworthy to mention that to have a large probability for any position, many posteriori updates needs to be accumulated over time without any re-initialization. Also, we apply the same technique presented in Subsection IV.1.3 to avoid other problems caused by slow cars. Thus, it is clear that the probabilistic technique would not report false alarms because of slow vehicles or other type of fake incidents. This is because vehicles coming after the slow one will change lane at different positions resulting in re-initializing probabilities of these positions.

IV.2.3 Permanent Incident Detection

As we already mentioned before, one of the strongest points of our Bayesian approach is the ability to extend it to incorporate many pieces of evidence and also to detect various type of incidents. In this section, we describe how to extend our approach to detect permanent non-blocking incidents such as potholes on the road. There is one main difference between detecting blocking and non-blocking incidents. Namely, a driver may pass over a non-blocking incident if he could not avoid it while he has to change lane to avoid the blocking one. Thus, not every driver will be able to, or have to, change lane to avoid passing over a pothole. So, similar to Equation 3, let Z be a random variable that keeps track of the position at which vehicles change lanes to avoid passing over a pothole. We can write

$$f_Z(z) = I(D).\frac{1}{\sqrt{2\pi}}e^{\frac{-(z-p)^2}{2}}. \tag{8}$$

where $I(D)$ is an indicator function returning 1 if the driver could change lane to avoid the pothole and 0 otherwise.

To enhance our technique's capability of detecting potholes, in addition to vehicle's assumptions introduced in Section III.2, vehicles are assumed to be equipped

with a sensing device that can detect when they pass over a speed bump or a pothole. Similar to all other devices, this sensing device should feed its readings to its vehicle's EDR.

Similar to the $TempProb$ table, a belt maintains another table $PermProb[m][n]$ where m is number of rows that matches the number of lanes and n is the identity of the column that is equal to number of segments between each two consecutive belts. Moreover, each belt maintains a list $ManMade$ that stores information about man-made road anomalies like railroad crossings and speed bump positions. This information is very helpful to automatically filter out these anomalies when reported by cars.

Maintenance of $PermProb$ table can be done in a similar way as $TempProb$ as follow.

For each position $p = (lane = i, position = j)$ that a car x has passed over, the following steps are performed:

- If x's EDR has a record for a suspected pothole at position p, based on the input fed from the sensing device where $p \notin ManMade$, then the belt computes the posteriori probability of having a pothole at this position, updates its beliefs, as

$$PermProb[i][j] = \alpha \cdot PermProb[i][j] \cdot \Pr[Detection|Pothole] \qquad (9)$$

  where α is computed as discussed before and $\Pr[Detection|Pothole]$ is the probability that the sensing device successfully detects a pothole when one exists.

- If x's EDR shows that x has significantly decelerated around position p, $p \notin ManMade$, this means that there might be a pothole at position p that the driver wanted to avoid or reduce its effect on his car. None of the techniques proposed in the literature can detect such behavior because the car may slow down enough to cancel the effect of the pothole and hence no device can detect such potholes when passing over very slowly. In this situation, the belt needs to update its belief about having permanent incident at p as

$$PermProb[i][j] = \alpha \cdot PermProb[i][j] \cdot \Pr[Reduce|Pothole] \qquad (10)$$

  where α is computed as before and $\Pr[Reduce|Pothole]$ is the probability that the driver slows down when he sees a pothole. This probability depends on many factors like driver response and how close are other cars behind the driver.

- If x's EDR shows nothing about position p, two options exist. Either that position is clear or the driver could avoid the pothole by taking it between wheels or maneuvering around it without changing lanes. Hence, when x's EDR shows nothing about position p, the belt will simply ignore that report about p.

On the other hand, for each position $p(i, j)$ that car x has avoided, the following steps need to be performed, assuming that x's driver has made the last lane change at position $y < j$. We may need to compute the posteriori probabilities as follow. It is important first to note that a driver might have avoided position p because it may have pothole, a temporary accident or some other non-incident situation such as a slow car.

- x's driver might have changed lanes because of a pothole at p. So, the belt computes the posteriori probability of having a pothole based on this new evidence

$$permprob[i][j] = \alpha \cdot permprob[i][j] \cdot P_{y,j} \tag{11}$$

  where $P_{y,j}$ is the probability of changing lane at position y given that a pothole does exist at position j and can be computed from Equation 8.

In contrast to our technique for accident detection, pothole probabilities are never re-initialized after receiving an EDR from a passing vehicle. This is because a pothole might exist but the driver could avoid it by taking it between wheels for example. However, if a belt continues updating pothole probabilities with the arrival of each EDR showing lane changes, false positive alarms will be generated after some time. Therefore, it is important to re-initialize pothole probabilities after some detection duration to avoid the detection of false potholes. If this detection duration is very short, shorter than the mean detection time, we may never detect any pothole. On the other hand, if it is very long, false alarms will be eventually generated. We will study this parameter setting shortly in Subsection IV.4.3.

IV.2.4 Incident Detection

If any of the computed probabilities exceeds a certain threshold, an alarm is raised about an incident at the corresponding position. The larger the threshold is, the more conservative the belt is, the longer the time needed to detect incidents, the less incident detection rate and fewer false alarms reported.

One of the strong points about our proposed techniques is that the threshold is not a magic number that is very hard to set like most traditional AID techniques. On the other hand, the values in our case are probabilities that have meaningful information and setting a threshold is just how confident a belt is in detecting an incident.

It has been noticed in real roads that lanes do not have equal preference (usage) from drivers, i.e. drivers may prefer some lanes over others. For example, if the drivers know that certain lane will be right only (or left only) after a while, they would prefer to stay away from that lane early enough if they don't want to make that turn.

Thus, some sections of the road may be avoided even if they are clear of any incidents. So, we may use different threshold values for different sections of the road. For the above mentioned example, we may assign large threshold for those not-preferred section to avoid generating many false alarms.

One of the important points to mention is that a belt will not declare the existence of incident once it detects a probability larger than the detection threshold. It actually declares an incident if the probability exceeds the threshold for multiple data points to avoid any transient flapping in the probabilities. The same technique is also used to declare the clearance of an incident, that is a belt waits until the probability is below the detection threshold for some data points before declaring the incident clearance.

One more advantage of the probabilistic approach is its ability to tune the initial probability of having an accident on the road. A belt, after its installation, may start with a very low probability, < 0.0001. Then as it infers and detects more accidents on its local section of the road, it can simply adjust that initial probability based on current history of accidents making our technique a self learning detection engine.

IV.3 INTEGRATION WITH EXISTING TECHNIQUES

For permanent incident detection, the proposed technique is novel in the sense that none of the existing AID techniques would outperform it in detection rate, if any even can detect those kind of incidents at all. However, for temporal incident detection like accidents, the proposed Bayesian-based technique would work well under non dense traffic. When traffic becomes very dense and/or the incident blocks many lanes, vehicles will simply be stuck behind the incident and would not be able to continue and provide their information to next roadside and hence

the proposed technique would not work as intended.

Also, we believe that the market penetration will impact the deployment of our technique because it will take time until s sufficient number of cars will be equipped with EDRs and wireless capabilities.

Therefore, to overcome these limitations, we propose to integrate our techniques with existing work from literature to provide improved performance as market penetration increases. For evaluation purposes, we integrated our probabilistic technique with the California Algorithm.

The good thing about the proposed NOTICE architecture is that any ILD-based technique can be implemented perfectly on top of it with minimal changes, if any. Thus, NOTICE can be built and installed *today* on roads and any ILD based technique can be implemented on top of it while it also supports the adoption of equipped vehicles over time.

IV.4 PERFORMANCE EVALUATION

An accurate simulator is a very important part in an incident detection system to mimic the exact behavior of vehicles on the road. We developed a mobility traffic simulator in Java based on the vehicle following model and Intelligent Driver Model (IDM) [60]. Drivers may accelerate or decelerate but should maintain some safety distance between cars. Drivers are also divided into two categories: normal and greedy drivers, which is a known characteristic of traffic. Greedy drivers try to take advantage of every possible situation like changing lanes whenever possible in order to speed up when the free distance in the other lane is larger than their own lane. several entry points are distributed along the highway with one entry every 1000 m. Also, a vehicle may take an exit with some probability, and exits are distributed one every 1000 m. Different lanes may have different average speeds which is a well-known scenario in many areas, and even enforced by law in some countries like Germany.

Unless otherwise specified, we assume a detection threshold of 0.7, the distance between belts is to 1000 meters, 60% of drivers provide input to the system where 10% of those inputs are incorrect, and traffic flow is assumed to be 1200 vehicles/hour/lane.

For the sake of different experiments, the traffic is given some time to warm up so that cars distribute themselves over the road. Then an accident is deployed at random between two belts before it is cleared after some time duration, which is 15 minutes unless otherwise specified. If a belt could not detect the accident

before its clearance, it is considered undetected.

IV.4.1 Performance Metrics

The performance of an AID model is usually evaluated using three indices commonly adopted in AID research [10]: detection rate (DR), false positive rate (FPR) and mean time-to-detection (MTTD), which can be defined as follows:

- Detection rate is defined as the ratio of the number of incident cases correctly detected by the algorithm to the total number of incident cases known to have occurred

$$DR = \frac{Number\ of\ incidents\ detected}{Total\ number\ of\ incident\ cases} X100\% \tag{12}$$

- False alarm rate is defined as the ratio of the number of false alarm cases to the total number of applications or decisions made by the algorithm

$$FPR = \frac{Number\ of\ false\ alarms}{Total\ number\ of\ incident - \ free\ input\ patterns} X100\% \tag{13}$$

- The mean time-to-detect is the average time an algorithm takes to detect incidents. It is measured as the mean delay in seconds between the apparent occurrence of an incident and its detection, averaged for all incidents detected over a period of time

$$MTTD = \frac{1}{n}\dot{\sum}(t_d - t_o) \tag{14}$$

  Where t_d and t_o are the detection and occurrence times of an incident respectively.

IV.4.2 Temporary Incident Detection

Impact Of Traffic Flow

Figure 10 shows the impact of traffic flow on mean detection time for our probabilistic technique, California based technique (CA) and the integration of both. Under very sparse traffic, the probabilistic technique requires long time to collect sufficient number of reports from passing cars and come up with a confirmed probability about the incident. As the traffic becomes denser, less time would be needed because more cars would exist and provide their input to next belt. However, under dense traffic, more than 2500 cars/hr/lane, lane changes would be

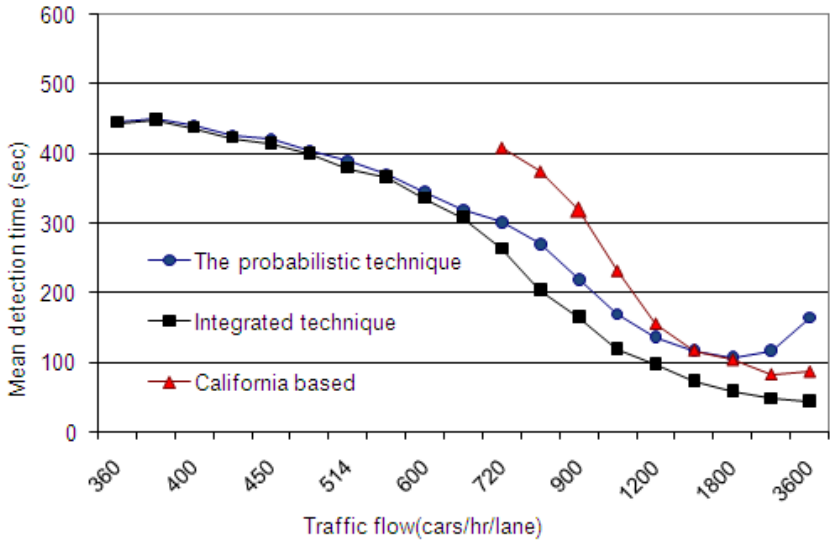

FIG. 10: Impact of traffic flow on mean detection time for accident detection

very difficult and cars would slow down or stop to make a lane change and hence more time would be needed to detect incidents.

On the other hand, it is fundamentally difficult for existing techniques , including CA, to detect incidents under sparse traffic as the deviation in traffic parameters is very small. With the increase of traffic flow, CA was able to detect incidents before they are cleared with longer detection time than our probabilistic technique. Actually, this is not a surprise as the probabilistic technique is fed with more pieces of evidence than traffic occupancies.

As expected, the integrated approach takes advantage of both techniques. Under sparse traffic flow, the integrated approach behaves like our Bayesian based technique. Under dense traffic, the integrated approach inherits the benefits of both Bayesian and CA.

Figure 11 shows the impact of traffic flow on the detection rate. For the same reasons described above, as the traffic flow increases, the detection rate for our Bayesian approach increases until it becomes 100% at 900 cars/hr/lane. However, as we explained before, under very dense traffic, more than 2500 cars/hr/lane, the detection rate decreases.

For CA, under sparse traffic, it could not detect the accident. As the traffic flow increases, it would be able to detect more incidents until its detection rate reaches 100% under dense traffic.

Figure 12 shows the effect of traffic flow on false positives for both CA and

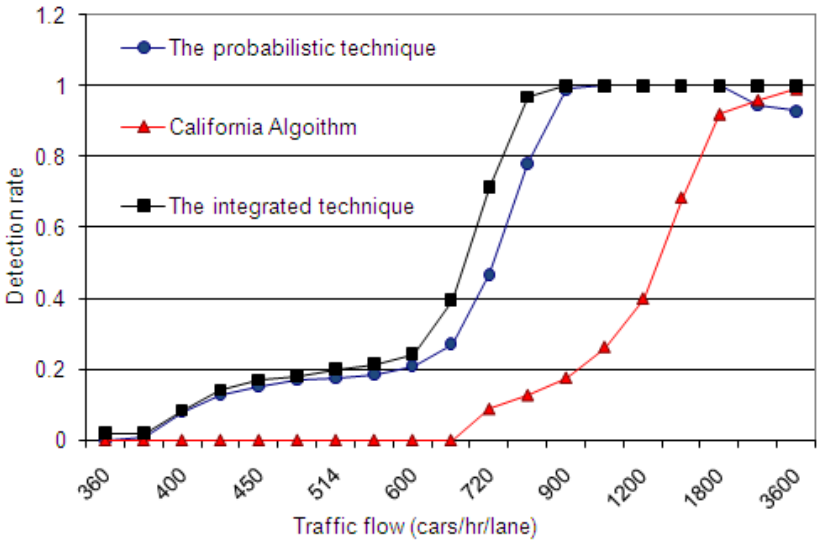

FIG. 11: Impact of traffic flow on detection rate for accident detection

the deterministic technique. The probabilistic technique offers a *zero* false positive alarms, assuming 0.7 detection threshold, which makes it perfect and very trustable to traffic operators. This is because the proposed technique re-initializes any position that a car has passed over. Thus, even if some pieces of evidence accumulated against some positions by a slow car for example, these pieces of evidence will be canceled when a single car passes over these positions.

Under sparse to moderate traffic flows, CA could not detect any change in the traffic parameters and could not even detect real incidents. So, it offers a *zero* false positive rate. As the traffic becomes denser, few false alarms would be detected because of the high densities. On the other hand, the deterministic technique detects incidents based on the difference between occupancies at different positions. Thus, it may mistakenly generates false alarms more often when density becomes large and some cars favors one lane over another.

Impact Of Detection Threshold

Figure 13 shows the impact of the detection threshold on the accident mean detection time under different probabilities of driver inputs. This figure shows that drivers input provided a great enhancement to the detection process. As expected, as the detection threshold increases, more time would be needed by a belt to detect the existence of an accident. As also expected, more driver input means simply less time to detect the accident when it happens.

Figures 13 also shows that our probabilistic technique outperform incident

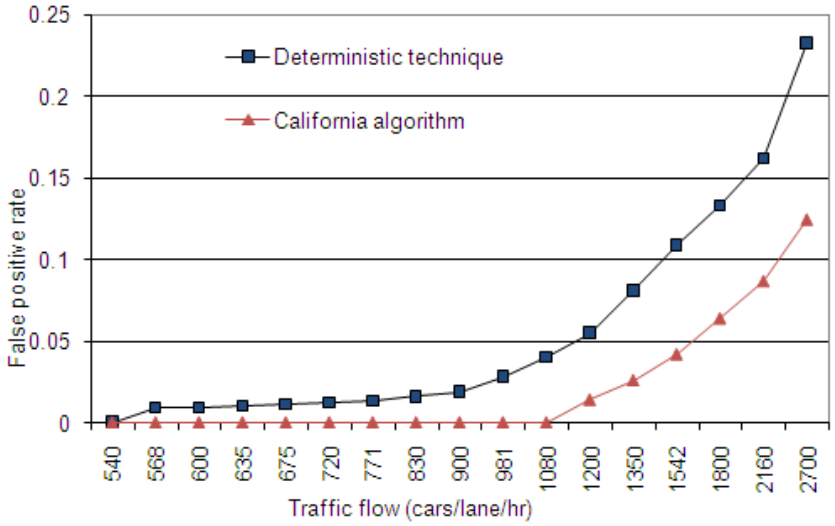

FIG. 12: Impact of traffic flow on false positive rate for accident detection

detection using cell phones only that has an average of 5 minutes detection time [12]. This is because our technique uses more pieces of evidence such as lane change in its detection process.

Figure 14 shows the impact of the detection threshold on the incident detection rate under different probabilities of driver inputs. It is not a surprise that detection rate decreases with the increase of detection threshold. However, even when the belt became very conservative, detection threshold of 0.9, it could still provide reasonable detection rate, for example more that 60% under driver input probability of only 0.6.

Impact Of Distance between Belts

Figure 15 shows the impact of the distance between belts on the mean detection time for our probabilistic technique under traffic flow values of 1200 and 1800 cars/hours/lane. This figure shows that the mean detection times increases slightly with the increase of distance between belts. This is expected because cars have to travel longer distance to report their EDRs to next belt. However, even when belts are installed with 5 km inter-spacing, detection time is still around 3 minutes. This shows that the NOTICE architecture is very efficient cost-wise.

Market Penetration

Figure 16 shows the effect of the percentage of vehicles equipped with EDRs and wireless devices on the mean detection time. For better illustration, we set the

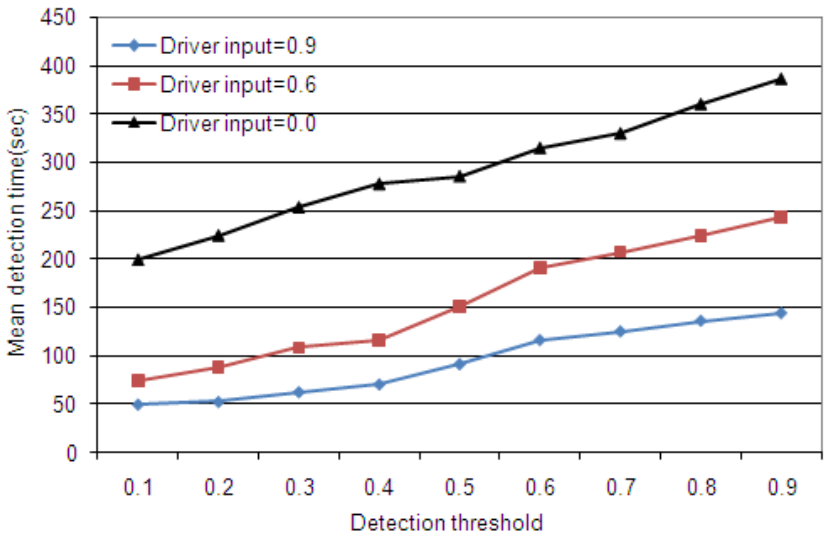

FIG. 13: Impact of detection threshold on mean detection time for accident detection

traffic flow to non dense traffic , 1200 cars/lane/hr, because otherwise the integrated technique will not be affected by the market penetration as it downgrades to pure California algorithm.

This figure shows that down to about 70% penetration, the probabilistic technique has almost the same performance as 100% case. However, for smaller penetration rates, a longer time would be needed to collect sufficient information from EDR equiped vehicles.

On the other hand, the integrated technique required less detection time than the the proposed technique because it is enhanced with the California based algorithm.

IV.4.3 Pothole Detection

This section is devoted to presenting simulation results and analysis for pothole detection. Unless otherwise specified, we assume that the pothole detection device detects a pothole with probability of 0.9.

Impact Of Traffic Flow

Figure 17 shows the impact of traffic flow on mean pothole detection time for different values of the detection threshold. For sparse traffic, a longer time is needed to accumulate more pieces of evidence until the probability of having a pothole reaches the specified detection threshold. However, as the traffic becomes

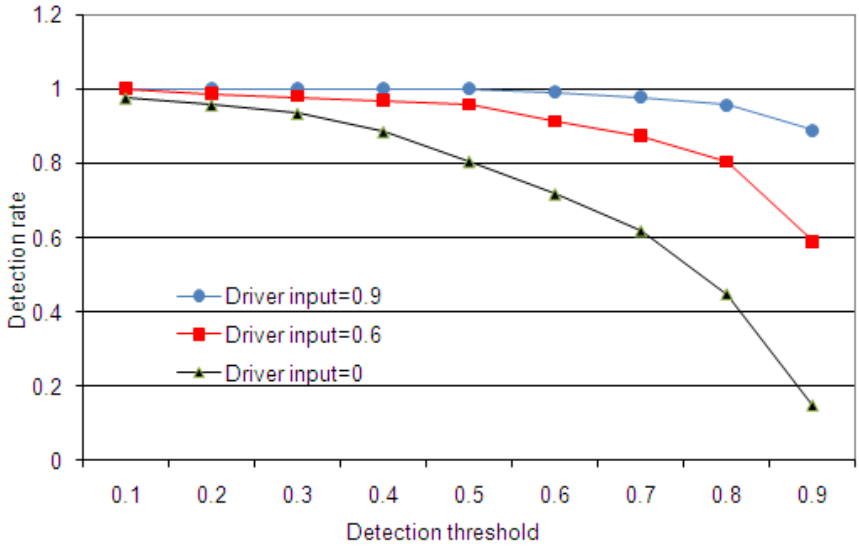

FIG. 14: Impact of detection threshold on detection rate for accident detection

more dense, more cars are available to report about the pothole resulting in shorter mean detection time. As an illustration, only about 40 seconds was needed to detect a pothole with detection threshold of 90% which is very reasonable time for that conservative detection threshold.

Impact Of Detection Threshold

Figure 18 shows the impact of the detection threshold on mean detection time under traffic flows of both 1200 and 2400 cars/hr/lane and 800 cars/hr/lane.

As expected, the smaller the detection threshold, the longer the pothole mean detection time as a belt needs to accumulate enough pieces of evidence. It is noteworthy to mention that the mean detection time does not exceed 2 minutes even under large values for detection thresholds. Hence, we can safely choose a high detection threshold, > 0.9, to avoid generating false positive alarms.

Impact Of Detection Duration

In this experiment, we study the effect of pothole detection duration on the false positive alarms. As we already mentioned before, it is important to re-initialize pothole probabilities after some detection duration time to avoid the detection of false potholes. If this detection duration is very short, shorter than the mean detection time, we may never detect any pothole. On the other hand, If it is very long, false alarms would be generated.

Figure 19 shows the impact of detection duration on the false positive rate

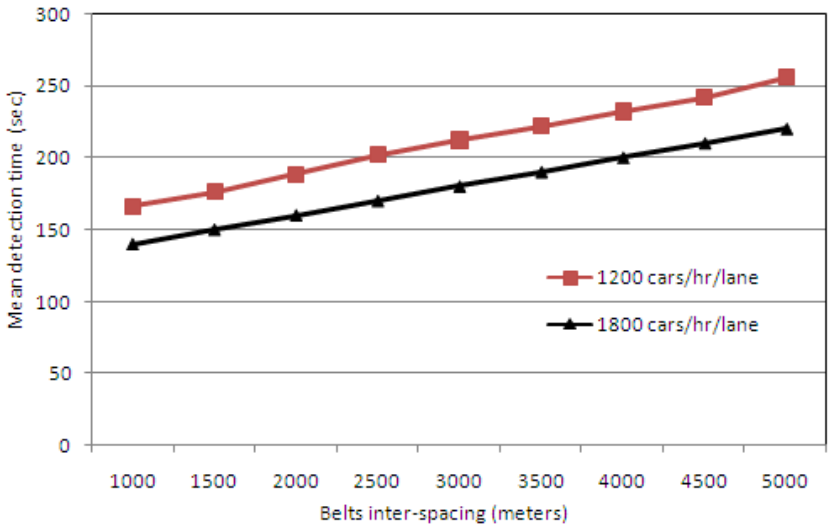

FIG. 15: Impact of belts spacing on mean detection time for accident detection

under different detection thresholds. This figure shows that zero false alarms are generated for detection duration up to 6 minutes. Joining this with the fact that pothole detection time is usually less than a minute, we can choose detection duration time to be around 5 minutes. Thus, we can use large detection thresholds and avoid generating false negative alarms.

Impact Of The Sensing Device Detection

Figure 20 shows the impact of the sensing detection probability, i.e. $P(Detection/Pothole)$, on the mean detection time under different traffic flow. This figure shows that even for $P(Detection/Pothole)$ around 0.5, the detection time would be less than a minute. As $P(Detection/Pothole)$ decreases, the mean detection time increases.

IV.5 SUMMARY

In this chapter, we presented our proposed techniques to enhance automatic incident detection in VANET. We started by proposing a deterministic technique where a belt collects EDR data from passing vehicles and maintains a table that stores occupancies for all positions between itself and the previous co-directional belt. These occupancies are used to detect possible blocking incidents that forced drivers to change lanes. A belt can identify blocking incidents by looking for low occupancies in one lane where the corresponding positions in adjacent lanes have

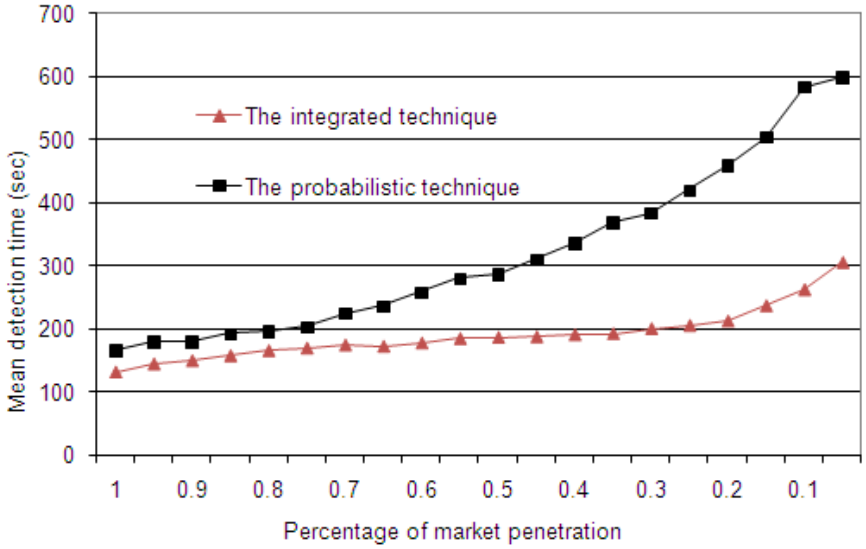

FIG. 16: Impact of market penetration on mean detection time for accident detection

higher occupancies. Then we presented a more generic Bayesian based probabilistic technique that incorporates more parameters in the detection process than just lane changes. Our probabilistic technique is capable of detecting both blocking incidents such as vehicle accidents and non blocking incidents such as potholes.

Our probabilistic technique also offers *zero* false positive alarms for most detection thresholds values, which makes it perfect and trustable to traffic operators. Also, our probabilistic technique added good enhancements to existing AID techniques in non dense traffic. However, as the traffic becomes denser, it is hard for vehicles to avoid the accident and continue to provide their EDR data to the next belt. For evaluation purposes, we integrated our probabilistic technique with the California Algorithm and showed that this will help detecting blocking incidents in all traffic conditions.

Simulation results showed that our probabilistic technique outperforms California Algorithm under non dense traffic in terms of mean detection time, detection rate and false positive rate. To the best of our knowledge, our probabilistic incident detection technique is the first VANET based approach capable of detecting both blocking and non blocking incidents.

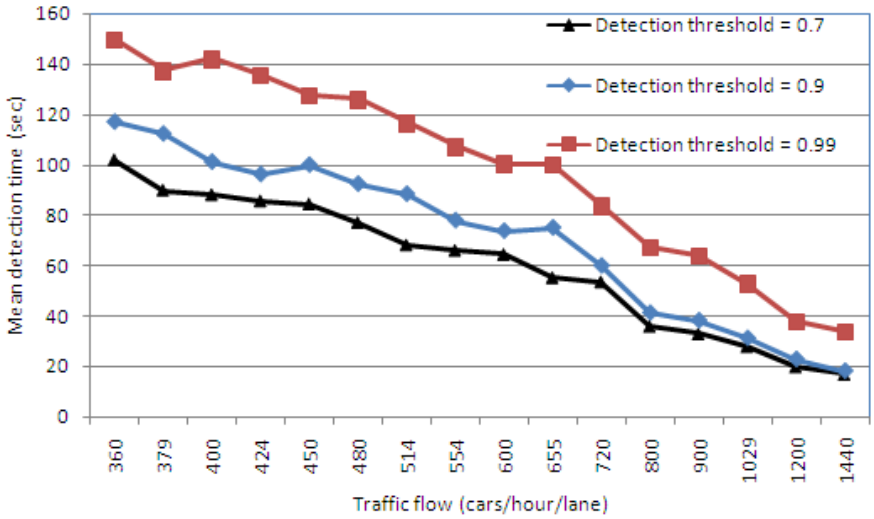

FIG. 17: Impact of traffic flow on mean detection time for pothole detection

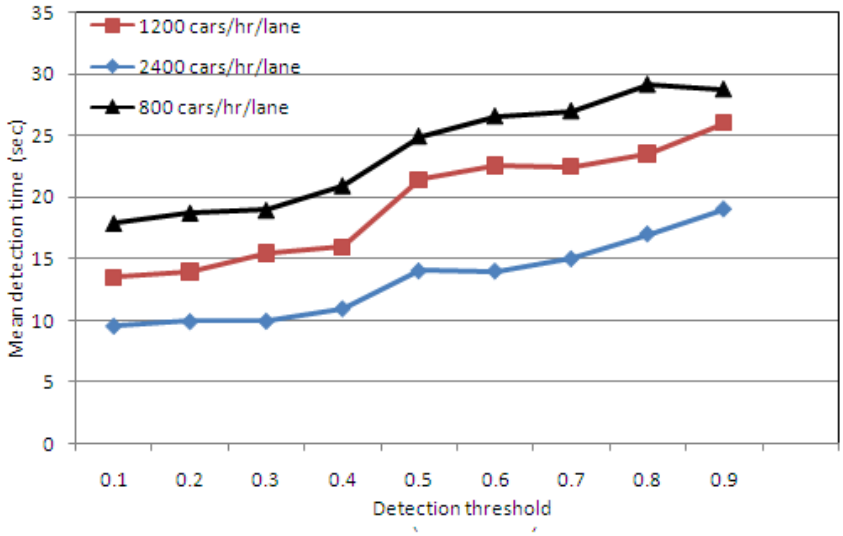

FIG. 18: Impact of detection threshold on mean detection time for pothole detection

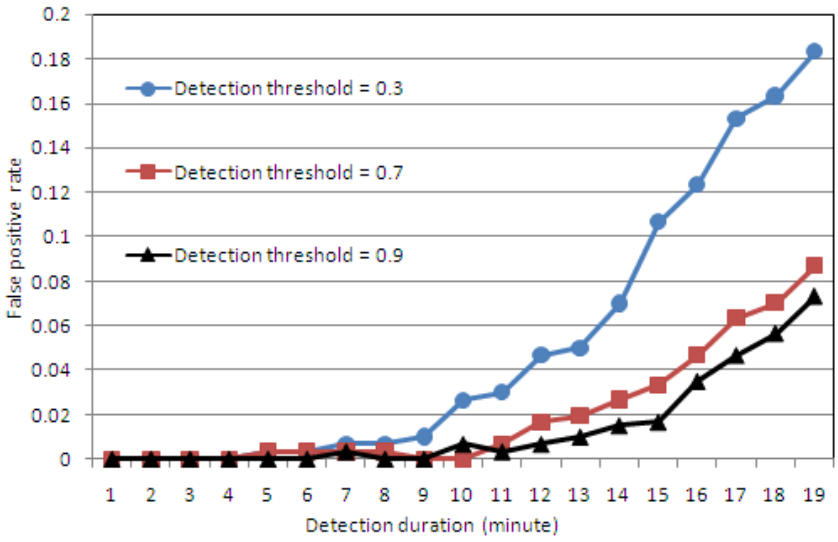

FIG. 19: Impact of detection duration on false positive rate for pothole detection

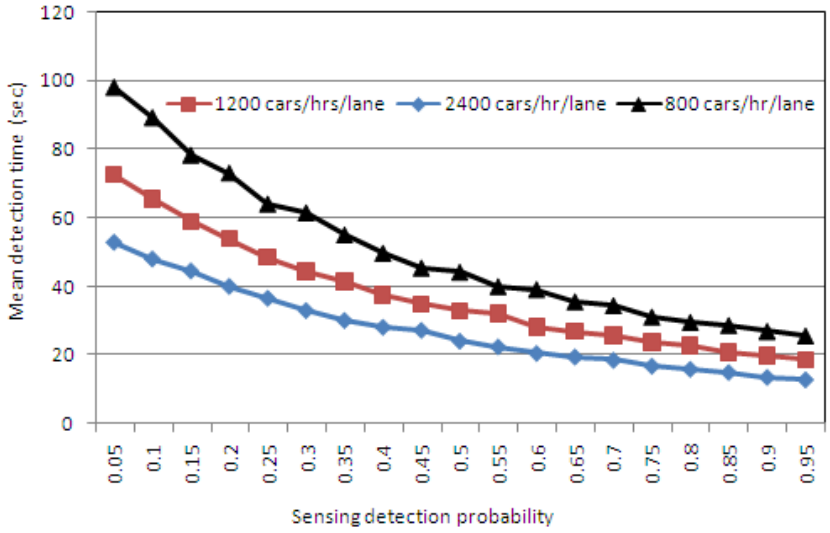

FIG. 20: Impact of sensing detection probability on mean detection time for pothole detection

CHAPTER V

TRAFFIC ANALYSIS

After detecting an incident, it is important to efficiently disseminate information about it to alert approaching vehicles on the road. In this chapter, we start by presenting traffic analysis while our data dissemination techniques are presented in Chapter VI.

Empirical evidence, accumulated over time, has shown that under many highway scenarios VANETs tend to be disconnected, consisting of a collection of disjoint clusters. This chapter is devoted to providing an analytical explanation of this result, thus confirming the findings of [36] [34] [61] [62]. We show that this phenomenon is present even in relatively dense traffic and provide an analytical expression of the expected size of a cluster, as a function of traffic density and communication range. We also show that the cluster size is quite stable and easy to maintain [35]. Actually, this finding was very important in developing our data dissemination approaches.

V.1 EVALUATING THE PROBABILITY OF LARGE GAPS IN CO-DIRECTIONAL TRAFFIC

While traffic displays diverse spatio-temporal patterns, several workers have pointed out that an *instantaneous* snapshot of a steady free flow of uncongested traffic can be approximated by uniform car density (measured in cars per kilometer), which translates into a uniform vehicular distribution [63–65]. It is very important here to mention that we are neither assume that vehicle's arrival rate nor inter-vehicle spacing is uniform. We instead assume that if we took a picture of traffic at a certain moment, vehicles would appear to be uniformly distributed as some vehicles have just entered the highway, others are about to take an exit while some others are accelerating/decelerating.

The goal of this section is to provide an answer to the following natural question: Given that m vehicles are deployed uniformly at random in a single lane of traffic of one kilometer and given that dependable radio communications between vehicles require a *maximum* inter-vehicle distance of d meters, what is the probability that there is end-to-end radio connectivity between the m vehicles? This question is fundamental. We prove that the number of vehicles per kilometer must be at least 16 in order to have a better than even chance for connectivity; it takes about 25 vehicles per kilometer for end-to-end connectivity to be present

with 90% probability.

Returning to our problem, we model the situation as follows: the m vehicles determine $m-1$ *distinguishable* bins (inter-vehicle spaces), enumerated in left-to-right order as $B_1, B_2, \ldots B_{m-1}$. The number of distinguishable ways in which the n indistinguishable balls (unit inter-vehicle spaces) can be distributed into the $m-1$ bins is easily seen to be $\binom{m+n-2}{n} = \binom{m+n-2}{m-2}$. To see that this is the case, observe that the $m-1$ bins involve m separators and that we can lay down the balls and bins in a linear sequence flanked on both sides by a separator. The problem now is that of selecting n places for the balls out of a total of $n+m-2$ places available. The conclusion follows.

Now suppose that we want a *given* bin to contain k, $(0 \leq k \leq n)$, balls. This amounts to distributing k balls into one bin and $n-k$ balls into the remaining $m-2$ bins. Reasoning as above, the number of distinguishable ways in which this can be achieved is $\binom{(n-k)+(m-3)}{n-k} = \binom{n+m-k-3}{n-k}$. As a consequence, the probability p_k, $(0 \leq k \leq n)$, of the event that a given bin contains *exactly* k balls is

$$p_k = \binom{n+m-k-3}{n-k}\binom{m+n-2}{n}^{-1}. \tag{15}$$

To show that the p_ks are a valid probability distribution, we need to prove that $\sum_{k=0}^{n} p_k = 1$. This, in turn, amounts to showing that $\sum_{k=0}^{n}\binom{(n-k)+(m-3)}{n-k} = \binom{m+n-2}{n}$. Indeed, recalling that for integers r and n,

$$\sum_{t\leq n}\binom{r+t}{t} = \binom{r+n+1}{n} \tag{16}$$

(see [66], (5.9) p.159), we write

$$\begin{aligned}
\sum_{k=0}^{n} p_k &= \binom{m+n-2}{n}^{-1}\sum_{k=0}^{n}\binom{(n-k)+(m-3)}{n-k} \\
&= \binom{m+n-2}{n}^{-1}\sum_{i\leq n}\binom{(m-3)+i}{i} \\
&= \binom{m+n-2}{n}^{-1}\binom{(m-3)+n+1}{n} \text{ [by (16)]} \\
&= \binom{m+n-2}{n}^{-1}\binom{m+n-2}{n} \\
&= 1,
\end{aligned}$$

as desired.

In our setup, two neighboring vehicles become disconnected if the bin corresponding to the distance between them accumulates at least $d+1$ balls, where d corresponds to the maximum effective transmission range. Let A_i, $(1 \leq i \leq m-1)$, be the probability that a generic bin B_i contains *at least* $d+1$ balls.

Lemma V.1.1 *For all i, $(1 \leq i \leq m-1)$,*

$$\Pr[A_i] = \binom{m+n-(d+1)-2}{m-2}\binom{m+n-2}{n}^{-1}$$

.

Proof We find it convenient to compute the probability of the complementary event $\overline{A}_i$. By (15) and (16) we can write

$$\begin{aligned}
\Pr[\overline{A}_i] &= \binom{m+n-2}{n}^{-1}\sum_{j=0}^{d}\binom{(m-3)+(n-j)}{n-j} \\
&= \binom{m+n-2}{n}^{-1}\sum_{t=n-d}^{n}\binom{(m-3)+t}{t} \\
&= \binom{m+n-2}{n}^{-1}\sum_{t=0}^{n}\binom{(m-3)+t}{t} \\
&- \binom{m+n-2}{n}^{-1}\sum_{t=0}^{n-d-1}\binom{(m-3)+t}{t} \\
&= 1-\binom{m+n-2}{n}^{-1}\binom{m+n-d-3}{m-2}
\end{aligned}$$

Thus, $\Pr[A_i] = 1-\Pr[\overline{A}_i] = \binom{m+n-2}{n}^{-1}\binom{m+n-d-3}{m-2}$, and the proof of the lemma is complete.

Let A be the event that there is *no* end-to-end connectivity between the m vehicles. Clearly, $A = \cup_{i=1}^{m-1} A_i$. Since the A_i's are not independent, the principle of *inclusion-exclusion* implies that $\Pr[A] = \sum_{i=1}^{m-1} Pr[A_i] - \sum_{1\leq i<j\leq m-1}\Pr[A_i \cap A_j] + \cdots + (-1)^i \sum_{1\leq j_1<j_2<\ldots<j_i\leq m-1}\Pr[A_{j_1}\cap A_{j_2}\cap\cdots\cap A_{j_i}] + \cdots$

Lemma V.1.2 *For all i,j, $(1 \leq i < j \leq m-1)$, $\sum_{1\leq i<j\leq m-1}\Pr[A_i\cap A_j] = \binom{m-1}{2}\binom{m+n-2(d+1)-2}{m-2}\binom{m+n-2}{n}^{-1}$.*

Proof We provide a purely combinatorial proof. First, to obtain $\Pr[A_i \cap A_j]$, observe that the number of distinguishable arrangements in which bins i and j contain at least $d+1$ balls is obtained by first placing $d+1$ balls in bins i and j and then by distributing the remaining $n-2(d+1)$ balls uniformly at random in *all* the $m-1$ bins. This can be done in $\binom{m+n-2(d+1)-2}{n-(d+1)} = \binom{m+n-2(d+1)-2}{m-2}$ distinct ways. Since there are $\binom{m-1}{2}$ distinct ways of choosing i and j subject to $(1 \leq i < j \leq m-1)$, the conclusion follows.

Lemma V.1.3 *For all $1 \leq j_1 < j_2 < \ldots < j_i \leq m-1$, $\sum_{1\leq j_1<j_2<\ldots<j_i\leq m-1}\Pr[A_{j_1}\cap A_{j_2}\cap\cdots\cap A_{j_i}] = \binom{m-1}{i}\binom{m+n-i(d+1)-2}{m-2}\binom{m+n-2}{n}^{-1}$.*

Proof This follows from Lemma V.1.2 by a simple inductive argument.

Theorem V.1.4

$$\Pr[A] = \frac{\sum_{i=1}^{m-1}(-1)^{i+1}\binom{m-1}{i}\binom{m+n-i(d+1)-2}{m-2}}{\binom{m+n-2}{n}}. \tag{17}$$

Proof This follows directly from Lemmas V.1.1, V.1.2 and V.1.3, combined.

Although a closed form for $\Pr[A]$ is hard to obtain, we have compared the results obtained by evaluating (17) for various values of m with those yielded by averaging 10 million simulations of an experiment that consists in generating uniformly at random m points in the unit interval and checking whether any two neighbors are separated by more than 0.2. As illustrated in Figure 21, our simulation results are virtually indistinguishable from the analytical result.

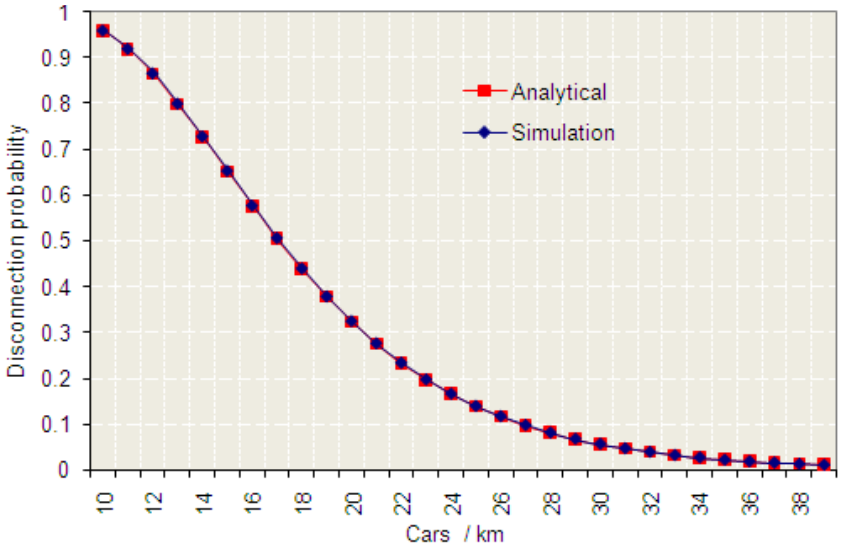

FIG. 21: Disconnection probability

One can interpret (17) as follows. Imagine sliding a 1 km window down a highway with one lane of traffic in each direction. If the window contains m co-directional vehicles, then the probability that there is no end-to-end connectivity between them is precisely $\Pr[A]$ in (17). For example, should there be 12 co-directional vehicles in the window, the probability of no end-to-end connectivity between them is about 86%. Naturally, the probability *decreases* with the number of co-directional lanes of traffic in each direction.

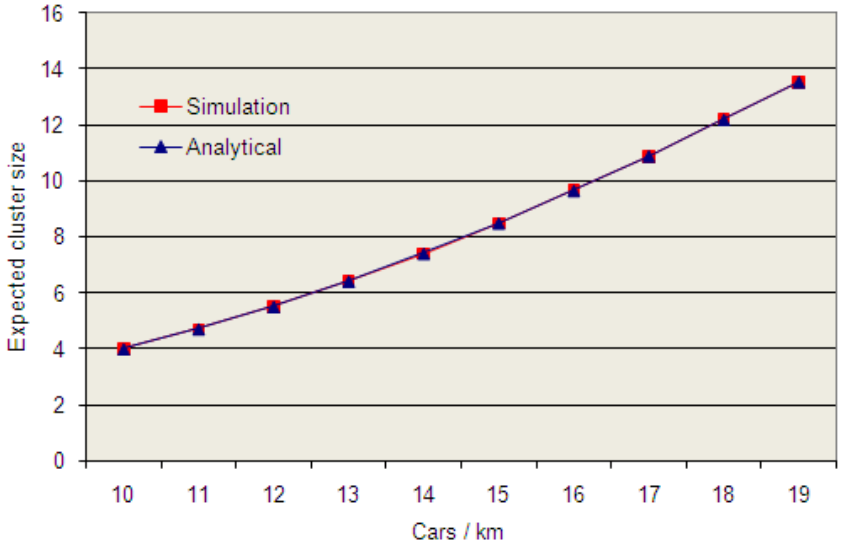

FIG. 22: Cluster size

V.2 EVALUATING THE EXPECTED SIZE OF A CLUSTER

Since, as we saw, co-directional traffic is *inherently* partitioned into clusters, an interesting question is to estimate the expected size of a cluster. The goal of this section is to provide an answer to this natural question. For this purpose, we inherit the notation and terminology of Section V.1.

Theorem V.2.1 *The expected size of a cluster is*

$$E[cluster\_size] = \frac{m \cdot \binom{m+n-2}{n}}{\binom{m+n-2}{n} + (m-1) \cdot \binom{m+n-d-3}{n}}. \tag{18}$$

Proof As we saw, the probability p that a given bin contains at *least* $d+1$ balls is $p = \binom{m+n-d-3}{m-2}\binom{m+n-2}{n}^{-1}$. Let X be the random variable that counts the number of "gaps" (i.e., the number of bins containing at least $d+1$ balls). Since X is binomial, the expected value $E[X]$ of X is

$$\begin{aligned} E[X] &= (m-1) \cdot p \\ &= (m-1) \cdot \binom{m+n-d-3}{m-2}\binom{m+n-2}{n}^{-1} \end{aligned} \tag{19}$$

Once we have the expected number of gaps in co-directional traffic, the expected number of clusters becomes $1 + E[X] = 1 + (m-1) \cdot \binom{m+n-d-3}{m-2}\binom{m+n-2}{n}^{-1}$. Thus, the expected size of a cluster is

$$E[cluster\_size] = \frac{m \cdot \binom{m+n-2}{n}}{\binom{m+n-2}{n} + (m-1) \cdot \binom{m+n-d-3}{n}},$$

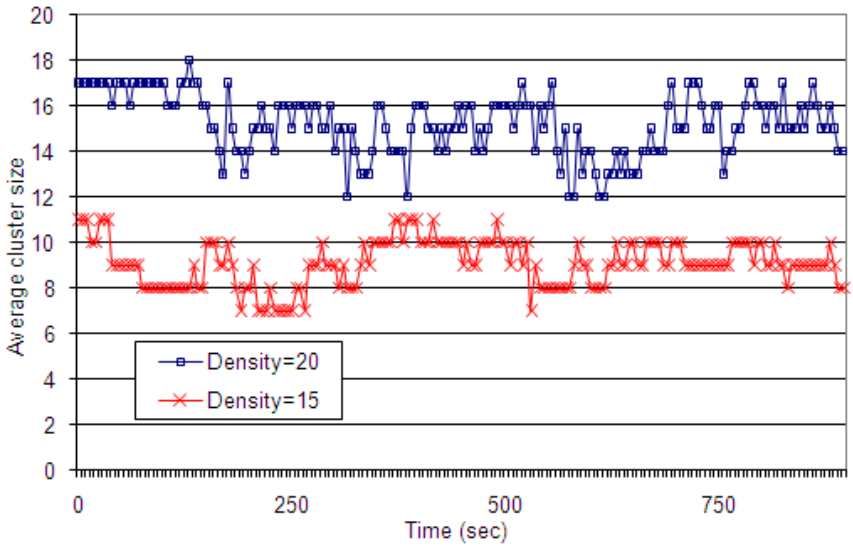

FIG. 23: Cluster stability

completing the proof of the theorem.

Figure 22 provides a side-by-side comparison of the expected cluster size predicted by (18) and the value obtained by simulation. As an illustration, imagine a two-lane road of 1Km and 10 vehicles distributed uniformly at random per lane of traffic. By virtue of (19) we expect to see about 2.47 clusters; by (18), we expect a cluster to contain between 4 and 5 vehicles.

V.3 CLUSTER STABILITY

As already mentioned, since co-directional vehicles move at a small relative speed with respect to each other, we expect clusters to be quite stable and easy to maintain. We defer discussing cluster maintenance until presenting our data dissemination protocols. In order to get a better understanding of co-directional cluster dynamics we have simulated a stretch of highway with two traffic flows of 15 and 20 vehicles/km. In both cases, the difference between the highest and lowest speed is 15km/hour. Figure 23 illustrates, side by side, the average cluster sizes over 15 minutes of simulation time. In both cases the simulation revealed that in spite of mobility, the expected cluster size is remarkable close to the theoretical prediction of 10 and 15 vehicles, respectively. Incidentally, this is also indirect validation of the uniformity assumption.

V.4 SUMMARY

In this chapter, we presented an analysis for vehicles traffic proving that disconnection in VANET is highly probable even in relatively dense traffic.

We also provided a formula to compute the expected cluster size in such a disconnected environment. We showed that cluster size is relatively small, around 16 vehicles per cluster assuming 19 vehicles/km. Finally, We showed that the cluster size is quite stable and easy to maintain.

To the best of our knowledge, we are the first in VANET community to prove analytically that disconnection is the norm rather than the exceptions in VANETs and provide these analytical results.

CHAPTER VI

DATA DISSEMINATION

This chapter is devoted to presenting our data dissemination techniques. We start by presenting a clustering technique that will be used during data dissemination. Then we present our first data dissemination approach for undivided roads [35, 67]. Section VI.3 presents our proposed data dissemination technique for divided highways. Finally, we answer one of the very interesting questions in incident notification which is how far from the incident location should the notification be sent. Throughout this section, for generalization, we assume that the source and destination are cars rather than belts. However, disseminating packets between belts is a special case by assuming that the source and destination cars have zero speeds.

VI.1 CLUSTERING TECHNIQUE

Section V shows that traffic is inherently partitioned into clusters disconnected from each other where vehicles enjoy end-to-end connectivity within each cluster. We have also computed the expected cluster size that was found to be relatively small and quit stable. All of these findings have motivated us to propose a clustering technique among cars on the road.

This section is devoted to discussing this clustering technique which is the core of our proposed data dissemination technique. We start by describing the basic format of beacons transmitted by cars. Then, we describe the clustering process and we show that it has low overhead in terms of bandwidth usage.

VI.1.1 Cluster Management Beacons

As stated in our assumptions in Chapter III, we assume cars to be GPS-enabled and to communicate using DSRC [68]. Being GPS-enabled, cars know their geographic position and are synchronized in time. As mandated by DSRC, every 300 ms each vehicle sends a beacon with a range of about 200–300 m [3]. Each beacon contains information that allows vehicles to handshake and synchronize. We are using these beacons for cluster formation and cluster maintenance as well. Mindful of their original intent we shall, nonetheless, refer to these beacons as *Cluster Management Beacons*, (CMB, for short). So, it is noteworthy here to mention that the clustering process introduces almost *zero* overhead because these beacons are transmitted anyway for many other purposes such as neighbor discovery. Figure

24 shows the proposed format of a CMB obtained by taking advantage of *unused* fields of the standard IEEE 802.11 beacon. Each beacon transmitted by a car x contains the following information.

- *ts*: this field contains a timestamp with a length of 8 bytes,
- *bi*: this field contains beacon interval with a length of 2 bytes, typical value is 300 ms.
- *pos*: this field contains x's position obtained by the GPS . This position may be encoded in 12 bytes [69].
- *sp*: this field contains x's speed measured as miles per hour and is encoded in 2 bytes. This value may be negative to determine direction
- *hv*: this field contains the position of the header car of x's cluster and is encoded in 12 bytes.
- *tv*: this field contains the position of the tail car of x's cluster and is encoded in 12 bytes.
- *hvs*: this field contains the speed of the header car of x's cluster and is encoded in 2 bytes.
- *tvs*:this field contains the speed of the tail car of x's cluster and is encoded in 2 bytes. Finally, there is a 4-byte unused filed. Thus, the CMB beacon size is 56 bytes.

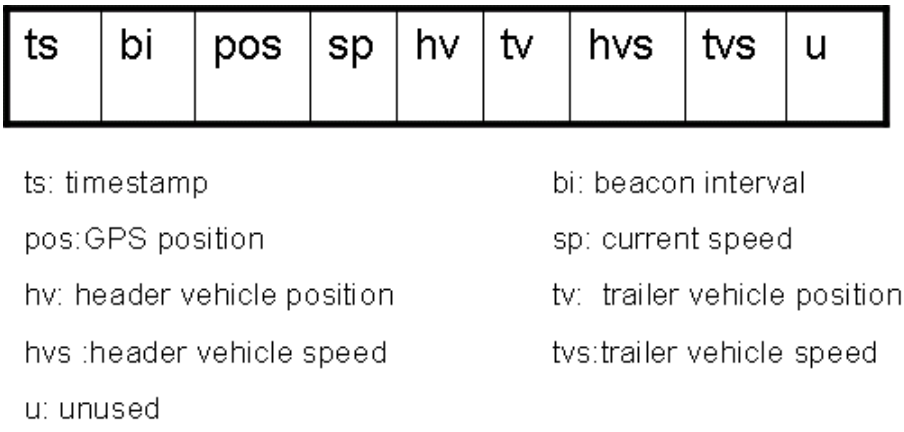

FIG. 24: Illustrating the layout of a CMB

VI.1.2 Cluster Formation

To begin, the task of clustering can be performed as follows: each car that has not received, within a certain time-out interval, a beacon from a co-directional car in front of it, declares itself header of the cluster and sends this information in the next CMB to the cars behind it. The message will be then multi-hopped, using CMB beacons throughout the cluster. Note that this information is piggybacked in regular beacons transmitted by cars. The CMB contains, in addition to the identity of the header, its geographic position, direction of movement and speed. (Suffice it to say that direction is immediately available if speed if kept as a signed integer.) Every co-directional car that receives such a CMB understands that it belongs to the cluster named after the header. In a symmetric way, the last car in the cluster informs, by virtue of a CMB, all the other cars in its own cluster of its geographic position and speed. The *head* and *tail* vehicles in a cluster, maintained proactively as described, play a special role in our proposed dissemination techniques and will be denoted, respectively, by $h(\cdot)$ and $t(\cdot)$. As we showed before, clusters are expected to be quite *stable* and easy to maintain. This is because co-directional vehicles move at a small relative speed with respect to each other. Figure 25 shows different clusters on a two-lane highway.

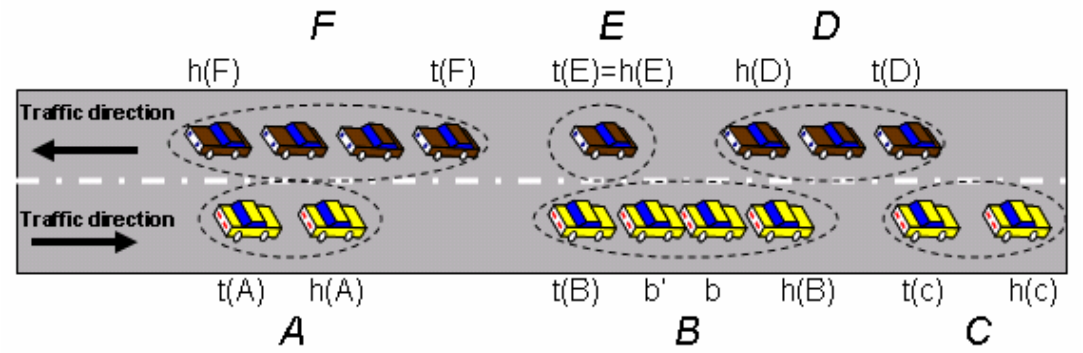

FIG. 25: Illustrating vehicles clusters on a two-lane highway

VI.1.3 Maintaining Cluster-Related Information

A generic vehicle x in a cluster X maintains *proactively* information about the cluster to which it belongs as well as *overlapping* clusters, that is, clusters in the oncoming direction with which a node in x's cluster is in direct radio contact. By propagating this information during cluster formation and maintenance, every car in the cluster acquires information that allows it to make adequate routing decisions. This ability to make routing decisions is of key importance in our

dissemination techniques. We note that all of the bindings described below are soft and are subject to time-out. Specifically, x maintains:

- A first record $(x.cid, x.pos, x.sp, x.X, x.O)$ where
  - $x.cid$ is the identity of the cluster to which x belongs,
  - $x.pos$ is the current position of x obtained from the on-board GPS device,
  - $x.sp$, is the current speed obtained from the on-board speedometer,
  - $x.X$, the sets of all neighboring cars in x's cluster that can be reached from x in one hop, these can be easily maintained from their beacons.
  - $x.O$, the sets of all neighboring cars in overlapping clusters that can be reached from x in one hop, assuming undivided highway.
- A second record, also maintained proactively as described in Section VI.1.2, contains information about head and tails of $x.X$.
- A third record contains a flag to determine whether there is any overlapping cluster in advance or not and the expected time to lose such overlap. This information may be maintained as follow. Once car y, in cluster Y, detects that $y.O$ is not empty, it can piggyback such information in its next CMB beacon, in the *unused* field. By exchanging beacons, all cars in the $y.Y$ will be aware about such overlap and its expected duration.

It is important to realize that, by virtue of Theorem V.2.1, the cluster size is bounded (see also Figure 22). Moreover, since co-directional clusters tend to be *stable*, and the underling topology of clusters *linear*, maintaining these records proactively is not a problem and we do not run into scalability problems.

Also, we will show shortly that these beacons introduce a small overhead in terms of bandwidth wasting while on the other hand they save much bandwidth and time in packets propagation. As clusters in opposite directions "meet", they exchange routing information. This allows the cars in each cluster to update their routing tables. Since the bindings are soft, as these clusters drift away from each others, the information is no longer reinforced and will be removed.

VI.1.4 Clustering Overhead

Despite being sent anyway for general purposes like handshaking and synchronization, it is helpful to show the overhead of beacon transmissions. We have

performed two experiments to measure both the time needed for cluster maintenance as well as bandwidth used in this maintenance.

In both experiments, we deployed cars at random but in a single cluster, by ensuring that interspacing between each two consecutive cars is less than the transmission range . We then, started to compute how much time was needed by these cars so that each car in the cluster would have correct information about its neighbors and its header car.

Figure 26 shows the impact of cluster size on the maintenance time, time until all cars in a cluster know about its head and tail. As shown in the figure, for a relatively large cluster size of 50 cars that may cover a distance over one kilometer, less than 5 sec was needed for all cars to learn about head/tail and all 1-hop neighbors in their cluster. Joining this result with the fact that clusters are relatively small and quite stable, as we showed in Section V, we claim that cars will have up-to-the-second information about their clusters most of the time.

A natural question that may arise here is how much bandwidth is being wasted in the clustering process? To answer this question, we measured the percentage of wasted bandwidth in the maintenance process. Figure 27 shows the impact of cluster size on the wasted bandwidth during maintenance. As shown in the figure, a very small amount of bandwidth was being wasted. For example, for up to 50 cars in a cluster, only about 4% of the bandwidth has been wasted in the maintenance process. Even in large clusters of 100 cars that would cover a distance more than 10 km, assuming an average of 100 meters interspacing, only 8% of the bandwidth would be wasted. Thus, clustering in most cases is a lightweight process that consumes little resources. It is important here to mention that in a very congested traffic, any greedy packet forwarding protocol would work fine and the clustering technique may be stopped.

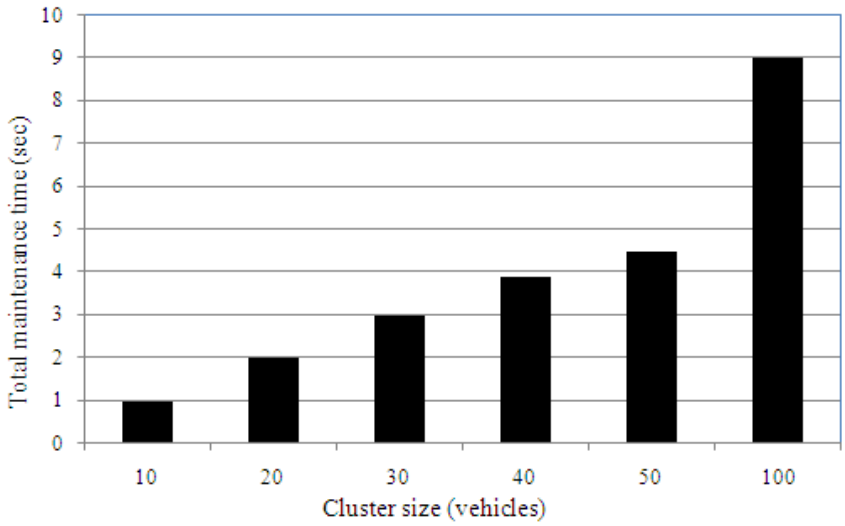

FIG. 26: Impact of cluster size on cluster maintenance time

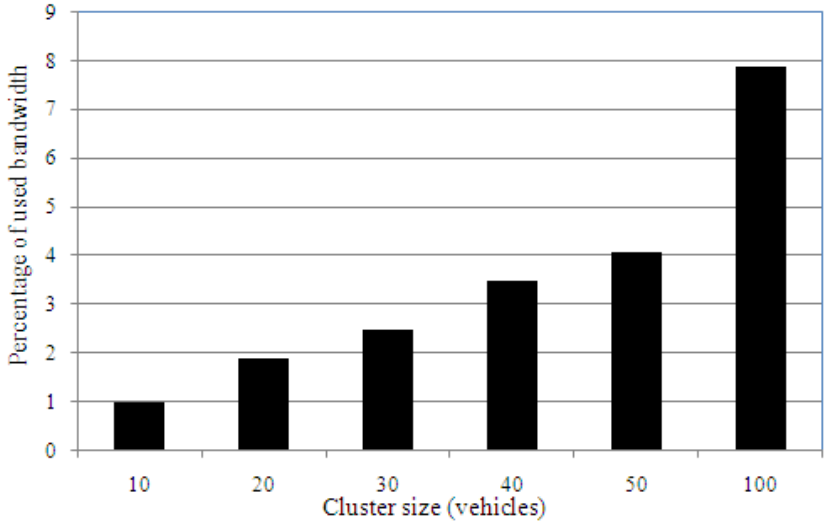

FIG. 27: Impact of cluster size on percentage of used bandwidth

VI.2 OPERA: OPPORTUNISTIC PACKET RELAYING IN DISCONNECTED VEHICULAR AD HOC NETWORKS

Now, we are ready to describe our first data dissemination approach namely OPERA (Opportunistic Packet Relaying in Disconnected Vehicular Ad Hoc Networks) [35, 67]. In OPERA, we assume an undivided highway where cars on opposite directions may successfully communicate with each other.

VI.2.1 Motivation Example

Referring to Figure 28, suppose that car a wishes to deliver a packet to car g. The simplest and most straightforward strategy, which does not require routing at all, is for car a to wait until, eventually, it meets car g and can deliver the packet directly, as illustrated in Figure 28(c). In such a scenario we say that car a acts as a *data mule* for car g [70]. Data mules guarantee delivery and are quite effective if the geographic distance between the two cars is modest and the traffic is sparse. Actually, in very sparse traffic, data mules may be the only workable strategy. In some cases, however, one can do better than data mules.

The majority of VANET routing protocols assume end-to-end connectivity of co-directional traffic with some exceptions like DPP [34] and CAR [26] that observed that co-directional cars are grouped into clusters that are disconnected from each other.

However, All of these protocols would work as follow, if they could even work in this disconnected scenario. Consider the example in Figure 28(a). Car a detects that the only vehicle in range is car b and sends the packet to it. Since the cars b, c, d, and e form a cluster, the packet sent to car b, will be multi-hopped, in the obvious way, to car e. Car e will keep the packet for a while until it meets car a to which it will upload the packet as shown in Figure 28(b). Thus, the packet that has originated at car a ends up at car a, again. Clearly, such a situation is most undesirable since a sizable amount of resources has been consumed (in signaling and routing the packet) and has achieved nothing. In this case valuable bandwidth was wasted in routing from a to e (along the chain b, c, d, e) and then back to a. It would have been much better for car a not to send the packet to b at all.

In addition to wasting bandwidth, most of the existing protocols suffer from many other disadvantages like routing loops that may exist in the previous example if car e misses the connection with car a, it may send the packet to any car behind a that would in turn be routed again to a!.

DPP avoids routing loops by applying the idea of custody, that is car a will not release the packet until it gets confirmation from next co-directional cluster about receiving the packet through the oncoming cluster. Referring again to Figure 28, when car a sends the packet initially to car b and received no confirmation, a has to resend the packet again to car c or d after some timeout. By doing that, more bandwidth is being wasted in addition to wasting time by waiting before re-sending the packet again. This waiting time may result in losing a possible overlap between the clusters.

In OPERA, a packet may "hop" between clusters or cars moving in opposite lanes until, eventually, it reaches its destination. In this sense, OPERA is actually a *hybrid* protocol as it alternates between applying proactive routing and data mules in a clever way to avoid delay or bandwidth wasting.

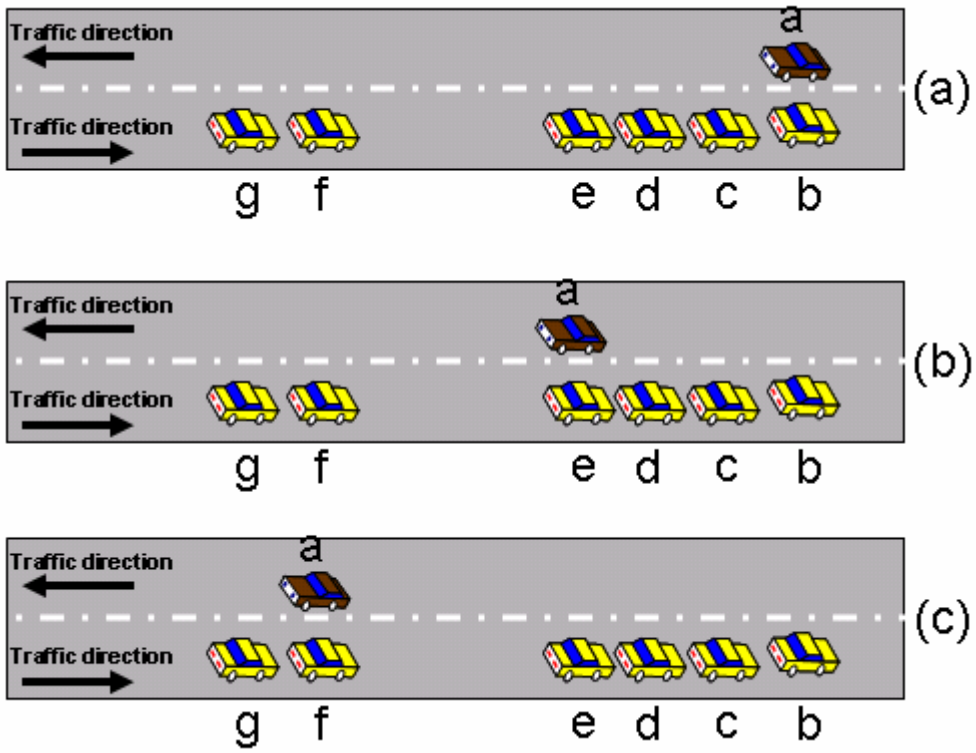

FIG. 28: OPERA: motivation example

VI.2.2 Clustering

The previous example suggests that it is important for a vehicle to have some idea about the local topology of cars in adjacent clusters as it was better for car a to know about the disconnection of car e. Thus, OPERA uses the clustering technique described in Section VI.1 in order to educate vehicles about the local topology of the network.

VI.2.3 The Baseline Algorithm

The metric we adopt for assessing the performance of OPERA is *delivery time* and, for the same delivery time, *hop count*. The main reason for this choice is that pure *data mule* achieves optimal hop-count, in fact, a hop count of 1, (see Figure 28(c)), at the expense of delivery time. An immediate corollary of this observation is that in order to optimize delivery time, packets that cannot be routed to an

overlapping cluster will be routed, internally, to $h(.)$, the first car in the cluster. Nonetheless, to give the reader the full generality of the situation, in the Baseline Algorithm discussed below, we assume that an overlapping cluster is available and that the packet to be routed is stored by an arbitrary car, not necessarily the first car in the cluster.

The *Baseline Algorithm* that we discuss in this section is the workhorse of OPERA.

$Baseline(a, A, x, X)$ assumes that some vehicle a in cluster A has a packet to relay to a vehicle x in cluster X such that the following conditions are satisfied:

- a and x are in opposite lanes of traffic, and
- $A \cup X$ is a connected graph.

Refer to Figure 29 for an illustration.

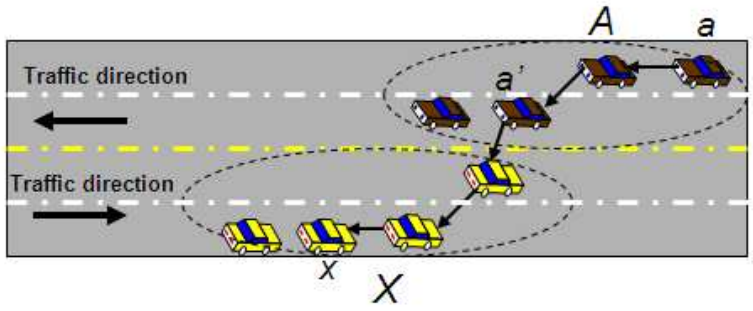

FIG. 29: OPERA: illustrating the Baseline Algorithm

Two nodes are *within range* of each other if their radio connection lasts longer than the packet transmission time. Since the cars know their location, this is easy to determine. We assume, without loss of generality, that x is to the left of a.

$Baseline(a, A, x, X)$ works as follows. If x is within range of a, the packet is delivered directly. Otherwise, by consulting its routing table, list of all 1-hop neighbors, a forwards the packet to a car (in either A or X) that minimizes the hop-count to x and start the baseline again.

Lemma VI.2.1 *Assuming correct cluster-related information,* $Baseline(a, A, x, X)$ *correctly relays the packet from* a *to* x *along a shortest path in* $A \cup X$.

Proof The correctness and the optimality of the Baseline algorithm follow directly by the choice of the next hop, one that minimized the hop-count to x.

VI.2.4 The General Algorithm

The General Algorithm, $General(a, A, x, X)$, assumes that vehicle a in cluster A has a packet to relay to some oncoming vehicle x in (known) cluster X but that the graph $A \cup X$ is not connected. In the terminology of Section VI.1.3, let B be the closest co-directional cluster to X that overlaps with A. Further, let D, if any, be the leftmost cluster among the clusters that overlap with B. We refer the reader to Figure 30 for an illustration. Specifically, in Figure 30(a) cluster B overlaps only cluster A and so, $A = D$; in Figure 30 (b) cluster B overlaps three clusters, namely D, C and A, in left-to-right order.

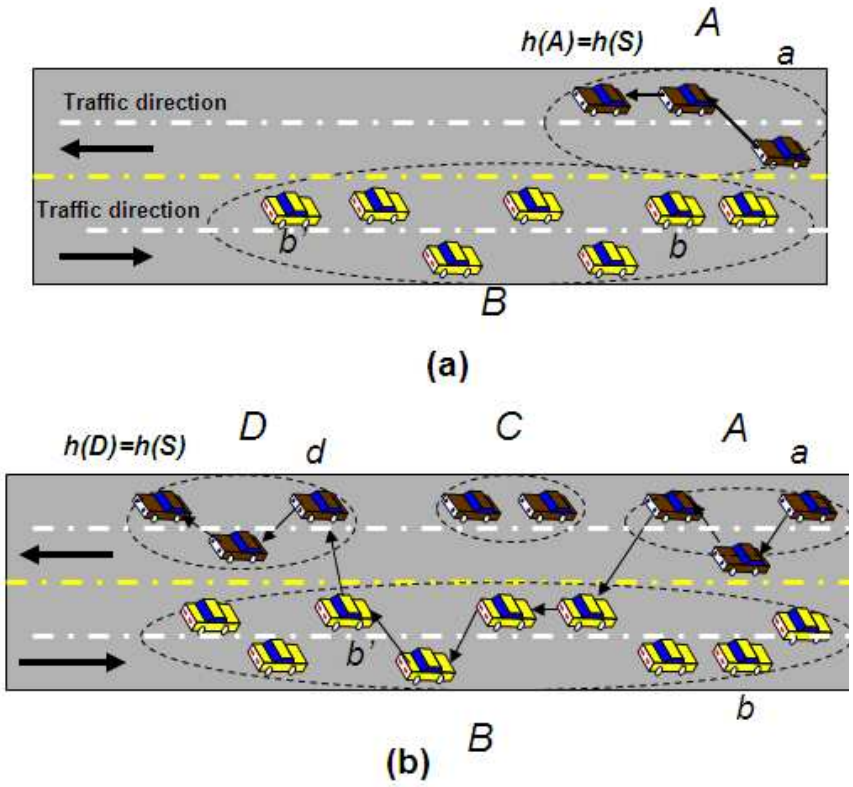

FIG. 30: OPERA: illustrating the General Algorithm

Let S be the graph induced by A, B and all the clusters overlapping B. It is clear that S is connected and, therefore, one can route the packet held by car a to the head of the leftmost cluster co-directional with A in S. By abusing notation a little, we let $h(S)$ denote the head of the leftmost cluster that overlaps B. Referring to Figure 30, $h(S)$ is either the head of cluster A in Figure 30 (a) or the head of cluster D in Figure 30(b). This routing decision is justified by our motivating example and the discussion in Section VI.2.3.

The routing itself can be performed by the following greedy approach. First, if $h(S) = h(A)$ then, clearly, all that needs to be done is to route the packet

to the head of a's cluster. If, however, $h(S) = h(D)$, for the leftmost cluster D co-directional and overlapping B, then the packet is routed using the Baseline Algorithm to one of cars, say, b' that has connectivity to some car in cluster D and then by the Baseline Algorithm again, to $h(S)$. The algorithm now continues recursively, until the packet is delivered to x. We emphasize that car a does not know and actually need not to know about clusters C and D: all it knows is that cluster B overlaps with some clusters ahead of its own. So, car a *formally* does the following. *If it realizes that the oncoming cluster has some overlapping with another co-directional cluster on the road, a can know that from beacons broadcasted by 1-hop cars in B, then the baseline algorithm is applied until the packet reaches that next cluster. Otherwise, it routes the packet to the header car.* Note that the above algorithm is repeated at each node until the packet reaches its destination.

VI.2.5 OPERA Performance Analysis

One of the key strengths of OPERA is to mix the idea of data mules and routing in *a smart way* that saves both bandwidth and time. In OPERA even if a local route seemed to exist, it may be better for the current car to carry the packet until it finds better route that guarantee fast and efficient delivery based on some performance metrics. Unlike DPP, any intermediate car in OPERA, not only the head and tail, can decide the current optimal route based on the most recent information.

We have integrated our mobility model with an 802.11b Java simulator [71], which integrated well with our mobility model, after modifying it to send beacons periodically every 300 ms. If a car detects a collision, received a corrupted beacon, it may choose another random time to start from. As in [63], the size of vehicles is ignored.

Messages Overhead

Figure VI.2.5 shows the bandwidth wasted by DPP, the number of extra messages sent by DPP over OPERA, when the oncoming traffic density is 100%, 80% and 60% of the co-directional direction density. As shown in the figure, DPP has a significantly larger overhead than OPERA. As we showed before in Section VI.2.1, DPP sends unnecessary messages when the co-directional cluster has no overlap with two adjacent oncoming clusters. Hence, co-directional clusters will not be able to work as bridges between two oncoming clusters as intended and DPP only

wastes the bandwidth. Also, DPP allows a message to be sent to the same vehicle more than once as we described before in Section VI.2.1.

OPERA avoids this overhead by sending a message only if it knows that it will achieve some progress along the propagation path; otherwise, it uses cars as data mules. Hence, OPERA does not send unnecessary messages. Indeed, Figure VI.2.5 shows that for a single packet, DPP would waste much more bandwidth than the clustering process.

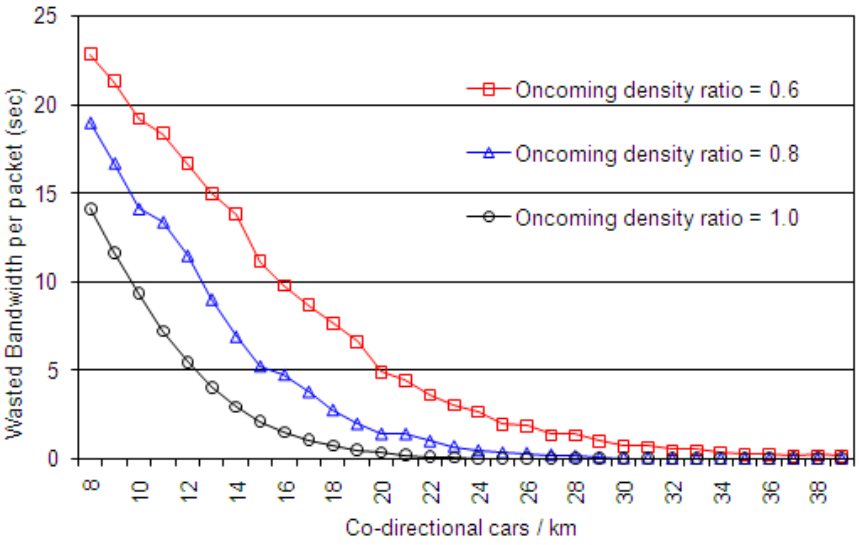

FIG. 31: DPP Overhead per packet

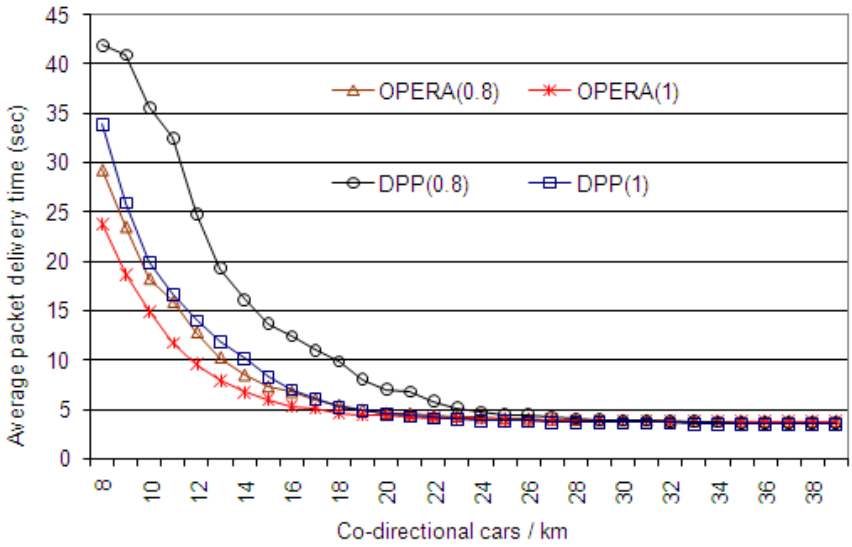

FIG. 32: Impact of cars density on packet delivery time

Packet delivery time

Figure 32 shows the impact of traffic density on the packet delivery time in both DPP and OPERA when the ratio between oncoming traffic density and co-directional density is 1 and 0.8. Figure 32 shows that OPERA outperforms DPP in terms of packet delivery time. The reason is that DPP favors the oncoming direction over the co-directional direction for packet propagation while OPERA does not favor any direction, seeking instead the optimal path to propagate the packet. Also, a header car in DPP would send the packet to oncoming cluster and wait for confirmation from co-directional cluster on the road. If for any reason, this confirmation is lost or did not even exist because there is no such cluster, the header car would wait for some time before trying again.

As it turns out, as the density of the oncoming traffic decreases, OPERA will be much faster than DPP. As an illustration, for 15 cars/km in the co-directional direction, packet delivery time in OPERA is about 50% of the packet delivery time needed by DPP. Under dense traffic, both protocols would have almost the same average packet delivery time as the network would be fully connected.

VI.3 SODA: A SMART OPPORTUNISTIC DATA DISSEMINATION APPROACH FOR DIVIDED ROADS

In Section VI.2, we presented OPERA and the clustering technique used by it. However, OPERA has the assumption that cars on opposite directions have the ability to communicate with each other. This assumption is not realistic in some highways where the two directions are separated by trees or some obstacles that prevent the communication between them. In this section, we present a modification of OPERA, which we may refer to as SODA [72], that works on divided roads as well by taking of mobility attributes into consideration.

It is noteworthy to mention that our proposed data dissemination for divided roads can be used with OPERA itself to improve data forwarding within a cluster before moving to a different cluster on the opposite direction.

Similar to Section VI.2.1, Figure 33 shows an example of how opportunistic data dissemination can be beneficial in divided roads.

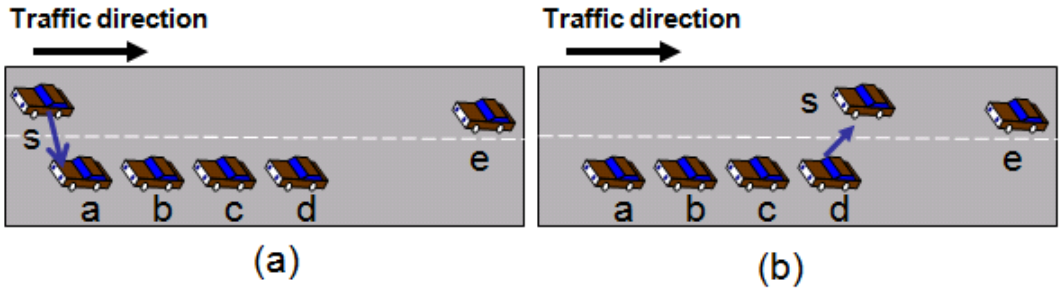

FIG. 33: A divided road motivating example

Refer to Figure 33(a) where car s wishes to deliver a packet to car e. Let us suppose that car s is faster than all cars a, b, c and d and is expected to meet the destination first. Most of the existing techniques in literature would work in this case as follows: Once car s detects some connectivity to car a on the road, it will send the packet to a. Then, the packet will be multi-hopped, in the obvious way, through cars b and c until it reaches car d. Since, car d has no connectivity to any other car in front of it, it will hold the packet until car s comes within range again, since car s is faster, as in Figure 33 (b).

Thus, the packet originating at car s ends up at car s again. Clearly, it would have been better for car s to keep the packet and not send it to car a at all. The above example shows how most of the existing protocols may waste bandwidth and/or suffer from routing loops.

To illustrate the basic idea of SODA, consider a multi-lane highway shown in Figure 34. Assume that car s carries a packet to be sent to a static destination d, d would be also a moving car but packet delivery would not be guaranteed in this case.

If s has no connectivity to any car in front of it on the road, it will simply carry the packet. However, if connectivity exists as in Figure 34, then s will have two options. The first option is to send the packet to the closest car to d on the road, this car can be selected from the list of single-hop neighbors maintained by s. The second option for s is to carry the packet and not to send it even if connectivity exists. In our motivation example shown in Figure 33, it was better for car s not to send the packet as we explained before.

Returning to our example in Figure 34, using the clustering information maintained by cars, car s can decide to send the packet or to keep it based on its information about its cluster as follows.

If car s finds itself expected to meet the destination before the header of its cluster, then it is better to keep the packet in order to save communication resources.

However, if s finds that the header of its cluster is expected to meet the destination before itself, then s will send the packet to the furthest car using its list of single-hop neighbors. Then SODA would be applied again by the new holding car.

It is noteworthy to mention that car s, the current holder of a packet, is dynamically taking its decision based on the current cluster information updated continuously. For example, if s's cluster has merged with another cluster ahead on the road, then car s will detect that by the clustering process and take the best routing decision based on its current information.

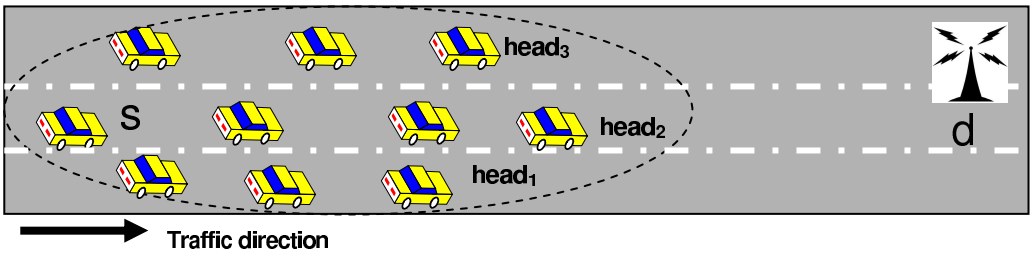

FIG. 34: SODA: illustration example

Actually, SODA would be very beneficial when each lane has its own average speed, this is usually the case and is even enforced by law in some countries. This is because it may be better for a car in the faster lane to carry the packet than to send it to cluster of cars in slower lanes then face lack of connectivity. Thus, the routing decision in SODA is being done based on both the speeds and connectivity between cars.

VI.3.1 SODA Performance Analysis

We implemented SODA on top of the traffic simulator described in Section VI.2.5. As we have shown in our motivation example, SODA can save much bandwidth by avoiding useless transactions and sending packets in a clever way. Unless otherwise specified, we assume a four-lane highway with the difference of average speeds between each two adjacent lanes equal to 10 meters/sec. We also assume that the source and destination are initially 4 km apart from each other. In [26], CAR was proven to outperform GPSR [73]. So,we compare SODA with CAR for our highway scenario.

Number of Messages and Bandwidth Usage

In addition to beacons that are transmitted by both SODA and CAR, CAR has a destination discovery process before sending the real data. Thus, we expect

vehicles in CAR to send more control messages in addition to unnecessary data packets as we explained in our motivation example. Figure 35 shows the impact of traffic density on the number of *data* messages sent to route *a single packet* for both two-lane and four-lane scenarios for both SODA and CAR.

Figure 35 shows that under sparse traffic, less than 12 cars/km, more data messages need to be sent for four lanes than for two lanes. This is because the topology changes very rapidly in four lanes as a consequence to large differences between speeds in different lanes. As expected, SODA sends fewer number of data messages than CAR for all cases especially under non dense traffic density. In dense traffic, in which connectivity exists most of the time, SODA would send the same number of data messages as CAR. However, the total number of messages sent in CAR, control and data, is twice those in Figure 35 because of the route discovery process. So, we can say that SODA always sends much fewer number of messages than CAR.

Figure 36 shows the amount of extra bandwidth used by CAR over SODA for both two and four lane scenarios assuming packet size of 512 kbytes. This figure shows that CAR would waste much bandwidth time in the non-dense traffic condition. However, as the traffic becomes dense, connectivity would exist between cars and little bandwidth is being wasted, only for the path discovery process. . Of course, if we consider large traffic between multiple sources and destination, we would expect huge amount of bandwidth to be wasted by CAR.

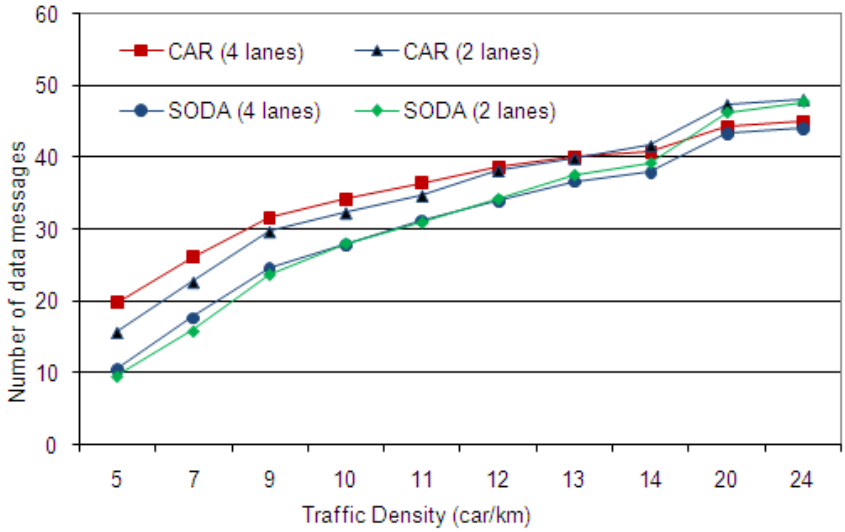

FIG. 35: SODA: Impact of traffic density on number of data messages

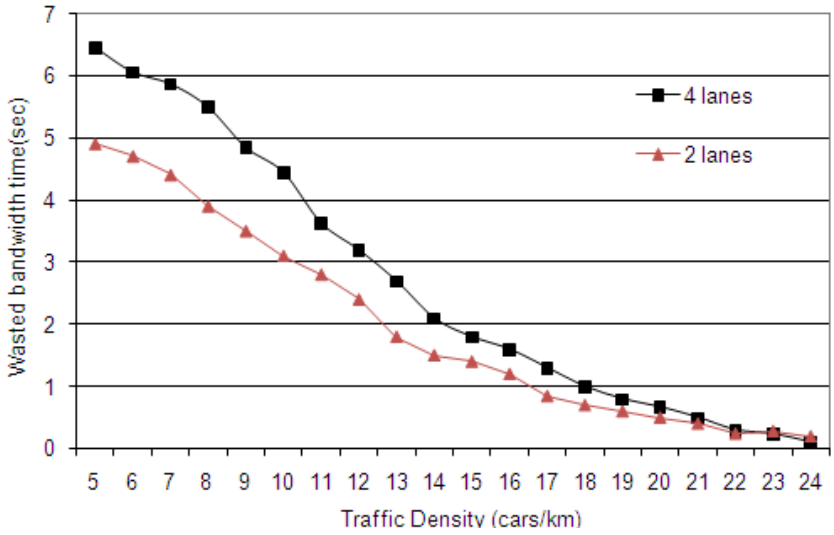

FIG. 36: SODA: Impact of traffic density on wasted bandwidth

Dissemination Time

Saving bandwidth and communication resources do not represent much importance if not joined with efficient dissemination time. In this experiment, we compare the dissemination time of both SODA and CAR when routing a single packet over 4 km.

Figure 37 shows the the impact of traffic density on dissemination time for both SODA and CAR. This figure shows that SODA outperforms CAR under all traffic conditions and number of lanes. This is because CAR waste much time in the discovery process, sending control messages to the destination and then back to the source. Figure 37 shows also that SODA has more improvement over CAR under non dense traffic than in dense traffic. As an illustration, for traffic density of 11 cars/km and 4 lanes, CAR spent about 132 sec in disseminating data while SODA required only about 77 sec which is about 40% improvement.

VI.4 HOW FAR SHOULD A MESSAGE BE PROPAGATED?

This section is devoted to answering one of the most interesting question that is how far should a notification be propagated to alert drivers approaching incidents? To the best of our knowledge, this is the first attempt to answer this important question in VANET research. If the notification is propagated for a short distance, drivers may not be alerted early enough to make appropriate decisions to avoid being blocked behind the accident. On the other hand, propagating the notification for a longer distance will waste a lot of expensive limited resources, like

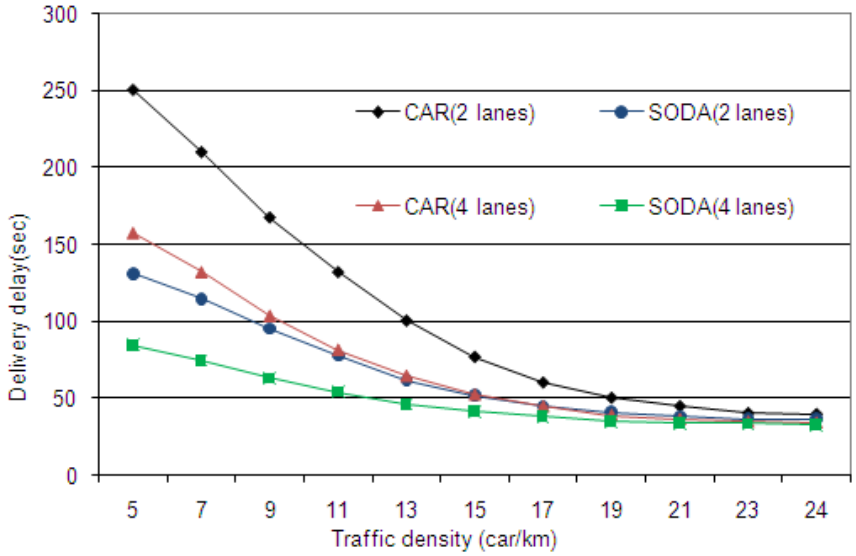

FIG. 37: SODA: Impact of traffic density on dissemination time

bandwidth, and moreover drivers far away from the accident may consider these alerts as false alarms and if happened more frequently they may even turn off their devices to avoid getting these many unnecessary alarms.

Refer to Figure 2 in Chapter III, assume that belt A' is aware of the accident and has informed belt A about that. Vehicles in the opposite direction of the accident may be used to carry the message in order to notify other vehicles approaching the accident as described before in Section III.5. The message will be propagated by vehicles like y to belts B, C, D and so on. So, the problem at hand is when should a belt decide to stop propagation of the message? A good answer for that question is that a belt should stop propagation of a message when it detects that cars on the opposite direction are running normally and have not been backed up yet behind the accident. Furthermore, those cars also should have at least one exit before being backed up behind the accident. Referring back to our example, if belt C' infers somehow that cars are running normally between D' and C', it may inform belt C about that. Thus, belt C may assume that it is not important to propagate the message to belt D and the message propagation may be interrupted.

Thus, the general approach can be described as follows.

- Each belt, A, computes the probability, $P_{AB_{normal}}$, that traffic between itself and the previous belt B, on same direction, is normal (cars are not stuck yet).

- Each belt A sends the probability computed in the previous step to its

adjacent belt A' on the opposite direction.

- Upon receiving the probability of normal traffic from opposite belt on the other direction, $P_{AB_{normal}}$, a belt A' propagates the alert messages, if any, to next belt on the road with probability $1 - P_{AB_{normal}}$.

The following subsections are used to describe each step of the general approach in more detail.

VI.4.1 Computing the Probability of Normal Traffic

The purpose of computing P_{normal} is to have an indication whether cars are getting stuck behind an accident somewhere in front or they still can freely move as *usual*, it is not intended to detect the accident itself.

Therefore, an easy efficient way to compute this probability is to use the running average speed computed by a belt as a good indication. So, each belt can compute the probability that traffic flow is normal or slower than usual. For example, if a belt computes the running average speed as S_{actual}, which will be provided to the belt as part of the EDR information. Let us also assume that the expected speed for that section of the road at that (time, day, date) is $S_{expected}$, then a belt may compute the probability of normal traffic, P_{normal}, as

$$P_{normal} = min(S_{actual}/S_{expected}, 1)$$

VI.4.2 Heart Beating

Every time interval, 30 seconds or so, each belt computes the probability of normal traffic, P_{normal}, and sends this probability to its adjacent belt on the opposite direction to help the latter make its decision of notification propagation.

VI.4.3 COX Distribution

The COX distribution is a generalization of the phase-concept by Erlang [74]. The phases are independent of each other and are exponentially distributed random variables with parameter μ_i, i = 1 , 2, ..., n. The arrangement of the phases is given in Figure 38. The COX distribution may be described as follow. Consider a service facility with n phases of service channels (nodes). The service time distribution at node (phase) i is exponentially distributed with parameter μ_i. A job (customer) enters from the left and moves to the right. After receiving service

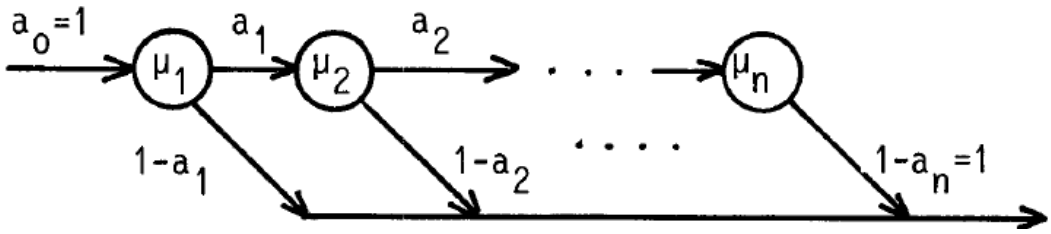

FIG. 38: Cox distribution

at node i, the customer may leave the system with probability $1 - a_i$ or move to next node with probability a_i.

We can define the probability of having exactly i phases as

$$P_i = (1 - a_i) \prod_{k=1}^{i-1} a_k \tag{20}$$

We can also define the probability of going through at least $i + 1$ phases as

$$q_i = \prod_{k=1}^{i} a_k \tag{21}$$

For better notation, let us name belts on the road as $b_1, b_2, ..$ and $b'_1, b'_2, ...$ Moreover, let us assume that a message is being propagated in the direction of b_1 to b_2 to b_3 and so on. The problem at hand, of identifying how far should a message be propagated, has the same behavior of the COX distribution where:

- The waiting time at each stage is equivalent to the time of propagating messages between belts. A belt waits for some time before it gets the message from previous one on the road to be able to make its decision to forward it to the next belt or not.

- After receiving the message, a belt i will propagate the message to the next belt with probability $1 - P_{i,i+1_{normal}}$ or it may stop the propagation with probability $P_{i,i+1_{normal}}$

Similar to Equations 22 and 24, we can write the probability that the message propagation will stop after exactly i phases as

$$P_{propagation\_exactly\_i} = P_{i,i+1_{normal}} \prod_{k=1}^{i-1} (1 - P_{k,k+1_{normal}}) \tag{22}$$

We can also define the probability of the message being propagated for at least $i + 1$ belts as

$$q_{at\_least\_i} = \prod_{k=1}^{i} (1 - P_{k,k+1_{normal}}) \tag{23}$$

And the expected number of belts to receive the message, how far should a message be propagated, would be

$$E = \sum_{i=1}^{\infty} (i * P_{propagation\_exactly\_i}) \tag{24}$$

VI.5 SUMMARY

Chapter V provided a proof that end-to-end connectivity is not guaranteed in VANET meaning that traffic is partitioned into clusters. Chapter V also provided a way to compute the expected cluster and showed that clusters are quite stable over time. All of these findings have motivated us to develop our data dissemination techniques in VANET that take disconnection into consideration.

In this chapter, we started by proposing a vehicles clustering technique so that each car within a given cluster maintains information about its neighbors as well as information about the header vehicle of its cluster. Simulation experiments showed that vehicles will have up-to-the-second information about its cluster while using little bandwidth in the maintenance process.

We also proposed our data dissemination approach for undivided roads namely OPERA. In OPERA, a packet may hop between clusters or cars moving in opposite lanes until, eventually, it reaches its destination. In this sense, OPERA is actually a *hybrid* protocol as it alternates between applying proactive routing and data mules in a clever way to avoid delay or bandwidth wasting. The main theme of OPERA is to send a packet only if it will make progress in the dissemination process. Simulation results showed that OPERA outperforms DPP in terms of delivery time and bandwidth used.

Following OPERA's philosophy, we proposed another data dissemination technique, SODA, for divided roads taking vehicles mobility and lack of connectivity into consideration. In SODA, faster vehicles take higher priority of keeping packets even if connectivity exists as long as these faster vehicles are expected to meet the destination before other cars in their clusters. Simulation results also showed that SODA outperforms CAR in terms of delivery time and number of messages.

Finally, we provided an answer to one of the very challenging questions that is how far should a notification be propagated to alert approaching drivers. We developed a probabilistic technique for belts to keep propagating packets to previous belts as long as they are detecting lower average vehicles speed than historical known average speed for that section of the road. By mapping the problem at hand to a standard COX distribution, we provided a formula for the expected number of bets to receive the notification when an incident is detected.

CHAPTER VII

SECURING NOTIFICATION DISSEMINATION

In Chapter VI, we have presented and discussed techniques for notification dissemination between belts for both divided and undivided roads as well as how far should a message be propagated. However, messages exchanged between belts have to be secured against different kinds of attacks [75]. This chapter is devoted to presenting security techniques used for secure data dissemination. We are concerned here about two main aspects of security: Firstly, making sure that vehicles receive notifications from real belts rather than someone throwing fake devices on the road. Secondly, securing the message propagation between belts to avoid some possible threat attacks that will be listed shortly.

General belt to vehicles communication were summarized in Section III.3. However, the intention of this chapter is how to secure data dissemination between cars as well as allow vehicles to accept and process only data sent by real belts.

VII.1 THREAT MODEL

First, we illustrate different threats that we are concerned with in notification dissemination

- Tracking a vehicle: A very dangerous and ignored fact about privacy is that innocent looking data from various sources can be accumulated over a long period and evaluated automatically revealing much information about these sources. Even small correlations of the data may reveal useful information. For instance, the knowledge about specific sensor characteristics may give some hints about the make and the model of the passing car. This in turn may be related to other information to identify a specific car.

  As privacy sometimes contradicts with security requirements, any security technique should not allow malicious attackers to violate drivers privacy. While system operators want to find or identify attackers to take proper countermeasures, the ability to do so may be used for less than noble reasons.

- Replaying old messages: An attacker may sit by the road with a small device that records communication between belts and vehicles. Then it replays these messages to deceive other vehicles. Thus, he may send old information to passing vehicles about an incident that does not exist anymore.

- Eavesdropping: In this type of attack, a malicious driver or person may try to record all messages sent by belts or vehicles in order to gain some information about vehicles and their drivers from a simple analysis of these messages.

- Changing the message: An attacker may try to change a message in order to change its meaning or even to corrupt it.

- Black hole: One of the dangerous attacks in data dissemination in NOTICE is to have a vehicle that is not cooperating and is trying to stop the propagation of messages and notifications. This may be a person sitting by the road receiving messages from vehicles and dropping them or he may be a driver that does not cooperate in notifications forwarding.

- Giving vehicle incorrect information: An attacker may put a small device over the road impersonating a belt and telling cars incorrect information about fake incidents on the road.

VII.2 NOTATIONS

We use the following notations throughout the rest of this section:

- $\{M\}_k$ is the result of encrypting message M with a key k.

- $[C]_k$ is the result of decrypting the cipher C using a key k. For example, for a symmetric key k, if C = $\{M\}_k$ then M = $[C]_k$

- $Public_x$ is the public key of node, vehicle or a belt, x.

- $Private_x$ is the private key of node x.

- key_{AB} is a shared symmetric key that is known by both A and B

- $M_1|M_2$ is the resulting message of concatenating two messages M_1 and M_2

- $A \rightarrow B : M$ means that node A, *vehicle or a belt*, sends a message M to node B. If B is *, this means that A is broadcasting M.

- The term *message* refers to an encrypted message that a belt wants to communicate with next belt by the help of passing vehicles. The term $notification$ refers to a traffic-related information that a belt informs vehicles about. For example, when a belt detects an incident on the road, it sends a *message* to next belt on the opposite direction to notify approaching

cars. Then the latter in turn will send *notifications* for cars to avoid the incident.

VII.3 BELTS FUNCTIONS AND ASSUMPTIONS

This section is devoted to describe belts assumptions and functions that are needed to enhance security of NOTICE and to provide defense against different attacks listed in Section VII.1.

- As has been described in Chapter III, any message exchanged between two consecutive belts is encrypted with a shared secret symmetric key known between them. This is important to prevent any attacker from changing or corrupting any message exchanged between two belts. Also, an attacker will not be able to understand the contents of the encrypted message.

- A belt gives a message to more than one vehicle to increase the probability of being received by next belt. This is because drivers may take an exit, stop by the traffic shoulder or even be malicious. So, relying on one vehicle is not practical and some forms of redundancy need to be maintained. However, the more vehicles to get the message from a belt, the more saturated the communication medium will be after that in order to disseminate all of these messages to next belt. This will in turn prevent man in the middle attacks, in other words minimizing the probability of having such attack in place.

- In addition to the shared symmetric key between any two consecutive belts, all belts within a given region share a private key that may be changed over time by some distribution authority, or belts can simply have a timed manner in switching between key chain, where the corresponding public key is well known between cars. i.e. may be available over the Internet or broadcasted frequently over cellular network or satellite communication using PKI certificates. This is also important to provide a way for passing drivers to authenticate belts as we explain in the next item.

- If a belt has a message to send to the next belt, it will timestamp and sign that message using its private key. It is very important here to mention that the signature is being done offline, not in the limited communication time between a belt and a car. So, if a belt has a message to send, it will encrypt and sign that message while waiting for first passing car to start the dissemination to next belt. By signing a message and adding timestamp to it, passing vehicles can authenticate belts and accept notifications only

from real belts. On the other hand, this will also prevent an attacker from sending fake messages to passing drivers.

VII.4 VEHICLES FUNCTIONS AND ASSUMPTION

This section provides a description for vehicles functions and assumptions in enhancing the security of NOTICE.

- Vehicles are constantly receiving the belt's public key for their area through satellite communication or cellular networks. These can be done by the use of standard PKI certificates that are widely used in distributing public keys in modern systems.

- Vehicles accept notifications about traffic conditions from belts or cars as long as they are signed using the belt's private key and it has not expired yet. So, no attacker can put a device that impersonates a belt and sends false notifications about non existing incidents. Moreover, that same attacker cannot also send fake message to next belt on the road as no car will cooperate in delivering his messages to next belt.

- Each vehicle that receives a message directly from a belt will never drop that message until it gives it to the next belt by itself. So, data mules are still working behind the scene, so that if some form of black hole exists, there is a chance for the message to be delivered. We may note here that a vehicle that receives the encrypted message from another vehicle, not directly from a belt, in the dissemination process will discard this message once it forwards it.

- While approaching a belt, not within the small communication time, a vehicle encrypts its EDR data using belt's public key before sending this data to next belt. Thus, *only* real belts can decrypt such information to do analysis and no attacker can understand EDR information which in turn preserves drivers privacy. Moreover, this prevents attackers from performing analysis similar to those performed by belts and getting some knowledge about this highly classified information that should reside only with authorized entities.

- All vehicles have the same communication range. So, if vehicle x can send to vehicle y, then x can also receive from y.

VII.5 SECURE DISSEMINATION

We may now start describing the rest of our secure data dissemination technique. When a vehicle x passes over a belt A that has a message to be sent to next belt B and/or a notification for x, the following will happen.

1. $A \rightarrow x : Id_x|msg_{id}|\{msg\}_{Key_{AB}}|notifications|expires\_at|A_{sign}$

   Basically, a belt A sends a message that consists of :

   - Id_x: a unique identifier for each message sent by A. This may simply be an random unique identifier string generated by the belt
   - msg_{id} : An unique identifier generated randomly at the belt to uniquely identifies the message itself. So, msg_{id} is unique for same message $\{msg\}_{Key_{AB}}$ even if it was sent to multiple cars while Id_x should be changing
   - $\{msg\}_{Key_{AB}}$: A message that needs to be communicated securely with belt B
   - notifications: Any notification that belt A would like x to be aware of, such as any incident/accident ahead on the road
   - expires_at: A time after which this message should be discarded. This will in turn limit the impact of replaying message attacks because a message will be invalid after a limited time.
   - A_{sign}: A's signature for that message.

It is worthy here to mention that if A does not have a message for B nor a notification for x, it will send the remaining parts to x to be used as a proof it has passed recently over the belt.

It is very important also to emphasize that a belt will start communication only with real moving vehicles using doppler effect to avoid having a person sitting beside the belt and trying to perform a DOS attack on the belt.

By verifying A_{sign} for the message, x can authenticate A. If that authentication was successful, the notification would be displayed to x's driver console and x will try to propagate the message to next cars on the road as we will describe next.

Let us assume for now that vehicle x has detected that there are some other vehicles ahead of it on the road and vehicle y has been selected by the dissemination technique to be next hop.

Therefore, to avoid routing holes if y is a malicious attacker, x will try to authenticate y first before dropping the the message as follows.

- $x \rightarrow y : Id_x|msg_{id}|\{msg\}_{Key_{AB}}|notifications|expire\_at|A_{sign}$

By sending the message it received from the belt, x is authenticating to y that it is a real vehicle that has passed recently over the belt. If y could successfully verify A's signature and make sure that the message has not yet expired, it will send the message it received from A, when it first passed over A. So, the following will be performed at y:

- If this is a new message, using the message id, then it shows the relevant notifications to y's console
- If y's position is beyond B_{Pos}, y has already passed the belt, then it ignores that request. Otherwise continue to next step.
- $y \rightarrow x : Id_y|\{msg\}_{Key_{AB}}|notifications|expires\_at|A_{sign}$

By receiving the message from y that contains Id_y and A's signature, x can verify that y is a real car that has passed recently over A.

The same thing will be repeated again with y when it tries to propagate the message to another car z selected by the dissemination process with the exception that y will discard the message once it can authenticate z. Remember also that x, the car the received the message directly from A will never drop the message until it delivers it itself to B.

According to the previous protocol, it is very clear that the following constraints are maintained and verified between x, y and A. Firstly, x and y can verify that A is a real belt using A's signature and the expires_at values. Secondly, y could verify that x is a car that has passed recently over A and the message to be propagated was really sent by A. Finally, x, or any car in the dissemination process, could verify that y is a real car that has passed recently over A.

One of the possible attacks to the proposed technique is to have a malicious driver, D_1 who receives a real message from a belt and then stops by the road replaying same message to all passing cars. It is very clear that this is a limited attack as the message will expire and no car will accept it after the expires_at time. Also, having the msg_{id} will allow B to easily accept the first message received and mark the others as duplicates

VII.6 NUMBER OF COPIES TO SEND

One of the very important parameters in message propagation is how many cars would a belt give its message to? The main reason for these copies is to avoid

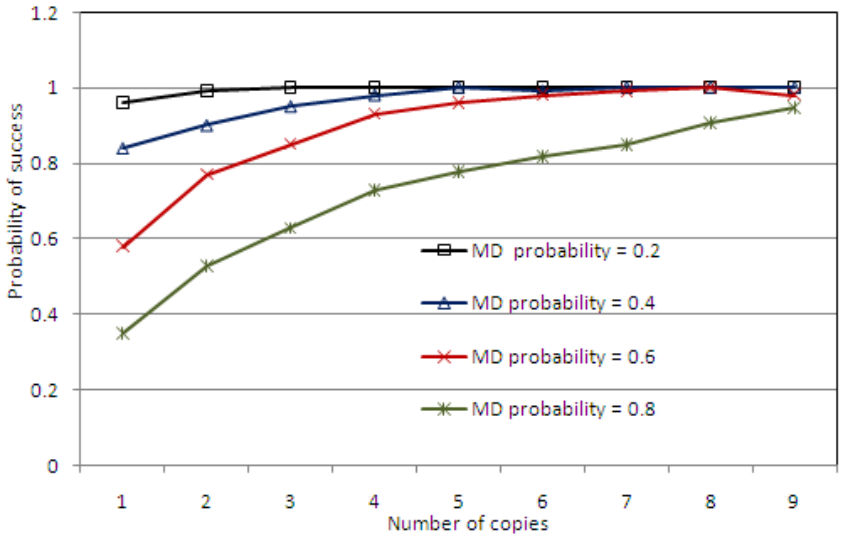

FIG. 39: Impact of misbehaved drivers on number of copies

giving a message to a car that may take an exit before the next belt, stop by the traffic shoulder for some reason or even be a malicious person who tries to stop message propagation between belts. We simply refer to all of these, from the belt point of view, as Misbehaving Driver (MD).

Figure 39 shows the effect of the number of copies on the probability that the message would reach next belt under MD probabilities of 0.2, 0.4, 0.6 and 0.8.

Figure 39 also shows that up to MD probability of 60%, 5 copies would give a very high probability that the message would reach next belt. Actually, this result is important because it emphasis hat a belt does not need to flood the entire vehicle fleet with messages for next belt.

This figure also shows that 9 copies of a message should be sufficient even under misbehaving drivers probability of 0.8.

VII.7 BELTS CLUSTERING

One of the very important aspects that has not been addressed yet is a belt's failure. What happens if a belt B fails? Unfortunately, if a belt B fails, then no other belt would be able to decrypt msg_{AB} sent from A to B and hence, all incidents detected by A, or by any other belt before A, will never be propagated beyond B. i.e. B becomes a black hole in message propagation.

Another problem is the operational load of managing private keys installed in belts. If a belt was compromised, then the private key has to be changed to avoid having a malicious user sending incorrect data to drivers. However, the process

of changing that key will be very difficult if all belts share the same private key. Note that it is highly recommended to change private keys *manually* to prevent the compromised belt from getting the new key.

The answer to the above two concerns is clustering. By having belts on the road forming clusters, where a cluster consists of a group of belts, managing keys can be easier and moreover the probability of having a black hole can be reduced. Within a single cluster, belts share a secret symmetric key to communicate messages effectively as well as sharing same private key that would be used by passing cars to authenticate belts in that cluster. Moreover, each belt in a cluster maintains a shared symmetric key with the previous and next clusters on the road. For example, in Figure 40, there exist three clusters A, B and C where all belts within each cluster share same private key, symmetric key as well as shared symmetric key with previous and next clusters. For illustration, all belts in cluster B share same private key, maintain a shared symmetric key between themselves key_B and with clusters A and C. i.e. key_{AB} and key_{BC}.

There are many advantages of belts clustering that can be summarized as follows:

- If a belt b_2 in cluster B fails and becomes completely out of service, then any message sent by b_1 can still be decrypted at b_3, since all belts within B share same symmetric key for encrypting messages. Also, if belt b_1, the first one in the cluster failed, then b_2 would still be able to decrypt any message sent by a_3, the previous cluster, since all belts in B share key_{AB}.

  The problem however is when belt b_3 fails, which is the last belt in the cluster, that is supposed to encrypt the message using the shared symmetric key with the next cluster, key_{BC}. This problem can simply be addressed by redundancy. So, last belt in each cluster may be replicated to reduce the probability of having a black hole in message propagation as we will describe shortly. Even if no redundancy employed, the probability of having black hole in NOTICE is still much lower than the no clustering case.

- If a belt b_2 is compromised, then the operator will have only to change $B_{private}$, key_{AB} and key_{BC} which were known to b_2. Hence, managing the compromised belt does not require enormous effort in NOTICE if the cluster size is relatively small.

- Message propagation within the same cluster can be much faster now as each belt will not need to decrypt/encrypt the message propagated from previous belt to next one, since they all share same symmetric key. For

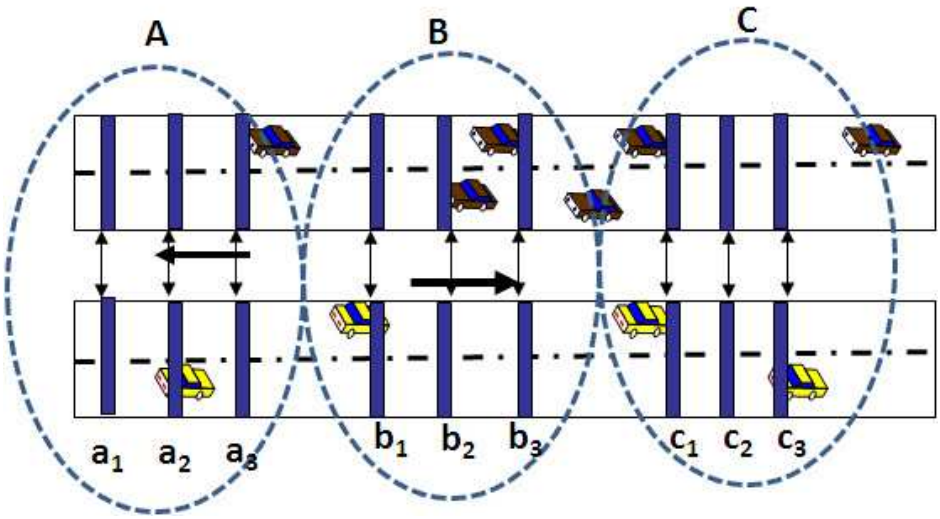

FIG. 40: Belts forming clusters

example, referring again to Figure 40, if belt b_1 detected an incident and started message propagating to belt b_2, b_2 can just pass the message to belt b_3 without any decryption/encryption since b_3 can decrypt any message sent by b_1 using the cluster symmetric key.

A natural yet important question is how large should a belt cluster be? On one extreme, if all belts form a single cluster then having any belt compromised would require a tremendous work from the operator to manage installing new keys in *all* belts and even worse, a malicious user would control all aspects of NOTICE if no one detected the compromised belt and can send fake messages to adjacent belts as well as fake notifications to passing vehicles. On the other extreme, if cluster size equals 1, no clustering, then a single belt failure means a black hole in message propagation between belts before and after that failed belt which will result of not alerting approaching drivers and got them all stuck behind the incident.

Let n be total number of belts of a given section of the highway and m be number of belts within a cluster, then

$$Number\ of\ clusters = \frac{n}{m} \tag{25}$$

Moreover, let us assume that P_f is the probability that a single belt fails and cannot do its functions any more. In case of cluster size equals 1, no clustering, we can write the probability of having black hole, P_{black_hole} as

$$P_{black\_hole} = 1 - (1 - P_f)^n \tag{26}$$

For example, if n = 1000 belts and $P_f = 0.01$ then $P_{black\_hole} = 0.9999$ which is

not an acceptable probability and simply means that black holes will be the norm in NOTICE rather than the exception.

Let us now consider the general case when belts forms clusters of size m, then the probability of having a black hole is the probability of having a failure in the last belt of a cluster, since the last belt is responsible for encrypting the message using next cluster's key. We can write P_{black_hole} as

$$P_{black\_hole} = 1 - (1 - P_f)^{\frac{n}{m}} \tag{27}$$

For example, if n = 1000, m = 50 and $P_f = 0.01$, we would have P_{black_hole} = 0.182 which is a great improvement over no clustering for such relatively small cluster size. Even better, if we install number of replicas of the last belt in each cluster equals to r , then we can write P_{black_hole} as

$$P_{black\_hole} = 1 - (1 - P_f^r)^{\frac{n}{m}} \tag{28}$$

For example, if r = 2, n = 1000, m = 50 and $P_f = 0.01$, then P_{black_hole} = 0.004 which is a very appealing probability of failure for such a small overhead of installing one extra belt for every cluster of size 50 belts. Table 1 summarizes the probability of having a black hole for some values of n, m, r and P_f. As an illustration, having a small cluster size of only 5 belts with no redundancy for the last belt will result in over 400% improvement over no clustering for n = 50. Moreover, the overhead of managing cluster of size 5 is very little and will be very easy to the operator to control.

VII.8 SUMMARY

Security is an essential component for any safety application for notifying drivers about traffic incidents. Otherwise, an attacker may send notifications about fake incidents, which discredit the otherwise useful system. In this chapter, we presented a technique to secure message dissemination and incident notification in NOTICE. We started by listing possible security threats that may affect NOTICE. Then we proposed enhancements to both belts and vehicles functions to make them resilient to the listed attacks.

Finally, we presented a belt clustering scheme to reduce the probability of having a black hole in the message dissemination if a belt fails and also to reduce the operational burden if a belt is compromised. We showed that even a small cluster size of 5 or 10 belts per clusters can significantly reduce the probability of having black holes in the data dissemination and is easy to maintain by traffic operators.

TABLE 1: Data dissemination black hole probabilities

n	**m**	**r**	P_f	P_{black_hole}
1000	1	1	0.001	0.63230
1000	20	1	0.001	0.04879
1000	20	2	0.001	0.00005
1000	50	1	0.001	0.01981
1000	50	2	0.001	0.00002
1000	1	1	0.01	0.99996
1000	20	1	0.01	0.39499
1000	20	2	0.01	0.00499
1000	50	1	0.01	0.18209
1000	50	2	0.01	0.00199
100	1	1	0.01	0.63397
100	5	1	0.01	0.18209
100	10	1	0.01	0.09568
50	1	1	0.01	0.39499
50	5	1	0.01	0.09568

CHAPTER VIII

CONCLUSION

In this thesis, we proposed a novel framework for incident detection and notification dissemination in VANETs. Our framework consists mainly of three components: system architecture, traffic incident detection and notification dissemination. We also proposed a security technique to prevent possible attacks in message propagation and incident notification.

The first component of our framework is an architecture for the notification of traffic incidents, NOTICE for short. In NOTICE, sensor belts are embedded in the road at regular intervals every mile or so. Each belt consists of a collection of pressure sensors, a simple aggregation and fusion engine, and a few small transceivers. The pressure sensors in each belt allow every message to be associated with a physical vehicle passing over that belt. Thus, no one vehicle can pretend to be multiple vehicles and there is no need for an ID to be assigned to vehicles. Vehicles in NOTICE are fitted with a tamper-resistant *Event Data Recorder* (EDR), like the well-known black-boxes on-board commercial aircraft. EDRs are responsible for storing vehicles behavior between belts such as acceleration, decelerations and change lanes. Importantly, the driver can provide input to the EDR, using a simple menu, either through a dashboard console or through verbal input.

NOTICE's belts collect data from passing cars about their experience on the road such as lane changes, stoppages, accelerations and deceleration. This information in turn is fed to our incident detection engine. We proposed techniques to enhance automatic incident detection in VANET. We started by proposing a deterministic technique where a belt collects EDR data from passing vehicles and maintains a table that stores occupancies for all positions since previous belt. These occupancies are used to detect possible blocking incidents that forced drivers to change lanes. A belt can identify those positions by looking for low occupancies in one lane where the corresponding positions in adjacent lanes have higher occupancies. Then we presented a more generic Bayesian based probabilistic technique that incorporates more parameters in the detection process than just lane changes. Our probabilistic technique is capable of detecting both blocking incidents such as vehicle accidents and non blocking incidents such as potholes. Our probabilistic technique added good enhancement to existing AID techniques in non dense traffic. However, as the traffic become denser, it is hard for vehicles to avoid the

accident and continue to provide their EDR data to the next belt. For evaluation purposes, we integrated our probabilistic technique with the California Algorithm and showed that this will help detecting blocking incidents in all traffic conditions. Our probabilistic technique also offers *zero* false positive alarms for most detection thresholds values, which makes it perfect and trustable to traffic operators. Simulation results showed that our probabilistic technique outperforms California Algorithm under non dense traffic in terms of mean detection time, detection rate and false positive rate. To the best of our knowledge, our probabilistic incident detection technique is the first VANET based approach capable of detecting both blocking and non blocking incidents.

Once an incident is detected, notification about this incident should be sent to all drivers approaching it. For this purpose we proposed data dissemination techniques designed specifically for VANET. We started by performing analysis, confirming empirical results found by other researchers, proving that VANETs tend to be disconnected in many highway scenarios, consisting of a collection of disjoint clusters. We have also measured the expected cluster size and we found that co-directional clusters tend to be relatively stable. Based on these traffic characteristics, we developed two data dissemination approaches that effectively disseminate messages in both divided and undivided roads to disseminate data in an opportunistic way. For undivided roads, we proposed an opportunistic data dissemination approach called OPERA. In OPERA, a packet may hop between clusters or cars moving in opposite lanes until, eventually, it reaches its destination. In this sense, OPERA is actually a *hybrid* protocol as it alternates between applying proactive routing and data mules in a clever way to avoid delay or bandwidth wasting. The main theme of OPERA is to send a packet only if it will make progress in the dissemination process. Simulation results showed that OPERA outperforms DPP in terms of delivery time and bandwidth used.

Following OPERA's philosophy, we proposed another data dissemination technique, SODA, for divided roads taking vehicles mobility and lack of connectivity into consideration. In SODA, faster vehicles take higher priority of keeping packets even if connectivity exists as long as these faster vehicles are expected to meet the destination before other cars in their clusters. Simulation results also showed that SODA outperforms CAR in terms of delivery time and number of messages.

We also provided an answer to one of the very challenging questions that is how far should a notification be propagated to alert approaching drivers. We developed a probabilistic technique for belts to keep propagating packets to previous belts as long as they are detecting lower average vehicles speed than historical known

average speed for that section of the road. By mapping the problem at hand to a standard COX distribution, we provided formulas for the expected number of bets to receive the notification when an incident is detected.

Finally, we presented a technique to secure message dissemination and incident notification in NOTICE. We started by listing possible security threats that may affect NOTICE. Then we proposed enhancements to both belts and vehicles functions to make them resilient to all possible attacks while preserving privacy of the drivers. We also presented a belt clustering scheme to reduce the probability of having a black hole in the message dissemination while reducing also the operational burden if a belt is compromised.

VIII.1 FUTURE RESEARCH DIRECTIONS

In this section, we discuss some ideas to extend our framework with more functionalities. We identified three directions to extend the proposed framework, which can be listed as follows:

1. Detect incidents in dense traffic

   In this thesis, we proposed AID techniques where NOTICE's belts collect EDR data directly from passing vehicles and perform some analysis to detect possible incidents. This will work well in none dense traffic as we pointed before in Chapter IV. However, in dense traffic and having a blocking incident that blocks one or more lanes, it would be difficult for vehicles to change lanes and reaches out to the next belt to provide their EDR information. Therefore, it is beneficial if a technique is developed so that cars, which are stuck behind incidents, can disseminate their EDR data to the next belt by the help of other cars.

2. Integrate NOTICE with the Internet

   One of the possible future research directions is to integrate NOTICE with the Internet. With this feature, the outcome of the incident detection engine can be fed to the Internet so that drivers can access such information from any place, and it is not necessary to be in the proximity of the incident. With this feature also, drivers can make appropriate decision before even reaching the incident location. On the other hand, traffic management can take a proactive role in managing alternative routes that drivers are expected to take to avoid the accident.

3. Advertisements in NOTICE

With the governments strugglingly to install traffic lights and patch the roads, it is hard to deploy new systems like NOTICE especially those that may be offer free services to the public. Therefore, it would be great if NOTICE is self-funded. One easy way to provide funding to NOTICE is to have belts advertise advertisements to passing cars about gas prices, hotel rates, restaurants and other possible services in the local proximity. It is noteworthy to mention that advertisements should be done in the belt's free cycle when it has nothing about traffic incident to tell passing cars or in general advertisements should take lower priorities than traffic related incidents.

BIBLIOGRAPHY

[1] J. Paniati, "Traffic congestion and sprawl," Federal Highway Administration, http://www.fhwa.dot.gov/congestion/congpress.htm, Nov. 2002.

[2] National Highway Traffic Safety Administration, "Traffic safety facts," http://www-nrd.nhtsa.dot.gov, 2005.

[3] US Department of Transportation, "Standard specification for telecommunications and information exchange between roadside and vehicle systems," *ASTM E2213-03*, August 2003.

[4] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, ser. SASN '05, 2005, pp. 11–21.

[5] A. Aijaz, B. Bochow, F. Dötzer, A. Festag, M. Gerlach, R. Kroh, and T. Leinmülle, "Attacks on inter-vehicle communication systems - an analysis," in *Proceedings of the International Workshop on Intelligent Transportation (WIT)*, Mar. 2006.

[6] J. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security Privacy*, vol. 2, no. 3, pp. 49–55, 2004.

[7] M. Raya, P. Papadimitratos, and J.-P. Hubaux, "Securing vehicular communications," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 8–15, 2006.

[8] E. Parkany and C. Xie, "A complete review of incident detection algorithms and their deployment: What works and what doesnt," in *The New England Transportation Consortium*, February 2005.

[9] M. Dan, D. Jasek, and R. Parker, "Evaluation of some existing technologies for vehicle detection," *TX: Texas Transportation Institute, Report FHWA/TX-00/1715-S. College Station*, 1999.

[10] K. N. Balke, "An evaluation of existing incident detection algorithms," *Texas Transportation Institute Research. Report 1232-20. Texas A&M University System College Station, Texas*, November 1993.

[11] B. N. Persaud and F. L. Hall, "Catastrophe theory and patterns in 30-second freeway traffic data – implications for incident detection," *Transportation Research, Part A, General*, vol. 23, no. 2, pp. 103–113,

1989. [Online]. Available: http://www.sciencedirect.com/science/article/B6X3B-466N1S1-37/2/2d7714895a574287a38c4b9b583fa368

[12] S. Alexander, C. , Ted, and R. Daniel, "The i-880 field experiment: Effectiveness of incident detection using cellular phones," *California PATH Research Report UCB-ITS-PRR-98-1*, May 1998.

[13] S. Kamijo, M. Harada, and M. Sakauchi, "Incident detection based on semantic hierarchy composed of the spatio-temporal mrf model and statistical reasoning," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, 2004, pp. 415 – 421 vol.1.

[14] D. Srinivasan, S. Sanyal, and V. Sharma, "Freeway incident detection using hybrid fuzzy neural network," *Intelligent Transport Systems, IET*, vol. 1, no. 4, pp. 249 –259, 2007.

[15] N.-K. Hong, J.-W. Choi, and Y.-K. Yang, "A study on incident detection model applying apid model, fuzzy logic and traffic pattern," *IEEE Intelligent Transportation Systems Conference. ITSC 2007*, pp. 196–203, September. 30 2007-October. 3 2007.

[16] K. feng Wang; Xingwu Jia; Shuming Tang, "A survey of vision-based automatic incident detection technology," *IEEE International Conference on Vehicular Electronics and Safety, 2005.*, pp. 290–295, 14-16 October 2005.

[17] P. Martin, J. Perrin, and B. Hansen, "An assessment and analysis of using dedicated short-range communications(dsrc) technology for incident detection on rural freeways," *PhD dessertationm, University of Kentucky*, 2004.

[18] E. Jakob, G. Lewis, H. Bret, N. Ryan, M. Samuel, and B. Hari, "The pothole patrol: using a mobile sensor network for road surface monitoring," in *MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services*, 2008, pp. 29–39.

[19] J. Karuppuswamy, V. Selvaraj, M. Ganesh, and L. E, "Detection and avoidance of simulated potholes in autonomous vehicle navigation in an unstructured environment. intelligent robots and computer vision," *Algorithms, Techniques, and Active Vision*, vol. 4197, no. 1, pp. 70–80, 2000.

[20] S. Tomas and H. Milos, "Towards a learning traffic incident detection system," *Proceedings of the Workshop on Machine Learning Algorithms for Surveillance and Event Detection*, May 2006.

[21] B. Williams and A. Guin, "Traffic management center use of incident detection algorithms: Findings of a nationwide survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 351 –358, June 2007.

[22] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Second IEEE Workshop on Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99.*, Feb. 1999, pp. 90 –100.

[23] Y. Zhong and D. Yuan, "Dynamic source routing protocol for wireless ad hoc networks in special scenario using location information," in *International Conference on Communication Technology Proceedings, 2003. ICCT 2003.*, vol. 2, 2003, pp. 1287 – 1290 vol.2.

[24] S. Manvi, M. Kakkasageri, and C. Mahapurush, "Performance analysis of aodv, dsr, and swarm intelligence routing protocols in vehicular ad hoc network environment," in *International Conference on Future Computer and Communication, 2009. ICFCC 2009.*, 2009, pp. 21–25.

[25] C. Lochert, M. Mauve, H. Fussler, and H. Hartenstein, "Geographic routing in city scenarios," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, pp. 69–72, January 2005. [Online]. Available: http://doi.acm.org/10.1145/1055959.1055970

[26] V. Naumov and T. R. Gross, "Connectivity-aware routing (car) in vehicular ad-hoc networks," *INFOCOM 2007. 26th IEEE International Conference on Computer Communications.*, pp. 1919–1927, May 2007.

[27] A. Bachir and A. Benslimane, "A multicast protocol in ad hoc networks inter-vehicle geocast," in *The 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring.*, vol. 4, 2003, pp. 2456 – 2460 vol.4.

[28] G. Korkmaz, E. Ekici, and F. Ozguner, "An efficient fully ad-hoc multi-hop broadcast protocol for inter-vehicular communication systems," *IEEE International Conference on Communications, 2006. ICC '06.*, vol. 1, pp. 423–428, June 2006.

[29] K. Ibrahim, M. Weigle, and M. Abuelela, "p-ivg: Probabilistic inter-vehicle geocast for dense vehicular networks," in *IEEE 69th Vehicular Technology Conference, 2009. VTC Spring 2009.*, 2009, pp. 1 –5.

[30] M. Mabiala, A. Busson, and V. Veque, "Inside vanet: Hybrid network dimensioning and routing protocol comparison," *IEEE 65th Vehicular Technology Conference, 2007. VTC2007-Spring.*, pp. 227–232, April 2007.

[31] W. Sun, H. Yamaguchi, K. Yukimasa, and S. Kusumoto, "Gvgrid: A qos routing protocol for vehicular ad hoc networks," *Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on*, pp. 130–139, June 2006.

[32] Z. Mo, H. Zhu, K. Makki, and N. Pissinou, "Muru: A multi-hop routing protocol for urban vehicular ad hoc networks," *Mobile and Ubiquitous Systems - Workshops, 2006. 3rd Annual International Conference on*, pp. 1–8, July 2006.

[33] A. Festag, R. Baldessari, and H. Wang, "On power-aware greedy forwarding in highway scenarios," Proc. 5-th International Workshop on Intelligent Transportation *(WIT'07)*, March 2007.

[34] T. Little and A. Agarwal, "An information propagation scheme for vanets," in *Proceedings of IEEE Intelligent Transportation Systems.*, 2005, pp. 155–160.

[35] M. Abuelela, S. Olariu, and I. Stojmenovic, "Opera: Opportunistic packet relaying in disconnected vehicular ad hoc networks," in *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems. MASS 2008.*, Oct. 2008, pp. 285–294.

[36] A. Agarwal, D. Starobinski, and T. Little, "Exploiting downstream mobility to achieve fast upstream message propagation in vehicular ad hoc networks," in *2007 Mobile Networking for Vehicular Environments*, May 2007, pp. 13–18.

[37] K. Plöíl and H. Federrath, "A privacy aware and efficient security infrastructure for vehicular ad hoc networks," *Comput. Stand. Interfaces*, vol. 30, pp. 390–397, August 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1379460.1379605

[38] B. Mishra, P. Nayak, S. Behera, and D. Jena, "Security in vehicular adhoc networks: a survey," in *Proceedings of the 2011 International Conference on Communication, Computing and Security*, ser. ICCCS '11, 2011, pp. 590–595. [Online]. Available: http://doi.acm.org/10.1145/1947940.1948063

[39] K. Plossl, T. Nowey, and C. Mletzko, "Towards a security architecture for vehicular ad hoc networks," in *The First International Conference on Availability, Reliability and Security, 2006. ARES 2006.*, 2006, p. 8 pp.

[40] F. Dotzer, "Privacy issues in vehicular ad hoc networks," in *Privacy Enhancing Technologies*, ser. Lecture Notes in Computer Science, G. Danezis and D. Martin, Eds. Springer Berlin / Heidelberg, 2006, vol. 3856, pp. 197–209.

[41] S. Eichler, "A security architecture concept for vehicular network nodes," in *6th International Conference on Information, Communications Signal Processing, 2007.*, 2007, pp. 1 –5.

[42] C.-T. Li, M.-S. Hwang, and Y.-P. Chu, "A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks," *Comput. Commun.*, vol. 31, pp. 2803–2814, July 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1389585.1389893

[43] G. Yan, S. Olariu, and M. C. Weigle, "Providing vanet security through active position detection," *Computer Communications*, vol. 31, no. 12, pp. 2883 – 2897, 2008, mobility Protocols for ITS/VANET. [Online]. Available: http://www.sciencedirect.com/science/article/B6TYP-4RP0MMN-4/2/f17aac95bc4acf0c94e8e9d2c999dd29

[44] T. Suen and A. Yasinsac, "Peer identification in wireless and sensor networks using signal properties," in *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005.*, 2005, pp. 825–833.

[45] K. Ibrahim, M. C. Weigle, and G. Yan, "Light-weight laser-aided position verification for CASCADE," in *Proceedings of the International Conference on Wireless Access in Vehicular Environments (WAVE)*, Dearborn, MI, December 2008.

[46] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "Efficient and robust pseudonymous authentication in vanet," in *VANET '07: Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*, 2007, pp. 19–28.

[47] A. Wasef, R. Lu, X. Lin, and X. Shen, "Complementing public key infrastructure to secure vehicular ad hoc networks [security and privacy in emerging wireless networks]," *IEEE Wireless Communications*, vol. 17, no. 5, pp. 22–28, 2010.

[48] M. Abuelela, S. Olariu, and M. Weigle, "Notice: An architecture for the notification of traffic incidents," *Proceedings of the IEEE 67th Vehicular Technology Conference, VTC Spring*, pp. 3001–3005, May 2008.

[49] Z.-X. Li, X.-M. Yang, and Z. Li, "Application of cement-based piezoelectric sensors for monitoring traffic flows," *Journal of Transportation Engineering*, vol. 132, no. 7, pp. 565–573, Jul. 2006.

[50] K. Plößl, T. Nowey, and C. Mletzko, "Towards a security architecture for vehicular ad hoc networks," in *Proceedings of The First International Conference on Availability, Reliability and Security (ARES 2006)*, 2006, pp. 374–381.

[51] National Highway Traffic Safety Administration, "2006 ruling," http://www.injurysciences.com/Documents/NHTSA%20Issues%20Final%20Rules%20for%20Automotive%20EDRs.pdf, 2006.

[52] T. W. Kaufmann, "Steering system with lane keeping integration," *2008 Patent 20080189012*, August 2008. [Online]. Available: http://www.freepatentsonline.com/20080189012.html

[53] C. Longjard, P. Kumsawat, K. Attakitmongkol, and A. Srikaew, "Automatic lane detection and navigation using pattern matching mode," in *Proceedings of the 7th WSEAS International Conference on Signal, Speech and Image Processing*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2007, pp. 44–49. [Online]. Available: http://portal.acm.org/citation.cfm?id=1364486.1364494

[54] W. Zhu, F. Liu, Z. Li, X. Wang, and S. Zhang, "A vision based lane detection and tracking algorithm in automatic drive," in *Proceedings of the 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application - Volume 01*, ser. PACIIA '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 799–803. [Online]. Available: http://dx.doi.org/10.1109/PACIIA.2008.155

[55] M. Abuelela, S. Olariu, and G. Yan, "Enhancing automatic incident detection techniques through vehicle to infrastructure communication," in *11th International IEEE Conference on Intelligent Transportation Systems, 2008. ITSC 2008.*, Oct. 2008, pp. 447 –452.

[56] M. Abuelela and S. Olariu, "Automatic incident detection in vanets: A bayesian approach," in *IEEE 69th Vehicular Technology Conference, 2009. VTC Spring 2009.*, April 2009, pp. 1 –5.

[57] M. Abuelela, S. Olariu, M. Cetin, and D. Rawat, "Enhancing automatic incident detection using vehicular communications," in *IEEE 70th Vehicular Technology Conference Fall (VTC 2009-Fall)*, Sept. 2009, pp. 1 –5.

[58] M. Abuelela and S. Olariu, "A probabilistic technique for detecting permanent and temporary incidents in vanets," in *Proceedings of the 6th International Workshop on Intelligent Transportation (WIT)*, Hamburg, Germany, mar 2009.

[59] K. B. Korb. and A. E. Nicholson., *bayesian artificial intelligent.* Chapman and Hall/CRC, 2003.

[60] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, p. 1805, 2000. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0002177

[61] T. Little and A. Agarwal, "A new information propagation scheme for vehicular networks," *Proceedings of the 3rd International Conference on Mobile Systems, Appplications and Service, (MobySys)*, September 2005.

[62] N. Wisitpongphan, F. Bai, P. Mudalige, and O. Tonguz, "On the routing problem in disconnected vehicular ad-hoc networks," *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 2291–2295, May 2007.

[63] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, and K. Sezaki, "CARAVAN: Providing location privacy for VANET," in *Proceedings of the Workshop on Embedded Security in Cars (ESCAR)*, Cologne, Germany, 2005.

[64] F. Picconi, N. Ravi, M. Gruteser, and L. Iftode, "Probabilistic validation of aggregated data in vehicular ad-hoc networks," in *Proceedings of the ACM Workshop on Vehicular Ad Hoc Networks (VANET)*, Sep. 2006.

[65] M. Treibner, A. Kesting, and D. Helbing, "Understanding widely scattered traffic flows, the capacity drop and platoon as effects of variance driven time gaps," *Physical Review E*, vol. 74, pp. 1–10, May 2006.

[66] E. Graham, L. Knuth, and O. Patashnik, *Concrete Mathematics.* Addison-Wesley, Reading, Massachusetts, 1989.

[67] M. Abuelela and S. Olariu, "Traffic-adaptive packet relaying in vanet," in *VANET '07: Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks.* New York, NY, USA: ACM, 2007, pp. 77–78.

[68] J. Yin, T. ElBatt, G. Yeung, B. Ryu, S. Habermax, H. Krishnan, and T. Talty, "Performance evaluation of safety applications over DSRC vehicular ad hoc networks," in *Proceedings of the ACM Workshop on Vehicular Ad Hoc Networks (VANET)*, Oct. 2004.

[69] R. Stoleru and J. Stankovic, "Probability grid: a location estimation scheme for wireless sensor networks," *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE SECON 2004. 2004*, pp. 430–438, Oct. 2004.

[70] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: modeling a three-tier architecture for sparse sensor networks," *Proceedings of the First IEEE. 2003 IEEE International Workshop on Sensor Network Protocols and Applications, 2003.*, pp. 30–41, May 2003.

[71] Interoperability Laboratory for Security in Ad-Hoc Networks, University of Luxembourg, "802.11 (wifi) simulator for ad-hoc networks," http://wiki.uni.lu/secan-lab/SECAN-LAB.html, 2008.

[72] M. Abuelela and S. Olariu, "Soda: A smart opportunistic data dissemination approach for vanet," in *Proceedings of the 6th International Workshop on Intelligent Transportation (WIT)*, Hamburg, Germany, mar 2009.

[73] F. Granelli, G. Boato, D. Kliazovich, and G. Vernazza, "Enhanced gpsr routing in multi-hop vehicular communications through movement awareness," *Communications Letters, IEEE*, vol. 11, no. 10, pp. 781 –783, October 2007.

[74] R. Augustin and K.-J. Büscher, "Characteristics of the cox-distribution," *SIGMETRICS Perform. Eval. Rev.*, vol. 12, pp. 22–32, December 1982. [Online]. Available: http://doi.acm.org/10.1145/1041818.1041821

[75] M. Abuelela, S. Olariu, and K. Ibrahim, "A secure and privacy aware data dissemination for the notification of traffic incidents," in *IEEE 69th Vehicular Technology Conference, 2009. VTC Spring 2009.*, April 2009, pp. 1 –5.

VITA

Mahmoud Abuelela
Department of Computer Science
Old Dominion University
Norfolk, VA 23529

Mahmoud received his Bachelors Degree in Computer Science with honors from Alexandria University in 1999. In 2005, he received the Master Degree in Computer Science from Alexandria University. His Master's thesis was focused on enhancing materialized view maintenance in data warehousing environment. After that he joined the PhD program in the Computer Science Department in Old Dominion University where he decided to pursue his dissertation under the supervision of Prof. Stephan Olariu in vehicular ad-hoc networks.

Mahmoud's main research focused on proposing data dissemination techniques for vehicular ad-hoc networks. He also worked on proposing novel automatic incident techniques using vehicular ad-hoc networks.

Mahmoud's research interests include a wide variety of interesting networks related topics such as incident detection, efficient data dissemination, security, privacy, P2P and cloud computing.

Recently, Mahmoud and his advisor, Prof. Olariu, are initiating the idea of providing cloud computing resources on the top of vehicular ad-hoc network to take advantage of the low utilized storage and computing resources expected to exist in the vehicles.

Mahmoud's research work and ideas were well received by the VANET research community, with over 20 publications in various IEEE and ACM journals, conferences and workshops at the time of writing this thesis.

Typeset using LaTeX.