Traffic monitoring and incident detection through VANETs

Mario De Felice, Andrea Baiocchi, Francesca Cuomo, Gaetano Fusco and Chiara Colombo
University of Roma “Sapienza”, Rome, Italy
Email: name.surname@uniroma1.it

Abstract—Road traffic monitoring is one of the key applications in the Intelligent Transport System field. New technologies are now provided in this field and among the most relevant ones there is the DSRC (Dedicated Short Range Communication) set of protocols and standards where vehicles wirelessly communicate. In this paper, we deal with the application of Vehicular Ad-Hoc Networks to road traffic monitoring and we present the design of two distributed protocols based on the DSRC. A realistic simulation of a main expressway in Rome, Italy, is implemented and the performances of the two proposed monitoring methodologies are evaluated in case of regular traffic conditions and in case of a car accident. In both cases the protocols are able to capture in a very quick time (few seconds) the current traffic conditions even on a quite long road of about 70 km. A discussion about the impact of the market penetration rate of the on-board DSRC devices on the protocols performance is also provided.

Index Terms—Traffic measurement; VANET; dissemination protocol; accident detection

I. INTRODUCTION

One challenging application in the field of the Intelligent Transport Systems (ITS) is the traffic monitoring to develop information strategies that help travelers in their travel choice. Advanced Traveler Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS) may take advantage by the use of new technologies like Vehicular Ad-Hoc Networks (VANETs) that allow Dedicated Short Range Communications (DSRC) of vehicles in the 5.9 GHz band. Applications of VANETs to ATIS and ATMS provide these systems with a monitoring subsystem that exploits equipped vehicles as probes in the traffic stream and, at the same time, with a communication system that allows vehicles to exchange information with each other regarding current traffic speed or any other useful message on traffic conditions. In this work we propose two road traffic monitoring approaches based on VANETs. To gauge the feasibility of the approach, we have set up a realistic simulation of a main expressway in Rome, Italy, calibrated by exploiting information provided by an extensive sample of floating car data. Joint vehicular mobility and communication network simulations have been carried out, pointing out that the designed protocols can collect quite accurate average speed estimates for different sections of the road with a very limited effort. No monitoring infrastructure is needed, except for a single Road Side Unit (RSU) on a road span of almost 70 km; collection times are within few seconds, the required bitrate is very low (0.08 kbps) and relative errors are about 2% on the long-term average (7% if real time, in the worst case) with one approach; with the other one, we have higher accuracy (mostly 100%), collection times in about 40s, but a higher bit rate (∼ 40-50 kbps). Furthermore, we tested our approaches when an incident occurs on two of the 3 lanes of the highway. The protocols are both able to quickly report the event at the RSU side and in particular, one of them can do it real time. The impact of the market penetration rate of VANET devices on the traffic monitoring capabilities in case of incident is also provided, such that our approach still works when not all the vehicles are equipped with DSRC devices. The rest of the paper is organized as following. Section II gives a short survey of related works, then the two VANET communication and monitoring protocols are defined in Section III. The considered test case and the performance evaluation are presented in Sections V and IV, respectively. Concluding remarks are given in Section VI.

II. RELATED WORK

In the last years, a broad literature arose from VANET applications to different ITS subsystems, ranging from ADAS (cooperative collision warning), to ATMS/ATIS (virtual traffic lights, incident detections) and ATIS (advanced speed control, route guidance). In the following, we will focus on traffic monitoring and incident detection. Traditional incident detection algorithms developed in 70s like California and Payne’s [1] are based on occupancy measures at fixed road sections and try to recognize anomalous conditions by comparing upstream and downstream traffic density measures; that is, by observing the effects of the incident on traffic flow. Statistical algorithms detect significant differences between observed detector data and traffic characteristics predicted by prior probabilities [2], time-series and filtering analysis ([3][4]).

In the following years, more complex mathematical methods were introduced. McMaster algorithm applied the Catastrophe theory to recognize an abrupt interruption of the regular pattern in the flow-speed-occupancy space [5]. In the latest years, several methods were developed that apply artificial Intelligence techniques, including neural networks ([6][7]), fuzzy logic [8] and a combination of these two techniques ([9][10]).

Techniques based on the use of surveillance cameras take advantage of temporal variations of traffic characteristics in addition to spatial ones. These methods provide their best
effectiveness when stopped vehicles are in the surveillance area. The performances of the aforementioned algorithms depend on the balance of the thresholds chosen for incident identification. If thresholds that provide false alarm lower than 2% are chosen, the mean time to detect an incident ranges from about 1 minute to 6-8 minutes [11].

Several authors have shown that vehicle-generated data can provide reliable estimates of traffic conditions, including identifying incidents and congestion ([12][13]).

Dia and Thomas [14] devised a neural network algorithm that uses a fusion of data from loop detectors and probe vehicles identified by fixed devices. Larger advantages can be obtained, however, by collecting data directly from floating cars; that is, vehicles that move on the road network. Geisler et al. [15] presented an evaluation framework for traffic information systems based on data streams from mobile phones and applied it in two case studies, namely queue-end detection and traffic state estimation, simulated by VISSIM microsimulation software in the idealistic case of two highways links of 5 km length.

Ma et al. [16] envisioned a real-time travel incident detection method based on Artificial Intelligence paradigms that exploits vehicle-infrastructure integration. They analyzed their method through a simulation experiment on a small freeway network simulated by Paramics, which showed that the proposed framework outperforms traditional incident detection methods based on point traffic measures like California algorithm. Leontiadis et al. [17] investigated opportunities for implementing VANET-based ITS systems that performs a fully distributed information system and compared different traffic prediction algorithms in a simulation of downtown area of Portland, Oregon (USA).

While traditional techniques mostly rely on monitoring infrastructure placed on the road, in this paper we use a different approach, where monitoring infrastructure is almost totally missing (one infrastructure node on 68 km), and instead we exploit DSRC features like multi-hop [18]. We can thus yield faster results for real time traffic monitoring and car incident detection, where we scale from the order of minutes, in the previous literature, to few seconds in our study.

III. THE VANET MONITORING SYSTEM

Several solutions [19] have been proposed for message dissemination in a VANET with the goal of extending the coverage area reached by message flows distributed by a RSU (or an On Board Unit-OBU) thanks to vehicle-to-vehicle multi-hop communications. A key issue is to avoid the broadcast storm problem [20]. To this aim, we exploit a backbone based approach [21] where a set of vehicles that are situated along the road are dynamically self elected to act as relay nodes. Many protocols in this category are based on timers. All nodes receiving a given packet delay the relaying of that packet in order to choose the node in the best position and to inhibit the other nodes. The DDT [22] approach is based on a simple and quite effective way for reducing redundant rebroadcasts and the consequent medium contentions and collisions. It assigns the tasks of relaying packets in each neighborhood solely to the receiver that is located farthest away from the sender. DDT assumes that each node is equipped with a GPS and an embedded system connected to vehicle sensor devices, so that both node position and speed can be estimated. Let $P_A$ and $v_A$ denote the vehicle position (coordinates) and speed, respectively.

Leveraging the DDT idea, we define a so called Distance Based Forwarding (DBF) dissemination protocol. DBF operations is based on the Forwarding and Inhibition Rules. Let $A$ be a vehicle located at a point of coordinates $P_A$, forwarding a message with sequence number $k$ and hop count $h$, at time $t$. Any other vehicle $V$, at position $P_V$ within range of $A$, receives the message sent by $A$. Let $k_V$ the biggest sequence number already seen and completely dealt with by $V$.

- **Forwarding Rule.** By checking that $k \leq k_V$, $V$ can discard old or duplicated messages. If the message is new (i.e., $k > k_V$) $V$ schedules its forwarding by setting a timer $T_{V,k} = T_{max}(1 - PVF/A/\max)$, where $T_{max}$ is the maximum forwarding delay, $R_{max}$ is an upper bound of the coverage radius of OBU transceivers, and $\max$ is the distance between $V$ and $A$. Hence, $V$ schedules the forwarding of the message at time $t + T_{V,k}$ with sequence number $k$ and hop count $h + 1$.

- **Inhibition Rule.** If during the time interval $(t, t + T_{V,k})$, $V$ receives another message with sequence number $k$ and hop count $h'$, $V$ checks that $h' > h + 1$. In that case, $V$ drops the scheduled message and will not forward it. Otherwise, no inhibition takes place.

In this paper we define a road traffic monitoring protocol by developing a new logic based on the DBF one.

A. Monitoring by sampling vehicles

In the following we define a collection logic called SAME (SAmpled Measurement Estimation). Let us consider two RSUs located along a road span, RSU$_a$ and RSU$_b$. RSU$_a$ originates a stream of messages, issuing one call for measurement collection (cmnc) message every time interval $T_{RSU}$. The cmnc message is passed over from vehicle to vehicle by using a DBF logic, until it reaches RSU$_b$, that is the final sink of the collected measurements. Messages are issued by the originating RSU with a sequence number, incremented by 1 at each new message, and with $h$ set to 0.

The logic defined to monitor the traffic on the road with the VANET is implemented by having a distributed collection of a vector $m = (m_1, \ldots, m_n)$ where the $i$th element is a 3-tuple $m_i = (v_i, P_i, t_i)$. Here $v_i$ is the speed of the $i$-th sampled vehicle, $P_i$ its geographical coordinates and $t_i$ a timestamp. The size of $m$ depends on the number of vehicles that collect this information, that is the number of vehicles forwarding the message (or the number of hops) from the source RSU up to the sink RSU. The message initially issued by RSU$_a$ has an empty payload. Each sampled vehicle, namely a vehicle acting as a forwarding node according to DBF, appends its 3-tuple to the current payload of the message. When the message reaches
RSU
, it carries all \( n \) collected measurements. \( n \) is related to the average hop length and to the length of the monitored road span. A fast polling is possible, since each forwarding hop takes a time in the order of \( ms \) and typical hop lengths can be several hundred meters. The traveling speed of the monitoring messages can be thus in the order of 50 km/s, three orders of magnitude more than vehicle speed. Measurements are collected at the RSU
 and can be used to track the traffic average speed and density in each sector of the monitored road span. Polling is repeated every \( T_{RSU} \) seconds. \( T_{RSU} \) is to be chosen so as to detect vehicular traffic variations quickly and reliably (that entails collecting tens of samples over a time scale comparable with that of vehicle motion, namely minutes) and to keep the collecting protocol performance near to sampled measurement collection, where samples naturally to sampled measurement collection, where samples are taken according to the distance covered at each hop. An exhaustive collection of vehicle measurements can be achieved by devising a variation of such protocol; the biggest difference is that all vehicles are assumed to send their own message, instead of having only a selection of relaying vehicles that forward the measurement message. We propose the Timer-based Ordered Measurement Estimation (TOME) protocol.

The protocol is based on an initiator station, namely an RSU, polling the system with \( cme \) messages with period \( T_{RSU} \). \( T_{RSU} \) is much bigger than the single hop time, so as to avoid MAC level congestion. The Collection Road Segment (CRS) is defined as the span of road for which the protocol aims at giving a single estimate of the average speed and density of vehicles. Let \( d_0 \) denote the target CRS length. For each CRS a node plays the role of the CRS Initiator (CRSI), i.e., it resets the message accumulator fields and starts off a new CRS.

Messages are made up of three parts: i) a header, that lists the sequence number \( k \), the identity of the sending node \( S \) and its position \( P_S \); ii) the list \( L \) of the measurements consolidated from previous CRSs; iii) the current CRS record \( \{\text{CRSI.id, CRSI.pos, AS, NV}\} \). \text{CRSI.id} and \text{CRSI.pos} are the CRSI identity and position of the vehicle playing the role of initiator of the CRS. \( AS \) is an accumulator field, whose value is the current sum of the speeds of vehicles that have been passing the message to one another within the CRS. \( NV \) is an integer counter, carrying the number of vehicles that have contributed to the accumulator \( AS \). We denote formally a message as \( M = [k, S, P_S; L; \{\text{CRSI.id, CRSI.pos, AS, NV}\}] \).

At the end of the CRS \( s \), the attained values \( AS_s \) and \( NV_s \) are frozen and stored into the list, i.e., \( L = L||\{(P_s, AS_s, NV_s)\} \), where \( P_s \) is the position of the first node of CRS \( s \). The notation \( || \) means appending the record appearing on the right to the list denoted at the left of the sign. The consolidation of the record appended to the list is done by the node that starts the new CRS \( s+1 \), namely the node in position \( P_s+1 \).

Each node maintains state variables. For node \( A \) we define: i) \( k_A \), the biggest sequence number read from incoming messages; ii) the current measurement tuple \( T_A = \{\text{CRSI.id}_A, \text{CRSI.pos}_A, \text{AS}_A, \text{NV}_A\} \), as updated after the reception of the most recent relevant TOME protocol message; iii) \( T_{MA} \) the current value of the timer of \( A \).

The basic idea of TOME is that each vehicle receiving a collection message will schedule the emission of its own message copy. The copy includes any accumulated speed sum heard of until emission time, plus the contribution of the vehicle’s own speed. The only exception is when a vehicle deems to be a new CRSI. In that case the most recent contribution, received before the scheduled emission time, is frozen into the list and a new accumulation starts off.

Let us consider a node \( B \) with state \( k_B \) and \( T_B = \{X, P_X, v, n\} \), when it receives the message \( M = [k, S, P_S; L; \{Y, P_Y, V, N\}] \).

If \( k < k_A \), the new message is ignored, as an old, out of date message.

If \( k = k_A \), \( A \) checks if \( Y = X \). In that case, \( A \) has received a message from a vehicle inside the same CRS as \( A \) is. So, if \( N > n \), \( A \) updates its state as follows: \( T_A \leftarrow \{X, P_X, V, N\} \) and reschedules its timer as \( T_{MA} = T_{max}P_SP_A/R_{max} \). If instead it is \( N \leq n \), \( A \) does nothing and ignores the received message.

If \( k > k_A \), but it is \( Y \neq X \), then \( A \) checks whether \( P_YP_A < P_XP_A \) and \( P_YP_X < P_XP_A \). In that case \( A \) reassigns itself to the CRS initiated by \( Y \) and updates its tuple to \( T_A = \{Y, P_Y, V, N\} \) and timer to \( T_{MA} = T_{max}P_SP_A/R_{max} \). In case it is found that \( P_YP_A \geq P_XP_A \) or \( P_YP_X \geq P_XP_A \), \( A \) does nothing and ignores the message.

If \( k > k_A \), a new measurement cycle is being run and \( A \) updates its state as \( k_A = k \). As for the tuple \( S_A \), the key point is to check whether \( A \) has to start a new CRS or it belongs to the current CRS. Hence \( A \) check if \( P_YP_A > d_0 \). In that case, \( A \) sets \( T_A = \{A, P_A, 0, 0\} \). If instead it is \( P_YP_A \leq d_0 \), \( A \) sets \( T_A = \{Y, P_Y, V, N\} \). In any case, a countdown timer is started with the initial value \( T_{MA} = T_{max}P_SP_A/R_{max} \).

Finally, when the timer \( T_{MA} \) expires and the state of \( A \) is \( k_A \) and \( \{X, P_X, v, n\} \), \( A \) sends out a message \( M = [k_A, A, P_A; L; \{X, P_X, v + v_X, n + 1\}] \), where \( v_A \) is the current speed of \( A \).

An example of the operation of TOME is given in Figure 1. The blue circles represent CRSIs, while the \( \times \) signs denote intermediate vehicles. Timers are highlighted as vertical bars. The black part of the bar represents the timer duration.

It can be noted that timer values are kept as chosen when

\( P \) are frozen and stored into the list, i.e., \( L = L||\{(P_s, AS_s, NV_s)\} \), where \( P_s \) is the position of the first node of CRS \( s \). The notation \( || \) means appending the record appearing on the right to the list denoted at the left of the sign. The consolidation of the record appended to the list is done by the node that starts the new CRS \( s+1 \), namely the node in position \( P_s+1 \).

Each node maintains state variables. For node \( A \) we define: i) \( k_A \), the biggest sequence number read from incoming messages; ii) the current measurement tuple \( T_A = \{\text{CRSI.id}_A, \text{CRSI.pos}_A, \text{AS}_A, \text{NV}_A\} \), as updated after the reception of the most recent relevant TOME protocol message; iii) \( T_{MA} \) the current value of the timer of \( A \).

The basic idea of TOME is that each vehicle receiving a collection message will schedule the emission of its own message copy. The copy includes any accumulated speed sum heard of until emission time, plus the contribution of the vehicle’s own speed. The only exception is when a vehicle deems to be a new CRSI. In that case the most recent contribution, received before the scheduled emission time, is frozen into the list and a new accumulation starts off.

Let us consider a node \( B \) with state \( k_B \) and \( T_B = \{X, P_X, v, n\} \), when it receives the message \( M = [k, S, P_S|\{(Y, P_Y, V, N)\}] \).

If \( k < k_A \), the new message is ignored, as an old, out of date message.

If \( k = k_A \), \( A \) checks if \( Y = X \). In that case, \( A \) has received a message from a vehicle inside the same CRS as \( A \) is. So, if \( N > n \), \( A \) updates its state as follows: \( T_A \leftarrow \{X, P_X, V, N\} \) and reschedules its timer as \( T_{MA} = T_{max}P_SP_A/R_{max} \). If instead it is \( N \leq n \), \( A \) does nothing and ignores the received message.

If \( k > k_A \), but it is \( Y \neq X \), then \( A \) checks whether \( P_YP_A < P_XP_A \) and \( P_YP_X < P_XP_A \). In that case \( A \) reassigns itself to the CRS initiated by \( Y \) and updates its tuple to \( T_A = \{Y, P_Y, V, N\} \) and timer to \( T_{MA} = T_{max}P_SP_A/R_{max} \). In case it is found that \( P_YP_A \geq P_XP_A \) or \( P_YP_X \geq P_XP_A \), \( A \) does nothing and ignores the message.

If \( k > k_A \), a new measurement cycle is being run and \( A \) updates its state as \( k_A = k \). As for the tuple \( S_A \), the key point is to check whether \( A \) has to start a new CRS or it belongs to the current CRS. Hence \( A \) check if \( P_YP_A > d_0 \). In that case, \( A \) sets \( T_A = \{A, P_A, 0, 0\} \). If instead it is \( P_YP_A \leq d_0 \), \( A \) sets \( T_A = \{Y, P_Y, V, N\} \). In any case, a countdown timer is started with the initial value \( T_{MA} = T_{max}P_SP_A/R_{max} \).

Finally, when the timer \( T_{MA} \) expires and the state of \( A \) is \( k_A \) and \( \{X, P_X, v, n\} \), \( A \) sends out a message \( M = [k_A, A, P_A; L; \{X, P_X, v + v_X, n + 1\}] \), where \( v_A \) is the current speed of \( A \).

An example of the operation of TOME is given in Figure 1. The blue circles represent CRSIs, while the \( \times \) signs denote intermediate vehicles. Timers are highlighted as vertical bars. The black part of the bar represents the timer duration.
the receiving node from the CRSI. For each CRS, the RSU receives at last the collected sum of speeds \( V \) in the segment and the overall number of vehicles \( N \), hence it can estimate the average speed \( V/N \) and the average vehicle density as \( \delta = N/d_0 \). Given \( V \) and \( \delta \), the average flow of the MCS is derived as \( \phi = V\delta \).

The speed of propagation of the measurement collection message is \( R/T_{max} \). For typical values of the involved parameters, the order of magnitude of this speed is tens of thousands of km/h, so that it is from two to three orders of magnitude bigger than vehicle speed. As a matter of example, with \( T_{max} = 500 \) ms, \( R_{max} = 830 \) m, to cover the GRA (about 68 km), it takes about \( 41 \) sec.

As for the size of the measurement collection message, we assume that a position can be represented with 16 bits, the accumulated average speed as a floating number with 32 bits and the number of vehicles per segment as a 16 bit integer. Then 80 bits = 10 bytes are sufficient for a single record of the list. The number of MCSs depends on \( d_0 \) and the length \( L \) of the observed road, namely it is \( \lceil L/d_0 \rceil \). If the measurement collection process involves no more than 100 road segments, 1000 bytes are enough to hold the whole list. Therefore, it can be expected that the message length is at most somewhat more than 1000 bytes, which is well compatible with message sizes sent over the DSRC interface.

IV. APPLICATION OF MONITORING PROTOCOLS TO A URBAN HIGHWAY

In order to set up a realistic simulation, we exploit a large database of about 104 millions of GPS traces collected by about 80,000 equipped vehicles that made about 9 millions trips during the month of May 2010 in the metropolitan area of Roma (Italy). We focused on a subset of 50,220 vehicles on a 68 km long ring-shaped expressway embracing the city called GRA, which collects and distributes long-haul traffic entering and exiting from the city (see Fig. 2). Each vehicle sends its information record every 30 seconds. A variety of information is provided by each record, including the vehicle ID, geographical coordinates, speed and quality of GPS signal. Our FCD (Floating Car Data) has been cleaned of the records with low GPS signal quality in order to consider only highly reliable data.

In order to analyse the data, we have divided the GRA in 29 different segments of length \( L_j \), \( j = 1, \ldots, 29 \), where the main exits from the GRA highway are the starting and ending points of each segment. Vehicles were then divided in two sets according to their traffic direction: clockwise and counterclockwise. Four different time periods of four hours each have been considered, starting from 7 am until 11 pm. Inter-vehicle distance distribution and speed distribution were obtained for each of the four time periods. This analysis showed that the highest density of vehicles is in the time period between 3 pm and 7 pm, which has the largest number of detected vehicles (9732 vehicles).

To set up a realistic mobility simulation of the urban traffic on the GRA, the available data sample have been inferred to the universe of vehicles by assuming a random uniform sampling. Let \( \Delta t \) be the sampling interval (30 sec in our study), \( v_i \) the average speed of vehicle \( i \), \( n_j \) the estimated number of vehicles traveling in the \( j \)-th segment, \( g_j(t_1, t_2) \) the number of detected GPS signals in the \( j \)-th segment during the observation time interval \([t_1, t_2] \), \( L_j \) the length of the \( j \)-th segment, \( a \) the probe vehicles penetration rate (\( a \approx 2.3\% \) in our study) and \( q_j \) the estimated flow on the \( j \)-th segment. Then:

\[
q_j = \sum_{i=1}^{n_j} v_i \Delta t / L_j
\]

and the resulting flow is \( q_j = \frac{n_j}{a(t_2 - t_1)} \). The above inference relations have been applied to the traffic flows in the peak period and have been used to enter into each segment \( j \) a flow of vehicles of mean intensity \( q_j \). The FCD have been used to estimate the OD matrix, where Origins and Destinations correspond to the exits of GRA. Vehicle attributes have been tuned so that the average speed values measured per lane and
We raised the speed of some vehicles, except that in a specific highway sector (#14), just to see the scenario. In the latter one, the flows are as in the regular traffic conditions, as given in the peak period identified in Section IV, based on the collected FCD on Rome GRA; ii) incident conditions, when an accident occurs; iii) mixed traffic, as the one sampled once per second. Between two consecutive sampling instants, NS2 are fed by the outcome of SUMO to give node positions, which are already built in NS2. Communication network simulations of VANET Layer protocols on top of the IEEE 802.11p MAC/PHY layers were implemented our VANET monitoring logics, as described in Section III, by adding code to NS2 [25] to realise the Forwarding Layer protocols on top of the IEEE 802.11p MAC/PHY layers already built in NS2. Communication network simulations of NS2 are fed by the outcome of SUMO to give node positions, sampled once per second. Between two consecutive sampling times, NS2 moves nodes according to linear uniform motion.

Since it will take years to deploy the VANET technology extensively, we assume different market penetration percentages of VANET DSRC radio equipment on board vehicles. Because of this, every new vehicle that is instantiated in our simulations, is equipped with this technology with probability \( p \in [0, 1] \), that represents the market penetration rate of the VANET technology. So, on average, only \( p \cdot 100\% \) of the vehicles are able to communicate with each other, while the others do not perform any transmission/reception operation, although influencing the vehicular micro-mobility simulated by SUMO.

We set up two simulation scenarios: i) regular traffic conditions, as given in the peak period identified in Section IV, based on the collected FCD on Rome GRA; ii) incident scenario. In the latter one, the flows are as in the regular scenario, except that in a specific highway sector (#14), just before time \( t = 2000 \text{ sec} \), we raised the speed of some vehicles (the dense high speed cloud in Fig. 3(d)) thus ending up with a car incident that obstructs two out of the three lanes of the road in the considered traveling direction, until the end of the simulation.

Tab. I lists the main simulation parameters.

**B. Traffic monitoring in regular conditions**

We assume regular traffic conditions on the considered urban highway and we feed vehicular traffic according to the statistics derived from measurements in the peak time interval, as explained in Section IV. A single RSU is placed as shown in Figure 2 and it acts contemporarily ascmc originator andsink. It sends out cmc with a time period \( T_{RSU} = 1 \text{ sec} \). Figure 3(a), 3(b) and 3(c) plot the average speed measured over the entire urban highway through SAME and TOME and as given by the simulation software SUMO (ground truth). Measurements arriving at the RSU after taking a full trip along the urban highway ring are averaged out over a time window of 5 sec (on top) and 1 sec (bottom). The three graphics assume different fractions of vehicles equipped with a VANET OBU, namely \( p = 1, 0.75 \) and 0.5.

By considering the top plots in Figures 3(a), 3(b) and 3(c), it is clear that TOME achieves a high accuracy degree in the estimation of the vehicle speed, since it collects speed samples from essentially all vehicles. Remarkably, also SAME turns out to be quite accurate, even though it just samples one vehicle every \( R_{max} \), which is one order of magnitude less vehicles than TOME. Overall, the estimate by SAME is based on about 90 samples per round. For the averaging time of 5 sec, at most about 450 vehicle samples are taken. This is apparently enough to provide an accurate estimation of the average speed in regular traffic conditions. The effect of limited availability of OBU equipment on board vehicles is that some messages get lost due to the temporary disconnection of the VANET chain along the ring highway. The disconnection extent over time grows as the market penetration rate gets smaller. During the presumably long time when VANET equipment is being deployed, heterogeneous networking, exploiting the cellular network should be a solution in order to prevent too frequent disconnections and allow a regular monitoring of the highway. Once OBU spreading overcomes sensitively 50% of vehicles, VANET alone can start providing a reliable means of collecting measurements. Another means to improve the VANET reliability is to deploy few more RSUs, even if it is reasonable the RSU density shall be much lower than cellular base station density.

These findings are confirmed by the bottom plots in Figures 3(a), 3(b) and 3(c), where the same quantities of the top graphs are plotted, except that the average is performed over 1 sec. It is evident that SAME tends to yield a less stable result than TOME, somewhat overestimating the true average value, but still it keeps the error below 10% in almost any condition or market penetration rate. Taking into account that we target real time vehicular traffic monitoring, we have to strike a balance between accuracy and delay. From our results, it appears that 5 sec can yield a good compromise.

Similar results are obtained by analyzing each highway sector, with more variability for SAME, due to the limited number of samples it collects (few units per sector, a sector being 2.4 km long on average).

A last consideration should be said about the required bitrate both for SAME and TOME: in the first case it is almost null (0.08 kbps), which yields the possibility of a permanent nearly invisible service, running on the control channel or on any non-dedicated sub-channel; in the second case, the bitrate is...
Fig. 3. Speed measures on the whole highway, when varying the market penetration rate of the system, in case of regular traffic conditions (a,b,c), with results aggregated every 5 sec (on top) and every second (bottom) and speed measures of sector #14 of the highway in case of incident (d,e,f)

about 56 kbps, better suited for a channel explicitly dedicated to traffic monitoring services.

C. Incident detection through VANET monitoring

In this section we analyze the behavior of our protocols when dealing with an incident (Figure 3(d), 3(e) and 3(f)).
These figures refer to only one sector of the GRA, namely sector #14, that is the one where the incident occurs. Moreover, SAME points are the individual speed samples taken from sampled vehicles within the tagged sector and transported by the protocol.

It is clear that in this case the market penetration rate is a very important factor, since after the accident occurs, the vehicle flow is strongly slowed down and becomes irregular in the interested sector. In terms of communications working, this means that connectivity is not always granted. In fact, even with $p = 1$, we have blank spots in Fig. 3(d), as for $2160 < t < 2200$ sec.

By looking at Fig. 3(d), we get a very clear information about what is going on from SAME. As soon as an incident occurs, immediately we observe two aspects:

1. higher variance in the monitored speed.
2. a sudden increase of (almost) zero speed vehicles.

These two aspects combined together immediately trigger an alarm, detecting the presence of an incident. The average speed of the vehicles from our ground truth and from TOME, instead, needs some time to drop down, because at the beginning the problem is very localized and the observed sector is 2 Km long, so vehicles that overcome the incident, accelerate and take the same speed they had before they encountered the incident, or even a higher one, because of the reduced density downstream the bottleneck formed by the stopped vehicles. This means that only after $50 – 100$ sec TOME really gets aware that an anomaly has occurred (it is stopped vehicles. This means that only after $100$ sec SAME takes one order of magnitude less time than TOME to detect an alarm, detecting the presence of an incident. The average speed of the vehicles from our ground truth and from TOME, instead, needs some time to drop down, because at the beginning the problem is very localized and the observed sector is 2 Km long, so vehicles that overcome the incident, accelerate and take the same speed they had before they encountered the incident, or even a higher one, because of the reduced density downstream the bottleneck formed by the stopped vehicles. This means that only after $50 – 100$ sec TOME really gets aware that an anomaly has occurred (it is possible to notice the descending slope after the incident in the figure). Moreover, the fact that SAME detects many zero speed values is a clear signal that a stable queue has been formed upstream a sudden bottleneck: this fact identifies an incident. As market penetration rate drops down, the capability to reveal an incident is delayed. Finally, it is important to notice that SAME takes one order of magnitude less time than TOME to collect and deliver such information (about 4 sec vs 41 sec respectively).

In order to help showing the SAME capability to reveal an incident, we can automate the incident detection process (e.g., by building an app that automatically yields an alert signal at the traffic monitoring center). The idea is based on the number of zero speed vehicles in a short time frame. Let us define $T_{observation}$, which represents the period of time between a cmc message and the following one. Say that $T_{observation} = 1$ sec, and let $z[k]$ be the number of speed samples that fall below a threshold that represents “blocked” vehicles at the $k$-th $T_{observation}$ (e.g., we could set a threshold at 10 or 20 km/h). Then, an alert is triggered if the following condition is verified:

$$ e + \sum_{i=k-4}^{k-5} z[i] < 2 \sum_{j=k-4}^{k} z[j] $$

where $e$ is a sensibility constant that we set equal to 5, needed to protect the system from false positives when only few vehicles slow down. This calculation is performed every time a new cmc message is received (thus every $T_{observation}$). By applying this procedure, we can get aware of the incident in 4 sec for $p = 1$, 103 sec for $p = 0.75$ and 115 sec for $p = 0.5$, transmission times included.

VI. CONCLUSION

In this work we defined and evaluated two traffic monitoring approaches that can be used leveraging the potentiality of VANETs. The protocols are based on a message exchange through a multi-hop path built on vehicles equipped with DSRC devices. To show the potentiality of these approaches we simulated a main expressway in Rome, Italy, through a road traffic model calibrated by exploiting information provided by an extensive sample of floating car data. Joint road traffic network and communication network simulations have been carried out, pointing out that the proposed protocols can collect quite accurate average speed estimates for different sections of the road with a very limited effort. No monitoring infrastructure is needed, except for a single Road Side Unit on a road span of almost 70 km. Collection times of the proposed techniques are within few seconds and suitably match with the data that would be acquired if all vehicles were able to provide their speeds to a centralized entity. In particular, one of them is very accurate, at the price of a slightly higher bit rate and information delivery time, while the other one is slightly less accurate, but with very low bit rate and fast delivery times. We also tested our approaches when an incident occurs on the highway. The protocols are both able to quickly report the event to the RSU and in particular, one of them can do it real time. An analysis of the market penetration rate of DSRC devices on the traffic monitoring capabilities is also provided and it highlights that the proposed approaches still work when not all the vehicles are equipped with DSRC devices.

REFERENCES


