

Multimedia Multimetric Map-Aware Routing Protocol to Send Video-Reporting Messages Over VANETs in Smart Cities

Ahmad Mohamad Mezher¹ and Mónica Aguilar Igartua

Abstract—One of the most important goals of vehicular ad hoc networks (VANETs) in smart cities is the efficient management of accidents, specially to prevent them. Our research lies on a promising smart service, which soon might be available in our cities. After the occurrence of an accident, a vehicle could make a light and short video of the situation and send it through the VANET till reaching an access point in the infrastructure of the city to alert the emergencies service (e.g., 911 or 112). With a video message, the level of seriousness of the accident could be better interpreted by the authorities (i.e., health care unit, police, ambulance drivers) than with a simple text message. In this way, vehicles could participate in reporting a situation in the city using the ad hoc network so it would be possible to have a quick reaction of the emergency units and even prevent further accidents. The deployment of an efficient routing protocol to manage video-reporting messages in VANETs has important benefits by enabling a fast warning of the incident, which potentially might save lives. To contribute with this goal, we propose a multimedia multimetric map-aware routing protocol to provide video-reporting messages over VANETs in smart cities. Furthermore, a realistic scenario is created by using real maps with SUMO including buildings that may interfere the signal between sender and receiver. Also, we use our REVsim tool that allows vehicles to avoid choosing vehicles behind buildings to be chosen as next forwarding nodes. Simulations show the benefits of our proposal, taking into account the mobility of the nodes and the presence of interfering buildings.

Index Terms—Building attenuation, realistic urban scenarios, smart cities, vehicular ad hoc networks, video-streaming services.

I. INTRODUCTION

A VEHICULAR ad hoc network (VANET) is an infrastructureless type of network where nodes are vehicles [3], [4]. VANETs are wireless networks that are emerging thanks to advances in wireless technologies and in the automotive industry. Vehicular networks are formed by moving vehicles equipped with wireless interfaces, as it is shown in Fig. 1. VANETs

Manuscript received April 2, 2016; revised October 7, 2016; accepted May 21, 2017. Date of publication June 16, 2017; date of current version December 14, 2017. This work was supported by the Spanish Government under Project TEC2014-54335-C4-1-R “INcident monitoRING In Smart COmmunities, QoS and Privacy” (INRISCO). The review of this paper was coordinated by the Guest Editors of the Connected Vehicle Series. (Corresponding author: Ahmad Mohamad Mezher.)

The authors are with the Department of Network Engineering, Universitat Politècnica de Catalunya, Barcelona 08034, Spain (e-mail: ahmad.mohamad@upc.edu; maguilar@entel.upc.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TVT.2017.2715719

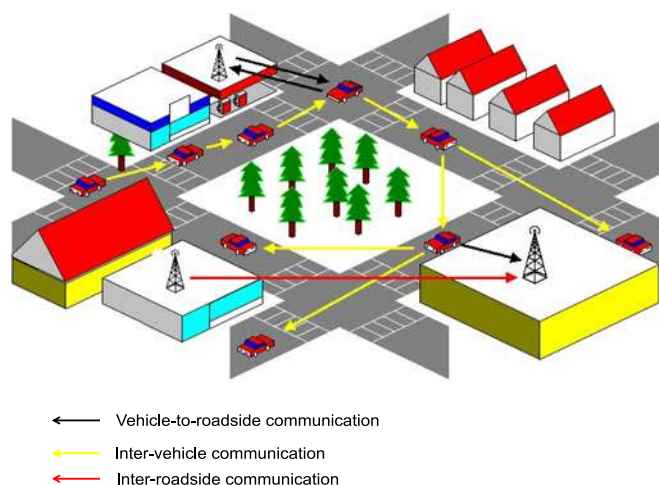


Fig. 1. Different types of communications in VANETs.

are considered as one of the most promising ad-hoc network technologies for real life applications, enabling communications among nearby vehicles as well as between vehicles and fixed equipment in the infrastructure also called roadside units (RSUs).

On the other hand, emerging smart city applications include new services that involve vehicles, drivers, pedestrians and city infrastructure. Smart city applications could be used in emergency situations that require the coordination and collaboration between implicated parts to, for instance, treat urgently wounded people who have suffered an accident. Emergency prevention and response are key issues for smart cities to face any kind of sudden incident such as traffic accidents, traffic jams, public transport delays, *etc.* For example, after the event of an accident, one of the involved vehicles could shoot a light and short video of the accident (with an external camera mounted on the vehicle) and send it through the VANET to alert the emergencies service (e.g., 911 or 112). With a video message, the level of seriousness of the accident could be better interpreted by the authorities (i.e., health care unit, police, ambulance drivers) than with a simple text message. The prevention and management of accidents is one of the most important goals in smart cities, and nowadays information and communication technologies (ICTs) in collaboration with citizens play an essential role in this field. In this way, vehicles could participate in reporting

a situation in the city using the VANET network so it would be possible to have a quick reaction of the emergency units that receive video-reporting warning messages about an accident. In this kind of situations, VANETs play an important role in smart cities, where vehicles are welcome to participate and interact with the management of the city. Diverse kinds of data could be gathered from the vehicles, e.g., need for health care after an incident or state of the traffic density. This way, the public administration would have an interactive network of vehicles who actively participate in the management of the city. It would be possible to have a quick answer to solve daily problems and help the emergency units in case of incidents (e.g., traffic accidents or traffic jams).

This article is an extension of our preliminary paper [2], in which we proposed REVSIM, a tool that helps us to attain realistic simulation results. REVSIM is able to detect the presence of obstacles in real maps so that each time a node (i.e., a vehicle) is going to send a packet, a check is done to ensure that no obstacles are found between the current and the next forwarding node; otherwise, the packet would be dropped. Furthermore, in this paper, we implement the use of the REVSIM tool in our forwarding algorithm, so that the forwarding operation is building aware and the current forwarding node can avoid vehicles behind buildings to be chosen as next forwarding nodes. In addition, in this work we present our proposal of a geographic multimedia multimetric map-aware routing protocol based on hop-by-hop forwarding decisions, which is building aware, for VANETs to send video-reporting messages in a smart city. Also, an algorithm to update the weights of the metrics dynamically throughout time is proposed in this work, so that those most decisive metrics are highlighted.

The special requirements and the unique characteristics of VANETs (e.g., special mobility patterns, short life links, rapid topology changes) generate challenges for the research community. Due to that, it is necessary to develop new routing protocols specially designed for VANETs that are able to provide the above mentioned video-reporting services. In this work, a new proposal of a multimetric geographical routing protocol for VANETs to transmit video-reporting messages is presented. Our proposal considers several quality of service (QoS) metrics to select the best next forwarding vehicle for each packet in each hop towards its destination. These metrics are properly weighted to obtain a multimetric score for each vehicle in the transmission range so that the current forwarding node can take the best next hop forwarding decision. In addition, the weights of the QoS metrics are self-configured. We have designed an algorithm to compute and update those weights throughout time so that nodes would be better classified according to the current state of the environment. In this way, each time the forwarding algorithm needs to classify nodes, a proper weight value for each metric will be updated so that the adaptive framework is able to self-configure and the best next forwarding node can be chosen. Simulation results show the benefits of our protocol in terms of average packet losses, average end-to-end delay, average jitter delay and Peak Signal to Noise Ratio (PSNR).

The rest of the paper is structured as follows. Section II includes some relevant related work. In Section III we

summarize the features of our java program REVSIM [2] that detects obstacles in a real map obtained from the OpenStreetMap [5]. Section IV presents our multimedia multimetric geographical routing protocol, the metrics, the way metrics are evaluated and how we compute the final multimetric score for each candidate neighbor to be the next forwarding node. Section V analytically describes how to compute and update the values of the weights of the metrics. Simulation results are shown and analysed in Section VI. Finally, conclusions and future work are given in Section VII.

II. RELATED WORK

Routing in VANETs is the process of selecting the best vehicle or vehicles in the network through which data will be forwarded. The best forwarding node is not necessarily the closest one to destination, although nodes usually are selected using the shortest path. Similar proposals of routing protocols for vehicular ad hoc networks close to our work can be classified in two categories: (a) geographical routing protocols for VANETs; and (b) routing protocols used to transmit video over VANETs. In the following we summarize some representative works related to our proposal in both categories.

- a) Regarding geographical routing protocols, many protocols were designed in the last years for VANETs. The work in [6] shows that the best routing protocols for VANETs are based on the information of the instantaneous locations of nodes. Geographic unicast protocols for VANETs can be classified into three categories [7]: (i) greedy, (ii) opportunistic, (iii) trajectory based. The most common approach in VANETs is the greedy strategy where a node forwards packets to its neighbor located closest to destination. Opportunistic strategies use the store-carry-and-forward technique to avoid dropping packets when no forwarding node is available. However, this strategy incurs high delays, which are not suitable for video-streaming of delay sensitive content. Using a trajectory-based strategy, a vehicle has more chances to be selected as a forwarding node if it is moving towards destination. On the other hand, GPSR [8] is a well-known greedy geographic unicast protocol designed for VANETs. Nodes are assumed to know their locations as well as the destination location. GPSR has two different modes to forward packets: greedy mode, which is used by default, and perimeter mode used when it is not possible to use the greedy mode. Several proposals have been presented in the literature to improve the basic GPSR. Movement prediction routing (MOPR) in [9] improves the routing process of GPSR by including a link stability concept to choose the best forwarding node. Authors in [10] present the I-GPSR (Improvement GPSR) that incorporates four metrics (distance to destination, vehicle density (VD), moving direction and vehicle speed) used to select the best forwarding node. In [11], the authors propose MMMR that uses four metrics (distance to destination, vehicle density, trajectory and available bandwidth) to select next forwarding nodes. MMMR is map-aware considering possible

obstacles, although only in Manhattan scenarios. Our proposal improves MMR since it is building-aware in any real map scenario, it also considers an additional metric of MAC losses and metrics are weighted dynamically.

- b) Regarding routing protocols used to transmit video-streaming in VANETs, only a few studies have been proposed so far. We highlight the work [12], where the authors present VIRTUS (Video Reactive Tracking-based UnicaSt), a proposal which extends the duration of the decision of nodes to forward packets from a single transmission moment to a time window. VIRTUS evaluates the suitability of a node to relay packets, including a density-aware relaying node selection in the video transmission process showing significant improvements. In [13], the authors propose a multipath solution for VANETs to provide a high quality video-streaming on VANETs. Due to the special characteristics of VANETs and the large amount of video data, extra interference and contention during the video-streaming is due to the redundancy of Forward Error Correction (FEC). To cope with this issue, authors use the TCP protocol to transmit the I-frames to ensure their transmissions and UDP protocol to transmit P and B frames to reduce the delay of the transmissions. I frames (Intra-coded frames) do not require other video frames to be decompressed; P frames (Predictive-coded frames) use data from previous I or P frames to be decompressed and are smaller than I frames; B frames (Bidirectionally-predictive-coded frames) use both previous and forward I or P frames for data reference and achieve the highest amount of data compression. Furthermore, authors use the node disjoint and link disjoint algorithms to further minimize the delay by transmitting I-frames and inter-frames (P and B frames) through separate paths. Simulations show that our multipath protocol 3MRP provides a higher video quality with an acceptable delay compared to other protocols.

Our multimetric routing protocol takes the three aforementioned forwarding aspects (greedy, opportunistic and trajectory) into consideration to select the best next node to forward video-reporting messages over VANETs. In addition, we use realistic scenarios by considering obstacles present in real maps when a forwarding node is going to be selected. Our proposal includes five metrics to optimize the selection of the best forwarding node in our geographic-based routing protocol. These metrics obtained from hello messages periodically interchanged by vehicles are:

- 1) *Distance to destination*: It is the distance between each candidate node and the destination node.
- 2) *Vehicle density*: It is computed as the number of vehicles in the neighbors' list of node I divided by $\pi \cdot TR_I^2$, being TR_I the transmission range of the candidate node I . For the sake of a simple model, we assume a circle to define the transmission range of the nodes.
- 3) *Trajectory*: It is computed as a comparison of the current distance of a candidate node to destination with a future distance between those same two nodes. This way we

detect if the candidate node is getting closer to or going away from destination.

- 4) *Available Bandwidth Estimation (ABE)*: The ABE [14] algorithm is used to estimate the available bandwidth in a link between two nodes, in our case between the current node and each candidate node.
- 5) *MAC layer losses*: Our routing protocol uses the packet losses computed at the MAC layer as a kind of local feedback information.

Then, we weight those five metrics into a single multimetric score using firstly equal and secondly variable weights updated with an algorithm.

To the best of our knowledge, the aim to use several metrics dynamically weighted to distribute video reporting messages over VANETs in realistic urban scenarios, is novel. We test our proposal in a realistic urban scenario, using the NS-2 simulator [15], including real maps from OpenStreetMap [5], realistic mobility patterns with C4R [16] and a tool named REVsim [2] that we have developed to detect buildings efficiently. In the next section we have described the REVsim operation.

III. DETECTING OBSTACLES IN REAL MAPS

In this section, we give a brief summary of the main features of our program REVsim [2] which is able to detect obstacles at each time-stamp in real maps, e.g., taken from the OpenStreetMap [5].

A. Motivation

To trust the results of any performance evaluation of a new proposal using a network simulator, a realistic environment should be used. Buildings as obstacles present in a real map could attenuate or even block the signal. Due to this fact, it is extremely important to consider obstacles that can actually be found in any real city map. Therefore, two nodes that belong to the same transmission range will actually be able to communicate with each other depending on the presence or absence of obstacles between them. Common traffic simulators do not take into account this transcendent issue. These facts lead us to design a program called REVsim [2] to detect if two nodes within the same transmission range could actually interchange packets or not during simulation. REVsim is able to detect the presence of buildings in the cities that could avoid the communication between two vehicles. Furthermore, this detection is done efficiently and fast, as we will explain in the following.

B. Program Scheme

In Fig. 2, we illustrate a scheme of REVsim. Next, we explain all the input files, the input parameters and the output files of our tool.

C. Input Files

The program has two input files:

- 1) A file *.net.xml* that contains the output format provided by the C4R simulator [16] with the transformed coordinates

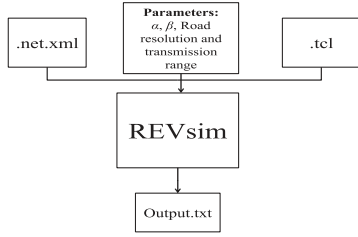


Fig. 2. REVsim tool to detect buildings efficiently in real maps.



Fig. 3. The α parameter to detect if two vehicles are in line of sight or not.

of the map downloaded from OpenStreetMap [5]. From that file we extract different kind of objects that define a map, such as a simple 2D point location, or a discrete line reaching a junction. These parameters with information about the map are extracted from the file *.net.xml* using java code and are saved in arrays in order to use them efficiently later by our program REVsim.

- 2) A file *.tcl* that contains the output of the C4R simulator. This file contains information about nodes, positions and their movements at each moment during simulation.

D. Input Parameters

We define four parameters to help the forwarding algorithm to determine if two vehicles in the same transmission range could establish a communication or not taking into account the presence of buildings between them. These parameters are α , β , road resolution and transmission range.

1) *The α Parameter of Our Algorithm to Detect Buildings:* As it is depicted in Fig. 3, the α parameter is defined as the angle that relates two roads (road A and road B in Fig. 3) starting from a common junction. An obstacle is assumed to be between those roads. The α parameter is used to determine till which angle α we can still consider that vehicles in road A can establish communication with vehicles in road B. In this way, the visibility between two vehicles is true or false depending on the value of α , which is calculated using (1).

$$\alpha = 180^\circ - \arccos\left(\frac{\vec{U} \cdot \vec{V}}{\|\vec{U}\| \cdot \|\vec{V}\|}\right) \quad (1)$$

where α is the angle between the two vectors \vec{U} and \vec{V} that define roads A and B, respectively. $\|\vec{U}\|$ is the magnitude of vector \vec{U} and $\|\vec{U}\| \cdot \|\vec{V}\|$ is the scalar product of both vectors. A node in street A can establish communication with another

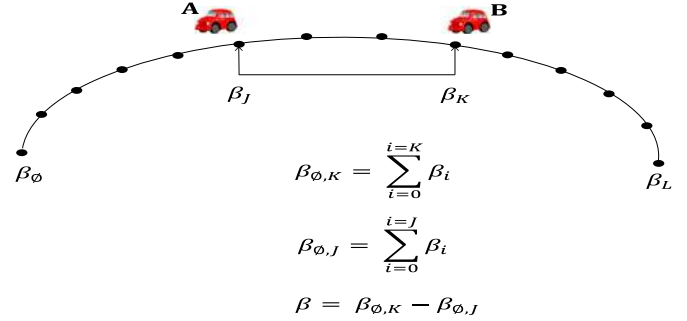


Fig. 4. An example of how the β parameter is computed between two vehicles A and B located on a curved road.

node in street B if $\alpha < \alpha_t$ where α_t is a threshold value for α to be chosen experimentally.

2) *The β Parameter of Our Algorithm to Detect Buildings:* The β parameter is defined to establish the relationship between two or more lines that are part of a same curved road. In this way, we determine the grade of curvature as a vehicle travels through the consecutive lines of a same curved road. Besides, β is used to efficiently determine the relationship of two related curved roads by means of the lines that compose both roads. By related roads we mean that any two nodes in those roads could establish communication.

Our algorithm begins analysing the value of α as a first step and then the value of β as a second step, locating the points (vehicles) in the map using lines instead of roads. We consider that roads are formed by lines so for each single point (vehicle) in the map, its associated line is determined. That is, the algorithm determines on which line of the road the vehicle is located.

Mathematically, β is the sum of a set of small betas obtained between two vehicles A and B (see Fig. 4), $\beta_{\phi, K} - \beta_{\phi, J}$, $\phi \leq J \leq K \leq L$, being J the line number of vehicle A, K the line number of vehicle B and L the total number of lines in that road. Those small betas define the curvature between two vehicles A and B, according to Fig. 4. The parameter β is calculated as seen in (2).

$$\beta = \sum_{i=\phi}^{i=K} \beta_i - \sum_{i=\phi}^{i=J} \beta_i. \quad (2)$$

Two vehicles in that road can establish communication with each other if $\beta < \beta_t$, where β_t is a threshold value for β .

3) *Road Resolution:* Every line of a road is sampled into discrete points that describe that line. This raises a trade-off between the processing time and the precision of the algorithm, knowing that increasing the number of samples per line means a higher precision in the location of vehicles, but also an increment in the processing time. In Fig. 5, we show two different values of road resolution where we can notice that the high road resolution generates more samples than the low road resolution, so the positions of vehicles are more accurate if the road resolution is higher.

4) *Transmission Range:* For every vehicle in the urban scenario, the algorithm only analyses the presence of buildings in the line formed with every neighbor vehicle within its transmission range. This way we reduce the analysis only for those



Fig. 5. Two different road resolutions.

TABLE I
BUILDING-AWARE OUTPUT FILE *Output.txt*

$D T N$
$b_{1,2,1}$
$b_{1,2,2}$
.....
$b_{1,2,T}$
$b_{1,3,1}$
$b_{1,3,2}$
.....
$b_{1,3,T}$
.....
$b_{1,N,T}$
$b_{2,3,1}$
$b_{2,3,2}$
.....
$b_{2,3,T}$
.....
$b_{2,N,T}$
.....
.....
$b_{N-1,N,1}$
$b_{N-1,N,2}$
.....
$b_{N-1,N,T}$

vehicles with which the vehicle could establish a communication. This is done for every simulation moment, analyzing the snapshot of every vehicles' location and the interaction with the buildings in the real map. It is important to highlight that this analysis is done off-line, once we select the piece of real map, from OpenStreetMap [5], that will feed the NS-2 simulator. This way our building-aware forwarding algorithm is not affected by any computational delay produced if the analysis were done online during simulations.

E. The Output Building-Aware File

As Fig. 2 shows, after REVsim processes all its inputs, it obtains an output file named *output.txt*, whose format is shown in Table I. During simulation the forwarding algorithm will look at this file to quickly find out if two nodes are in line of sight (LOS) or not in case of having a building between them. This will be done each time a node needs to select a next hop among its neighbors to forward a packet towards destination.

Algorithm 1: Calculate the position value of the REVsim output file for (*Source*, *Destination*) in a specific *Timestamp*.

Require: Input data (*Source*, *Destination*, Total simulation time, Number of nodes, *Timestamp*)

Require: $i = 1$, $Ipos = 0$ # $Ipos = \text{initial position}$

```

1: if  $Source > Destination$  then
2:    $tmp = Destination$ 
3:    $Destination = Source$ 
4:    $Source = tmp$ 
5: end if
6: if  $Source = 1$  then
7:    $Position = (Destination - Source - 1) * \text{Simulation time} + \text{Timestamp}$ 
8: end if
9: while  $i < Source$  do
10:   $Ipos = Ipos + (\text{Number of nodes} - i) * \text{Simulation time}$ 
11:   $Position = Ipos + (Destination - Source - 1) * \text{Simulation time} + \text{Timestamp}$ 
12:   $i = i + 1$ 
13: end while

```

The first row of the *output.txt* file contains the trio $D T N$, where D signals the beginning of the file, T is the simulation time in seconds and N is the number of nodes. These specific header information will help us to determine each value of the file (1 or 0) regarding each combination of two nodes at each timestamp of the simulation. Values of $b_{1,2,1}, b_{1,2,2}, \dots, b_{N-1,N,T}$ will be set to 0 or 1, where 0 means that both nodes cannot communicate each other due to the presence of a building between them, whereas 1 means that they can communicate. For example, $b_{1,2,3}$ represents the relation between nodes 1 and 2 in timestamp 3 second. If $b_{1,2,3} = 0$ means that node 2 cannot receive any packet from node 1 in timestamp 3 second. Conversely, if $b_{1,2,3} = 1$ means that it is possible that node 2 receives packets from node 1 in that moment. We can notice that $b_{1,2,T}$ is the last value of the relation between nodes 1 and 2 in the last time-stamp T (i.e., the simulation time). After $b_{1,2,T}$, the file follows with $b_{2,3,1}$ and not $b_{2,1,1}$ because the relation between nodes 2 and 1 in timestamp 1 (i.e., $b_{2,1,1}$) is the same as $b_{1,2,1}$ which is found above in the file, so there is no need to repeat it again. This format of output file allows a very fast look up of the $b_{i,j,k}$ value to know if there is a building ($b_{i,j,k} = 1$) or not ($b_{i,j,k} = 0$) between vehicles i and j in the moment k . This way our building-aware forwarding algorithm operates fast.

If the routing protocol needs to know during simulation if any two nodes can communicate or not, the algorithm can easily calculate the position of the binary answer of the output file *output.txt* using Algorithm 1 and go directly to the proper position to read the value. Algorithm 1 explains how to compute the row where the answer is found using simple information such as source node, destination node, number of nodes, simulation time and the specific time-stamp we want to check. Let us see a simple example: The simulation time is 10 seconds ($T = 10$), there are 10 nodes ($N = 10$) and we want to know if in time-stamp 3 second node 4 can receive packets from node 1 or not. For that, we need to go to find the value of $b_{1,4,3}$. As

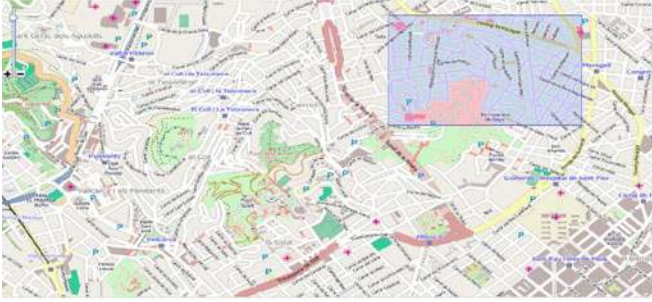


Fig. 6. Generic map of the city of Barcelona, Spain, used in our tests.

the source is 1, we use the equation written in line 7 and as a result, $Position = 23$. This means that we have to go directly to row 23 of the file *output.txt* to find the binary answer $b_{1,4,3}$. Depending on the binary answer, the receiver node 4 will be able to receive the packet ($b_{1,4,3} = 1$) sent by node 1 or will have to drop it ($b_{1,4,3} = 0$). This way, we emulate what would have happened actually in real life, so our simulations are more realistic and we can trust our results. Furthermore, our building-aware forwarding algorithm can prevent vehicles to forward packets to next hop vehicles located behind buildings. This will improve the performance of our proposal.

Moreover, using this efficient external file obtained offline for the specific simulation map, we save a lot of processing time which is an important factor when we are carrying out many simulations. It is important to highlight that with this proposal it is not necessary to check online during simulation if two nodes are in LOS or not, which would add delay to the simulation. With this method, it is only necessary to make a fast look-up to a text file previously obtained for the real map taken from OpenStreetMap.

To sum up, Algorithm 1 is used to compute a position in the file *output.txt*. That position corresponds to the binary answer of the communication checking between nodes *Source* and *Destination* at time *Timestamp*. If the positions of *Source* and *Destination* are not in increasing order, the algorithm interchanges the role of both nodes (lines 1 to 4). After that, the initial position, $Ipos$, is computed for the values corresponding to node *Source* (line 10). Finally, the algorithm computes the value $Position$ (line 7 or 11) associated with the *Destination* node at time *Timestamp*.

F. Tuning the REVsim Parameters for a Generic Scenario

To obtain the building-aware output file *output.txt* we need to tune the REVsim parameters α , β , road resolution and transmission range, so that the program is able to detect the presence of every building in the map that may block the signal between every pair of vehicles during the whole simulation. To do this, we have used a representative enough generic map from Barcelona (see Fig. 6) that includes either curved and straight roads, a highway and a Manhattan-style area with a grid of streets.

1) *Alpha Threshold*: To obtain which is the proper range of values for the threshold α_t , we carried out simulations with different values of α_t , using many snapshots of the vehicles'

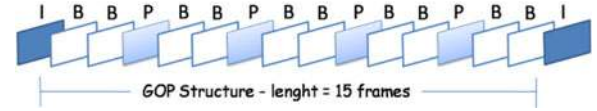


Fig. 7. Typical MPEG-2 GoP structure with 15 frames per GoP.

positions in different time-stamps. We analysed if all the vehicles in a specific road would see or not other vehicles in other roads within their transmission range throughout simulation. For each α_t tested, we concluded if using that threshold α_t the blocking buildings would be detected or not. After many tests [17], $\alpha_t = 20^\circ$ showed to be an optimal value for different generic city maps like the one shown in Fig. 6. Therefore, any α angle lower than 20° means that nodes found on those roads will be able to send and receive packets without any problem caused by a blocking building.

2) *Beta Threshold*: After many simulations [17], we obtained that $\beta_t = 60^\circ$ was the optimal threshold value below which all the blocking buildings located in curved roads were detected. Basically, β depends on how long and how curved is the road. The longer and curvier the road, the higher the value of β .

3) *Road Resolution*: After making many tests varying the road resolution value, we got that using a road resolution equal to 1m produced a good trade-off between precision and processing time.

4) *Transmission Range*: Here, we just put the transmission range value of each vehicle. This value will allow us to make a pre-filter step of the number of nodes to be analysed, since it has no sense to check nodes that are outside the transmission range of the node under study.

Concluding this section, our REVsim [2] tool allows our building-aware forwarding algorithm to quickly find out which are the actual neighbors of a node with which the node is in LOS (i.e., both nodes could actually communicate) in every moment. Our REVsim tool is available at [18]. In the following, we will introduce our proposed routing protocol whose goal is to choose the best next forwarding node among the node's neighbors in LOS.

IV. MULTIMEDIA MULTIMETRIC MAP-AWARE ROUTING PROTOCOL (3MRP)

In this section we present our proposal of geographic multimedia multimetric map-aware routing protocol based on hop-by-hop building aware forwarding decisions.

A. Basics of the General Framework

Our novel geographical routing protocol for VANETs is based on the GPSR protocol to find the best next forwarding node in a hop-by-hop scheme from source to destination. Video is distributed using RTP (Real-time Transport Protocol) over UDP as transport protocols. Our system uses a layered MPEG-2 VBR coding of the video flow, which is formed by sets of frames, 15 in our case, called GoP (Groups of Pictures), see Fig. 7. A GoP has three types of frames: I, P and B, and has a unique

TABLE II
IEEE 802.11p ACCESS CATEGORIES

AC in 802.11p	CW_{\min}	CW_{\max}	AIFS
0	15	1023	9
1	15	1023	6
2	7	15	3
3	3	7	2

frame-pattern in a video repeated in each GoP. I (Intra) frames encode spatial redundancy, they form the base layer, provide a basic video quality and carry the most important information for the decoding process at the receiving side. The whole GoP would be lost if the corresponding I frame were not available at decoding time. P (Predicted) and B (Bi-directional) frames carry differential information from preceding (P) or preceding and posterior (B) frames, respectively. Considering these characteristics, we assign different priorities to the video frames according to their importance within the video flow. Therefore, I frames should have the highest priority (AC3), P frames the medium priority (AC2) and B frames the lowest one (AC1), as shown in Table II. This table shows the values for each access category (AC) of the minimum contention window (CW_{\min}) and maximum (CW_{\max}), and the AIFS (Arbitration InterFrame Space) period. Basically, the lower those values, the sooner the AC tries to access the common medium so the higher is the priority.

B. Motivation of Our Routing Protocol Design

In a VANET, nodes are vehicles that move along roads, potentially at high speed, following transit rules, direction of streets, respecting traffic lights and also the presence of buildings and other vehicles. The vehicle density in VANETs constantly changes depending on environmental conditions such as area, time of the day and day of the week. Thus, it is difficult to establish and maintain end-to-end communication paths between sources and destinations as it is traditionally done in MANETs. Our aim is to design a proper data forwarding mechanism to transmit video-reporting messages considering real scenarios and the special constraints of VANETs.

In this work, we propose a new routing protocol for VANETs in realistic urban scenarios called 3MRP (Multimedia Multimetric Map-aware Routing Protocol). 3MRP seeks to improve the next forwarding node decision based on five metrics. We weight those five metrics to finally obtain a multimetric score associated to each neighbor node in LOS that is a candidate to be the next forwarding node. The weight calculation is self-configured and able to adapt to the changing environment conditions in real time.

In our proposal we can distinguish five processes: routing, signalling, evaluation of metrics and forwarding decision. We explain each one in the following.

C. 3MRP Routing

Our routing proposal 3MRP includes a forwarding decision similar to the one used in GPSR [8], although we use a

TABLE III
FORMAT OF THE NEW HELLO MESSAGES (NHM)

ID	x	y	v_x	v_y	L_{MAC}	ρ
----	-----	-----	-------	-------	-----------	--------

multimetric score instead of just the distance to destination. Basically, it consists in choosing the neighbor with the highest multimetric score. Besides, it also includes new improvements so that the multimetric score adapts to the current environment. Furthermore, we substitute the inefficient perimeter mode of GPSR by the use of a local buffer.

First, we have included in our proposal the REVsim [2] tool described in Section III. REVsim tackles the important issue of checking which neighbors cannot actually establish communication with the considered node due to any obstacle and excludes them from the final list of real neighbors in LOS. This avoids sending packets to an unreachable node located behind a building. The neighbors' list will only include those nodes that are in LOS with the current node. When no forwarding node is found, the packet will be stored temporarily in a local buffer during a maximum of 2 seconds. A time-out value is set because higher delays will not be acceptable for video-streaming content.

Afterwards, the decision of the next forwarding node is done based on the combination of five metrics as it is detailed in Section IV-E.

D. 3MRP Signalling

3MRP gathers information from the periodic interchange of hello messages (HM) that nodes use to announce their presence to their neighbors in transmission range. To obtain precise location information about each node, without introducing much extra overhead, new fields were added in those HM. The format of our new hello messages (NHM) is presented in Table III, and they include these new fields, which are updated at the moment of sending the current hello message:

- 1) *ID*: It is the identifier of each node.
- 2) *Position* (x, y): This field contains the x and y axis positions that represent the geographic position of each node.
- 3) *Velocity* (v_x, v_y): This field represents the speed v_x in x axis and the speed v_y in y axis. Each node updates its own speed from two consecutive position points taken at times t_j and t_{j+1} :

$$v_x = \frac{x_{j+1} - x_j}{t_{j+1} - t_j}, \quad v_y = \frac{y_{j+1} - y_j}{t_{j+1} - t_j}, \quad j \geq 0. \quad (3)$$

In our case, $t_{j+1} - t_j$ is equal to 1 second.

- 1) *MAC layer losses* (L_{MAC}): Each node calculates the data losses in the MAC layer and this value is sent in the hello message as a kind of local feedback information. This value is updated every 10 seconds of simulation to track the recent state of the neighborhood.
- 2) *Density* (ρ): This field represents the number of neighbors within transmission range divided by $\pi \cdot TR^2$, being TR the transmission range of that node.

When a node receives a hello message from a neighbor in transmission range, the node stores the moment of reception

Algorithm 2: Updating the neighbors' list.

Require: A new hello message received with these parameters: ID, x , y , v_x , v_y , L_{MAC} , ρ .

```

1: # node is in LOS
2: if output value of REVSIM == 1 then
3:   if Neighbor already is in the neighbors' list then
4:     Update neighbor information
5:   else
6:     if Neighbor is not in the neighbors' list then
7:       Add node in the neighbors' list
8:     end if
9:   end if
10: else
11:   # node is not in LOS
12:   Ignore hello message
13: end if

```

TABLE IV
ADDITIONAL INFORMATION PER NODE IN THE NEIGHBORS' LIST

Neighbor Ngh	First NHM time	No. NHM	Last NHM time
----------------	----------------	---------	---------------

and updates all the values shown in Table III in its neighbors' list. This is done following Algorithm 2. If a hello message is received from a neighbor, the algorithm first checks using REVSIM if that node is in LOS before including or updating the information in the neighbors' list. If the node is not in LOS, it is discarded. To keep the neighbors' list updated and having only nodes that actually are in transmission range, nodes remain in a neighbors' list during twice the interval between consecutive hello messages, i.e., during 2 sec. Similarly, when a node receives a hello message from a new neighbor (i.e., a neighbor not registered in its list), it has first to check if that node is in LOS before adding it in the neighbors' list. To do this, a check is done using the output file *output.txt* from our Java program REVSIM to see if this candidate neighbor could actually receive a packet if no obstacles are found between them. If this condition is not fulfilled, that neighbor will not be included in the neighbors' list.

The sending period of hello messages could be smaller to obtain more accuracy in the composition of the neighbors' list, although a higher signaling traffic could produce an increase in packet collisions. By default, the sending period of NHM is set to 1 second and the results are good.

If the output binary value $b_{i,j,k}$ from the tool REVSIM is 1 (i.e., no obstacles are found between both nodes) and the neighbor is already found in the neighbors' list, then the algorithm updates the neighbor information depicted in Table III (Lines 1 to 4 in Algorithm 2). If the output binary value $b_{i,j,k}$ is 1 and the neighbor is not in the neighbors' list, then we add the node in the neighbors' list (Lines 5 to 7). If the output binary value $b_{i,j,k}$ is 0, we ignore this hello message because it means that an obstacle is found between both nodes (Lines 10 to 12).

The neighbors' list includes the data sent in hello messages (see Table III) and the data shown in Table IV. For each neighbor

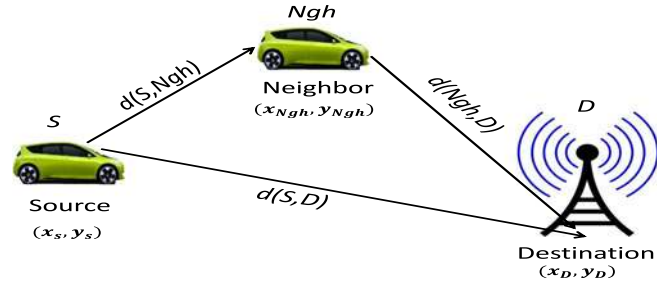


Fig. 8. Distances $d(S, D)$ from source S to destination D , $d(S, Ngh)$ from source S to a neighbor Ngh and $d(Ngh, D)$ from a neighbor Ngh to destination D .

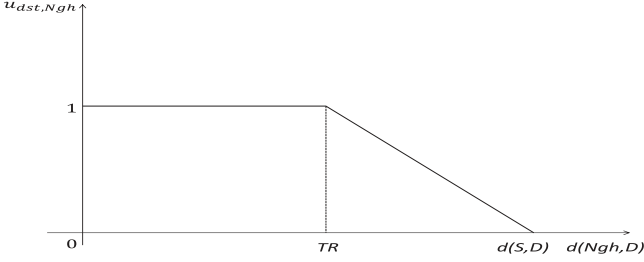
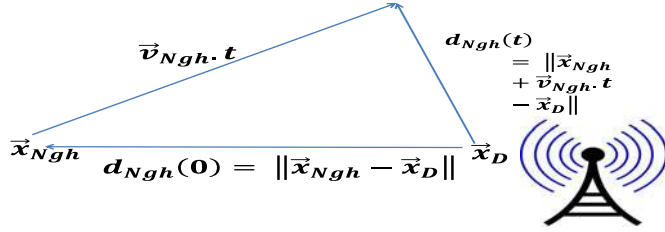
Ngh , we store the moment when the last hello message arrived (*Last NHM time* in Table IV). This is done to estimate the future position of that neighbor node, as it is explained in the next section. Also, we store the moment when the first hello message arrived (*First NHM time*) and the total number of hello messages received (*No. NHM*). These values will be used to estimate the available bandwidth using a metric explained in the next section.

E. Design of Routing Metrics for 3MRP

In this section we detail the design of each one of the five metrics included in our routing protocol 3MRP. The use of these metrics improves the selection of the next forwarding node. The five metrics considered are: distance to destination, trajectory of the vehicles, nodes density, MAC losses and available bandwidth. According to the adhoc principle of using only local information (i.e., infrastructureless operation) nodes will use those five metrics gathered from the hello messages (NHM) of the next hop candidates in their neighborhood to take the decision of the best next forwarding node. The five metrics are described in the following.

Distance: Geographic routing protocols forward packets hop-by-hop to their destination. In many routing protocols, the next forwarding hop is the closest neighbor to destination. Geographic protocols use the geographic information of every node to take forwarding decisions. We assume that all vehicles know their own position, the destination's position (x_D, y_D) as well as the positions of all their neighbors from periodic hello messages where nodes include their own position. Therefore, the position (x_{Ngh}, y_{Ngh}) of each neighbor Ngh can be obtained. The Euclidian distance $d(Ngh, D)$ from each neighbor node Ngh to destination D can be computed using (4), where d_{Ngh} is the distance of each neighbor Ngh to destination D , according to Fig. 8.

We have designed (5) to compute the metric of the distance, $u_{dst, Ngh}$, for each neighbor node Ngh . In this case, the shorter the value of the distance $d(Ngh, D)$, the better. That is, we prefer a neighbor as close as possible to destination. Besides, all neighbors whose distance to destination (i.e., $d(Ngh, D)$) is lower than the transmission range (TR), should have the maximum value of the metric because in this case all those neighbors are the last hop towards destination D . Due to that, we have designed the metric of the distance $u_{dst, Ngh}$ as shown in Fig. 9

Fig. 9. Distance metric $u_{dst, N_{gh}}$ for node N_{gh} .Fig. 10. Trajectory of node N_{gh} towards the access point (AP) destination.

and described in (5). While $d(N_{gh}, D) < TR$, $u_{dst, N_{gh}} = 1$ and when $d(N_{gh}, D) \geq TR$, $u_{dst, N_{gh}}$ decreases linearly till a minimum value of $d(N_{gh}, D) = d(S, D)$ which means that neighbor N_{gh} and source S are the same node. We can observe from Fig. 9 that $0 \leq u_{dst, N_{gh}} \leq 1$.

$$d(N_{gh}, D) = \|\vec{x}_{N_{gh}} - \vec{x}_D\|$$

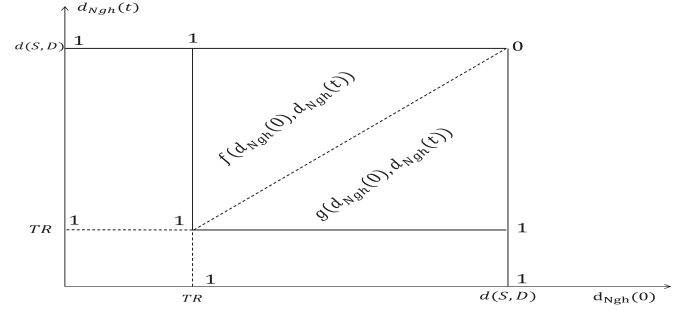
$$= \sqrt{(x_{N_{gh}} - x_D)^2 + (y_{N_{gh}} - y_D)^2} \quad (4)$$

$$u_{dst, N_{gh}} = \begin{cases} \frac{-d(N_{gh}, D)}{d(S, D) - TR} + \frac{d(S, D)}{d(S, D) - TR}, & \text{if } d(N_{gh}, D) \geq TR \\ 1, & \text{if } d(N_{gh}, D) < TR \end{cases} \quad (5)$$

Trajectory: The trajectory of vehicles in VANETs is a very important metric that might help to select a suitable next forwarding node that moves towards destination. We compute the trajectory of a node as a function of the current and future distances of that node to destination using the v_x and v_y velocities. This helps to determine the trajectory of that vehicle and as a consequence to detect if the node is getting closer or going away from the destination node. The aim of this metric is to avoid that the source could take wrong forwarding decisions based only on the distance and send packets to vehicles that were actually going away from destination, which could make packet losses increase as a consequence. Due to that, taking the moving direction of vehicles into account to take forwarding decisions is an important benefit for VANETs.

We obtain the trajectory metric $u_{trj, N_{gh}}$ of a candidate neighbor node N_{gh} using a future distance $d_{N_{gh}}(t)$ to destination of that node in the t moment and $d_{N_{gh}}(0) = d$ which is its current distance to destination. See (6) and Fig. 10 to see the meaning of the trajectory metric.

The distance $d_{N_{gh}}(t)$ is computed by estimating the future position of that candidate neighbor node N_{gh} using its speed according to (7).

Fig. 11. Projection of the function $u_{trj, N_{gh}}$ in $(d_{N_{gh}}(0), d_{N_{gh}}(t))$ plane.

The speed of the node, $v_{N_{gh}}$, helps us to give a higher score to nodes that sooner will be closer to destination (i.e., the AP). The idea is that with a higher speed, nodes moving towards destination may arrive sooner to destination since the distance to destination decreases faster. We compute the $u_{trj, N_{gh}}$ metric using (8).

$$d_{N_{gh}}(0) = \|\vec{x}_{N_{gh}} - \vec{x}_D\| \quad (6)$$

$$d_{N_{gh}}(t) = \|\vec{x}_{N_{gh}} + \vec{v}_{N_{gh}} \cdot t - \vec{x}_D\| \quad (7)$$

$$u_{trj, N_{gh}} =$$

$$\begin{cases} 1, & \text{if } d_{N_{gh}}(0) < TR \\ 1, & \text{if } d_{N_{gh}}(0) > TR \\ & \text{and } d_{N_{gh}}(t) < TR \\ f(d_{N_{gh}}(0), d_{N_{gh}}(t)), & \text{if } d_{N_{gh}}(0), d_{N_{gh}}(t) > TR \\ & \text{and } d_{N_{gh}}(0) > d_{N_{gh}}(t) \\ g(d_{N_{gh}}(0), d_{N_{gh}}(t)), & \text{if } d_{N_{gh}}(0), d_{N_{gh}}(t) > TR \\ & \text{and } d_{N_{gh}}(0) < d_{N_{gh}}(t) \end{cases} \quad (8)$$

where

$$f(d_{N_{gh}}(0), d_{N_{gh}}(t)) = \frac{d_{N_{gh}}(t) - d(S, D)}{TR - d(S, D)} \quad (9)$$

$$g(d_{N_{gh}}(0), d_{N_{gh}}(t)) = \frac{d_{N_{gh}}(0) - d(S, D)}{TR - d(S, D)} \quad (10)$$

$d_{N_{gh}}(t)$ is an estimation of the future position of node N_{gh} at moment t , as it is depicted in Fig. 10. $v_{N_{gh}}$ is the average speed of the evaluated neighbor N_{gh} with respect to destination and computed from two consecutive positions. $\vec{x}_{N_{gh}}$ and \vec{x}_D are neighbor and destination positions, respectively. Notice that $\|\cdot\|$ refers to the module function of a vector.

The trajectory metric increases when the vehicle moves towards destination, and decreases when the vehicle moves away from destination, as depicted in (8). As the distance metric, the trajectory metric $u_{trj, N_{gh}}$ has a range between 0 and 1. Fig. 11 is the projection of function $u_{trj, N_{gh}}$ in the $(d_{N_{gh}}(0), d_{N_{gh}}(t))$ plane. We can observe that if $d_{N_{gh}}(0) < TR$, this means that the neighbor is within the transmission range of destination and as a consequence we should give $u_{trj, N_{gh}}$ its highest value (i.e., 1) regardless of the value of $d_{N_{gh}}(t)$. In addition, if $d_{N_{gh}}(0) > TR$ and $d_{N_{gh}}(t) < TR$, this means that the future distance is within the transmission range of destination and as a consequence we should also give $u_{trj, N_{gh}}$ its highest value (i.e., 1). Finally, if $d_{N_{gh}}(0)$ and $d_{N_{gh}}(t)$ are higher than TR , we must

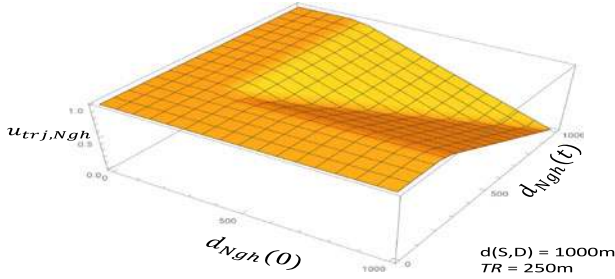


Fig. 12. Representation of the trajectory metric $u_{trj, Ngh}$ for $TR = 250$ m and $d(S, D) = 1000$ m.

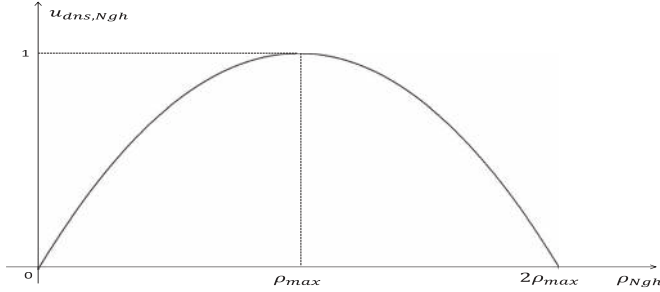


Fig. 13. Designed function for the vehicles density' metric $u_{dns, Ngh}$.

further analyse the subcases of $d_{Ngh}(0)$ being higher or lower than $d_{Ngh}(t)$. In the subcase of $d_{Ngh}(0) > d_{Ngh}(t)$, this means that the future distance to destination is smaller than the current distance (i.e., neighbor is getting closer to destination D). This subcase is better than the subcase of $d_{Ngh}(0) < d_{Ngh}(t)$, which means that the future distance to destination is higher than the current distance (i.e., the neighbor is getting far from destination D). For these two subcases, we designed two functions f and g (see Fig. 11) that satisfy the conditions mentioned before.

To understand more Fig. 11, we have plotted in Fig. 12 a 3D graph example for metric $u_{trj, Ngh}$ where $TR = 250$ m and $d(S, D) = 1000$ m.

Vehicles' density: It is computed as the number of vehicles in the neighbors' list of each node at the moment of sending the current hello message (N_v), divided by the area within the transmission range ($\pi \cdot TR^2$) of that vehicle. The neighbors' list of a node is composed by vehicles found in its transmission range. Each node computes its density of nodes ρ_{Ngh} using (11) and includes it in the next hello message.

The algorithm gives a higher score when the neighbor node Ngh has a higher value of ρ_{Ngh} . Nodes with a denser area in the transmission range will have more possibilities to forward the packet to a next node. This is true until reaching a maximum nodes' density ρ_{max} , above which the very high number of vehicles in the surrounding area of the node increases the collisions' probability. We set ρ_{max} to 200 vehicles/km², according to our simulations and other research works like [19].

We have designed a concave function for the density metric shown in Fig. 13. This function has its maximum at $\rho_{Ngh} = \rho_{max}$ and above ρ_{max} it decreases till $2\rho_{max}$ where again it reaches zero and keeps on zero for all $\rho_{Ngh} > 2\rho_{max}$.

Equation (12) describes how we calculate the vehicles' density metric $u_{dns, Ngh}$ as shown in Fig. 13. This way, we penalize those nodes whose number of neighbors in their transmission

range is above a threshold (i.e., $\rho_{Ngh} > \rho_{max}$). Following the same strategy as for the previous metrics, $0 \leq u_{dns, Ngh} \leq 1$.

$$\rho_{Ngh} = \frac{N_v}{\pi \cdot TR^2} \quad (11)$$

$$u_{dns, Ngh} = \begin{cases} \frac{-1}{\rho_{max}^2} \cdot \rho_{Ngh}^2 + \frac{2}{\rho_{max}} \cdot \rho_{Ngh}, & \text{if } \rho_{Ngh} \leq 2\rho_{max} \\ 0, & \text{if } \rho_{Ngh} > 2\rho_{max} \end{cases} \quad (12)$$

Available bandwidth: Video-reporting messages require a given amount of network resources (e.g., bandwidth) to achieve a good performance. To provide a certain level of QoS, we use an estimator of the available bandwidth in VANETs based on an approach developed for IEEE 802.11 networks called available bandwidth estimator (ABE) [14]. We use ABE as a metric in our forwarding decision algorithm to help in the selection of the best next forwarding node.

In the following, we briefly summarize the ABE operation to estimate the available bandwidth in a link between two nodes. A complete explanation of the authors can be found in [20]. Basically, each node estimates its percentage of idle time by sensing the common wireless medium. This value is included in its hello messages. The available bandwidth estimation of a wireless link in ABE uses the idle times of the emitter (T_e) and the receiver (T_r) of a link of capacity C . ABE computes the collision probability of the hello messages, named p_{hello} . The collision probability of packets of m bits, named p_m , is derived from the collision probability of the hello messages using (13), where N is the number of nodes in the scenario and s is the average speed of the nodes.

$$p(m, N, s) = f(m, N, s) \cdot p_{hello}(m, N, s) \quad (13)$$

This $f(m, N, s)$ was obtained in [14] by computing the Lagrange interpolating polynomial, taking pairs of values of packet losses and losses of hello messages from many simulations of a VANET scenario. In [14] we obtained the final expression for $f(m, N, s)$ for our urban scenarios, shown in (14).

$$f(m, N, s) = -7.475 \cdot 10^{-5} \cdot m - 8.983 \cdot 10^{-3} \cdot N - 1.428 \cdot 10^{-3} \cdot s + 1,984 \quad (14)$$

The additional overhead introduced by the binary exponential backoff mechanism was derived in [20]:

$$K = \frac{\text{DIFS} + \overline{\text{backoff}}}{T_m} \quad (15)$$

where T_m (in sec.) is the time separating the emission of two consecutive frames, DIFS (Distributed Coordination Function Interframe Space) [21] is a fixed interval so that nodes can have access to the medium if it is free for a time period longer than DIFS. Finally, $\overline{\text{backoff}}$ is the number of backoff slots decremented on average for a single frame. Merging the different mechanisms that impact the available bandwidth, the sender estimates the available bandwidth ABE on each neighbor' wireless link using (16) [14]. T_{Ngh} is T_r , being Ngh the receiver; and T_{Nan} is T_e , being Nan the node under analysis and emitter of the link.

$$\text{ABE}_{Ngh} = (1 - K) \cdot (1 - p(m, N, s)) \cdot T_{Nan} \cdot T_{Ngh} \cdot C \quad (16)$$

Finally, we divide ABE_{Ngh} by the link capacity C , obtaining (16) as the available bandwidth metric. Notice that $0 \leq u_{abe,Ngh} \leq 1$. A high value of $u_{abe,Ngh}$ means a high available bandwidth in the link formed with node Ngh .

$$u_{abe,Ngh} = ABE_{Ngh}/C \quad (17)$$

MAC layer losses: To compute the losses metric, we focus on the MAC layer instead of on the traditional routing layer since in VANETs we only manage local information instead of end-to-end information. This is an important principle in infrastructureless adhoc networks. We calculate the MAC layer losses and we use it in our routing protocol as a kind of local feedback within the neighborhood. Depending on this information, we can vary the decision of the best forwarding node. Furthermore, if we find that packet losses are very high, we can stop sending P and B frames and send only I frames, seeking to decrease the packet losses and improve the network performance.

$$u_{los,Ngh} = 1 - L_{MAC}(Ngh) \quad (18)$$

According to (18), $L_{MAC}(Ngh)$ is the MAC layer losses in the link formed between nodes Ngh and Nan . A value of $u_{los,Ngh}$ closer to 1 means that a low number of packets are lost, while a value closer to 0 means that losses are high. It is worth mentioning that MAC layer losses are locally computed at the node itself, i.e. by the wireless card driver at the vehicle.

F. 3MRP Forwarding Decision

3MRP takes hop-by-hop forwarding decisions based only on geographic information. When a node wants to send a packet it has first to choose the optimal next forwarding node from its list of actual neighbors.

When a sender node receives hello messages (HM) from its neighbors in transmission range, the node updates its neighbors' list with all those nodes in LOS that sent their HM with enough power to be considered as a neighbor. After that, the node evaluates and assigns a total multimetric qualification to each neighbor as a candidate for next forwarding node. As a first step, we assign the same weights (w_1, w_2, w_3, w_4, w_5) to each metric ($u_{dst,Ngh}, u_{trj,Ngh}, u_{dns,Ngh}, u_{abe,Ngh}, u_{los,Ngh}$), respectively in the multimetric score \bar{u}_{Ngh} of each neighbor Ngh .

$$\begin{aligned} \bar{u}_{Ngh} = & \sum_{i=1}^5 u_{i,Ngh} \cdot w_i = u_{dst,Ngh} \cdot w_1 \\ & + u_{trj,Ngh} \cdot w_2 + u_{dns,Ngh} \cdot w_3 \\ & + u_{abe,Ngh} \cdot w_4 + u_{los,Ngh} \cdot w_5 \end{aligned} \quad (19)$$

We finally obtain a multimetric score for each candidate node using (19). The final score varies between 0 and 5. The best next forwarding node is the neighbor with the highest multimetric value. We first give the same degree of importance to all the metrics, i.e., $w_i = 1/5$, $1 \leq i \leq 5$. In the next section, we propose an algorithm to compute self-configured weights of the metrics to dynamically update the scores of the candidate neighbor nodes..

V. ALGORITHM TO UPDATE THE WEIGHTS OF THE METRICS TO COMPUTE A MULTIMETRIC SCORE

A. Motivation

Several routing protocols for VANETs based on a hop-by-hop operation have been proposed in the literature, e.g., [8]–[11]. Some of them use several metrics to decide the next forwarding node. Usually, the weights of the metrics have the same value (i.e., the metrics play the same importance in the calculation of the multimetric score). Nonetheless, we claim that a better scheme could give each metric a variable weight depending on the current network conditions. As a consequence, nodes could be classified in a more accurate way. We foresee that since our algorithm needs to classify nodes from best to worst each time a packet must be forwarded, the weights of the metrics in our multimedia multimetric routing protocol could better take different values than $\frac{1}{5}$, being 5 the number of metrics.

We propose an algorithm to update the weights dynamically (i.e., re-calculate the multimetric score of neighbors) throughout time, so that those most decisive metrics are highlighted (their weights increase). This way, if a metric value in the nodes differs noticeably with respect to the average neighbors' value in that metric, we give more importance (i.e., a higher weight) to that metric. That is, we define as a decisive metric when the neighbor nodes have different values in that metric. This points out that its value may help the forwarding algorithm to better classify neighbor nodes. Conversely, a more constant metric value (i.e., all the neighbor nodes have roughly the same value) indicates that this metric is not so decisive to arrange the neighbor nodes, thus our algorithm gives a lower value to its weight. We call our algorithm to update the multimetric score of neighbors as Dynamic Self-configured Weights (DSW) algorithm.

B. DSW Algorithm Description

Each time a node needs to forward a packet, that node has to classify the nodes included in its neighbors' list (which are in LOS) from the best to the worst by using the multimetric score of (19). In (19), the weights w_1, w_2, \dots, w_5 are now computed by our algorithm so that they are dynamically updated depending on the current state of the neighborhood. The idea is to highlight those decisive metrics that can better help the current node under analysis (Nan) to choose the best next forwarding node among the nodes in its neighbors' list.

As mentioned before, we have five metrics: (u_1, u_2, u_3, u_4, u_5) = ($u_{dst,Ngh}, u_{trj,Ngh}, u_{dns,Ngh}, u_{abe,Ngh}, u_{los,Ngh}$) computed with (5), (8), (12), (17) and (18), respectively.

Let us denote R_m as the variation value for each metric m between time t_1 and time t_2 where $t_2 > t_1$, defined as:

$$R = \begin{cases} R_1 = \frac{[u_1(t_2) - A_1(t_2)] - [u_1(t_1) - A_1(t_1)]}{2} \\ R_2 = \frac{[u_2(t_2) - A_2(t_2)] - [u_2(t_1) - A_2(t_1)]}{2} \\ \vdots \\ R_5 = \frac{[u_5(t_2) - A_5(t_2)] - [u_5(t_1) - A_5(t_1)]}{2} \end{cases} \quad (20)$$



Fig. 14. Simulation scenario of Barcelona. It includes one emergency unit in the Hospital Clinic of Barcelona, named access point (AP) in the map.

where $0 \leq u_m(t_1), u_m(t_2) \leq 1$ and $0 \leq A_m(t_1), A_m(t_2) \leq 1$; $m \in [1, 5]$. $u_m(t_1)$ and $u_m(t_2)$ are the current scores of each metric m for times t_1 and t_2 , respectively. $A_m(t_1)$ and $A_m(t_2)$ are the average score values of each metric m computed for all the neighbors of the forwarding node. Now, if $R_m \leq 0$; this means that metric m is getting worst in the period of $(t_2 - t_1)$. As a consequence, R_m should be equal to zero.

This way, we will have a vector $R = [R_1, R_2, \dots, R_5]$. Suppose that the maximum value found in vector R is $R_{\max} = R_x$ where $x \in [1, 5]$. Now, we normalize vector R to be between 0 and 1 (i.e., $0 \leq R \leq 1$) and this new vector is named S .

$$S = \begin{cases} S_1 = \frac{R_1}{R_x} \\ \vdots \\ S_x = \frac{R_x}{R_x} = 1 \\ \vdots \\ S_5 = \frac{R_5}{R_x} \end{cases} \quad (21)$$

To be sure that the sum of weights of all the metrics is equal to one, we calculate the parameter ξ value using (22)

$$\xi = \frac{1}{\sum_{i=1}^5 S_i} \quad (22)$$

Next, the new normalized vector of weights W is:

$$W = \begin{cases} W_1 = S_1 \cdot \xi \\ \vdots \\ W_x = S_x \cdot \xi = \xi \\ \vdots \\ W_5 = S_5 \cdot \xi \end{cases} \quad (23)$$

Thus, instead of using equal weights (w_1, w_2, w_3, w_4, w_5) in (19), with $w_i = 1/5$, $1 \leq i \leq 5$, we use dynamic weights (W_1, W_2, W_3, W_4, W_5) computed as depicted in (23).

Finally, we must mention that in case that all the metrics for a specific node are getting worst, no preferences could be given to any metric and as a consequence we give all of them equal weights $W_i = \frac{1}{5}$. The probability for this special case to happen is very low.

TABLE V
SIMULATION SETTINGS OF THE VANET SCENARIO

Map Zone	Example District of Barcelona
Area	1700 × 580 m ²
Density of vehicles	50 vehicle/km ² (scenario 1), 100 vehicle/km ² (scenario 2)
Number of nodes	50 and 100 vehicles
Transmission range	250 m
Mobility generator	SUMO [1]/C4R [16]
MAC specification	IEEE 802.11p
Nominal bandwidth	12 Mbps
Simulation time	300 s
Video encoding	MPEG-2 VBR
Video bit rate	150 Kbps
Video sources	1
Video sequence sent	Traffic accidents [22]
Routing protocol	GPSR, 3MRP, 3MRP + DSF
Transport protocol	RTP/UDP
Maximum packet size	1500 Bytes
Weighting metric values	1/5 or dynamic
Queue sizes	50 packets

VI. SIMULATION RESULTS

We implemented our proposal in the open source network simulator NS-2 [15] where we conducted simulations to evaluate the benefits of our approach. We used a real city area obtained from the example district of Barcelona, Spain (see Fig. 14). Notice that our proposal works the same in any type of map, since it includes our RevSim tool to detect buildings in any map from OpenStreetMap. In order to simulate a realistic scenario, the CityMob for Roadmaps (C4R) [16] simulator was used to obtain the mobility model of the vehicles. C4R is a mobility generator that uses the Simulation of Urban Mobility (SUMO) engine [1]. Besides, C4R imports maps directly from OpenStreetMap [5] and generates NS-2 compatible files to specify the mobility model for the vehicles through the city along the whole simulation. Video flows are transmitted from a vehicle that suffers an accident to an access point (AP), set at the Hospital Clinic of Barcelona. The AP represents an emergency unit where the vehicle sends its video-reporting message upon the event of the traffic accident. The simulation settings of the scenario are shown in Table V. All the figures show confidence intervals (CI) of 90 percent obtained from 20 simulations per point, each simulation with an independent mobility scenario.

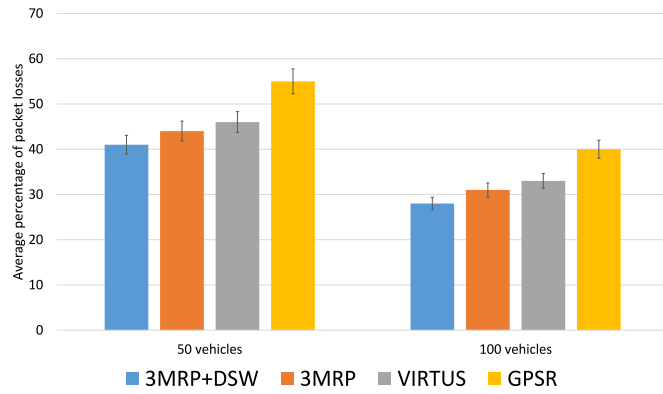


Fig. 15. Average percentage of packet losses.

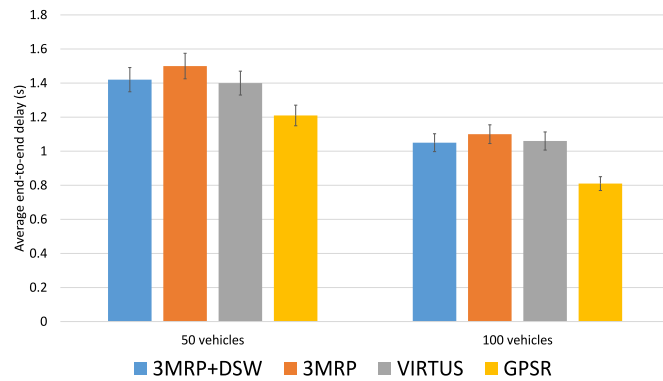


Fig. 16. Average end-to-end packet delay (sec).

We analyse the performance of our multimetric algorithm 3MRP compared to GPSR [8]. The simulation area is 1700 m × 580 m. We consider two densities of vehicles, 50 vehicles/km² (scenario 1) and 100 vehicles/km² (scenario 2) which are randomly positioned. The average speed of the vehicles is 20 km/h while the maximum speed is 50 km/h. The multimetric score used in the forwarding scheme of our proposed routing protocol, has equal weights (3MRP) or dynamic weights (3MRP+DSW). There is one fixed destination, an access point (AP), through which vehicles connect to the network to report traffic information, in this case a video-reporting message about a traffic accident.

Fig. 15 shows the average percentage of packet losses using our routing protocol (3MRP) with equal weights and using our dynamic self-configured weights (3MRP+DSW) scheme. We compare both proposals to GPSR and VIRTUS [12] and we present results for low and medium vehicles density. We can clearly notice how 3MRP+DSW obtains the best results in both scenarios reducing losses around 10% with respect to GPSR and around 6% with respect to VIRTUS [12]. This is due to the optimal selection of the next forwarding node based on the five proposed metrics with dynamic self-configured weights. Fig. 16 shows the results of the average packet delay. The delay is calculated based on those packets that successfully arrived at destination. Since GPSR takes the forwarding decision considering only distance, it obtains the lowest packet delay in both scenarios. However, GPSR shows the highest losses (see Fig. 15).

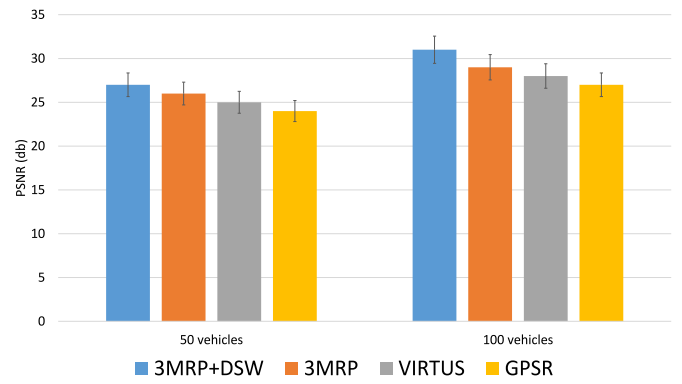


Fig. 17. Peak signal to noise ratio (PSNR).

This is because with GPSR, a lower number of packets arrived at destination and much of the lost packets traveled through a considerable number of hops before being dropped. Conversely, our proposal 3MRP is able to reduce packet losses although the delay slightly increases in about 0.3 seconds. Nonetheless, 3MRP+DSW reduces more the packet losses with a lower delay than the previous 3MRP.

Fig. 17 depicts the peak signal-to-noise ratio (PSNR) obtained for GPSR, 3MRP and 3MRP+DSW. Both versions of 3MRP clearly outperform both GPSR and VIRTUS [12] in more than 4 dB, 1 db, respectively. We can see that the case including the five metrics and the dynamic metric weigh distribution (3MRP+DSW) improves the PSNR in 2 dB compared to the case of using equal weights in 3MRP. This is because 3MRP+DSW selects the best forwarding node based on special characteristics for VANETs and also it uses a dynamic metric weigh distribution that classifies nodes in a better way giving each metric its importance depending on the current environment conditions.

A. Gain for I, P and B Video Frames

To better see separately the benefits of our proposal obtained for I, P and B video frames, we define the following parameters:

$$\%GainI = \left(\frac{IL_{GPSR} - IL_{3MRP+DSW}}{IL_{GPSR}} \right) \cdot 100 \quad (24)$$

$$\%GainP = \left(\frac{PL_{GPSR} - PL_{3MRP+DSW}}{PL_{GPSR}} \right) \cdot 100 \quad (25)$$

$$\%GainB = \left(\frac{BL_{GPSR} - BL_{3MRP+DSW}}{BL_{GPSR}} \right) \cdot 100 \quad (26)$$

where:

- 1) IL_{GPSR} : percentage of packet losses for I frames when GPSR is used.
- 2) $IL_{3MRP+DSW}$: percentage of packet losses for I frames when 3MRP+DSW is used.
- 3) PL_{GPSR} : percentage of packet losses for P frames when GPSR is used.
- 4) $PL_{3MRP+DSW}$: percentage of packet losses for P frames when 3MRP+DSW is used.
- 5) BL_{GPSR} : percentage of packet losses for B frames when GPSR is used.

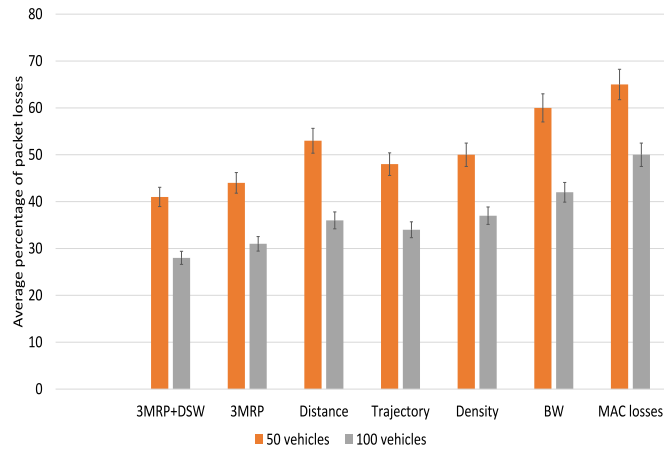


Fig. 18. Performance of 3MRP+DSW compared to 3MRP and different 3MRP options using single metric, in low (50 nodes) and medium (100 nodes) density scenarios.

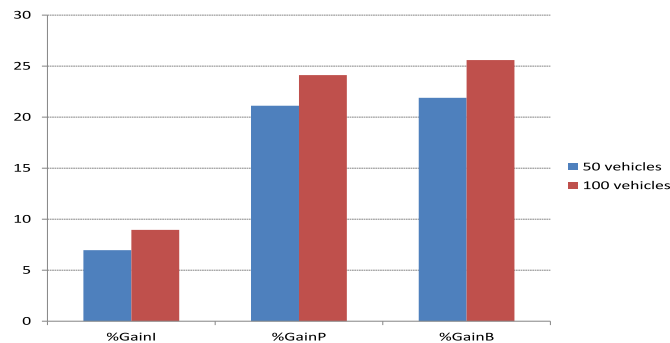


Fig. 19. Average percentage gain for I, P and B frames.

- 6) $BL_{3MRP+DSW}$: percentage of packet losses for B frames when 3MRP+DSW is used.
- 7) $GainI$: Gain obtained for I frames using 3MRP+DSW with respect to GPSR.
- 8) $GainP$: Gain obtained for P frames using 3MRP+DSW with respect to GPSR.
- 9) $GainB$: Gain obtained for B frames using 3MRP+DSW with respect to GPSR.

Fig. 18 shows the percentage of losses of our multimetric proposal with dynamic weights 3MRP+DSW compared to 3MRP and to several single-metric options of 3MRP. We can see in 3MRP that considering several metrics in the routing procedure shows remarkable benefits. In addition, 3MRP+DSW further improves the performance, since it includes dynamic weights.

Fig. 19 shows the average of all the simulation results for $\%GainI$, $\%GainP$ and $\%GainB$. We can see that using 3MRP+DSW, the gain is 7% (9%) for I packets, 21% (24%) for P packets and 22% (26%) for B packets, for 50 vehicles scenario (100 vehicles scenario) with respect to GPSR. As we can observe, $\%GainB > \%GainP > \%GainI$ in all the cases. The improvement is higher for the high density scenario, since there are more vehicles to choose the best candidates to forward packets according to our multimetric algorithm. Besides, the gain is much noticeable for P and B frames than for I frames.

We impute this fact to the higher size of I frames, which makes it more difficult to enhance the performance so notably as with smaller P and B frames.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have presented a new routing protocol named (3MRP) for VANETs to send video-reporting messages in urban scenarios. Our framework could be used in smart cities where prevention and management of accidents is an important goal. We understand that with a video message, the level of seriousness of the accident could be much better evaluated by the authorities (i.e., hospitals, ambulances) allowing a fast warning of the incident to emergency units, which potentially could save lives. Besides, the transport unit in charge of the traffic information services would quickly be warned by the vehicles immediately after the event of any incident (e.g., traffic jam or traffic accident). Furthermore, vehicles would instantaneously warn other vehicles about any accident in the roads. That would require the design of a proper dissemination protocol, which we will start in a future work. All these actions would improve the quality of life in the smart cities and even avoid accidents and save lives.

3MRP includes five metrics (distance, trajectory, density, available bandwidth and MAC layer losses) to take local forwarding decisions. Moreover, the proposal is building-aware, which avoids to select those nodes in transmission range located behind a building. This feature allows the network simulator to send packets only to nodes that are not behind buildings, thus mimicking what happens in reality which makes our simulations more realistic. The reason is that in real life buildings would block the signal and packets would be dropped if the receiver was behind a building. For that purpose, we have developed a program named REVSIM [2] that gives the state (i.e., in LOS or not) of any neighbor with respect to the current forwarding node in order to see if that neighbor can be a candidate as next forwarding node. Otherwise, this neighbor node will be deleted from the neighbor list, since it is behind a building.

In addition, a local buffer is used to temporarily store those packets when the routing protocol fails in finding a proper next forwarding node. A timer is activated and if the timer expires over 2 seconds, the packet is dropped since the video frame would reach destination too late for the decoding process.

The multimetric forwarding algorithm computes a global score value used to select the best next forwarding node among all the neighbors in LOS within the transmission range. In addition, we have developed an algorithm able to update the weights of the metrics dynamically (i.e., updating the multimetric score of neighbors) throughout time, so that most decisive metrics are highlighted. This helps the overall protocol to give each metric its importance at each moment and as a consequence we attain better results compared to giving fixed weights to all the metrics. We evaluated our proposal compared to GPSR and VIRTUS [12] in two scenarios with low and medium vehicles' density. In terms of packet losses and throughput, 3MRP+DSW improves 3MRP, VIRTUS [12] and GPSR, due to the new way of selecting the next forwarding node and the dynamic

scheme to give each metric its corresponding weight. We conclude that 3MRP+DSW performs better in both scenarios with low and medium vehicles' density. Our proposal makes the network more efficient as well as achieves a higher degree of satisfaction of the users by receiving much more frames with a good average end-to-end delay. This definitively will improve the quality of the video perceived by the end user as the PSNR value shows.

As future work, we plan to apply a game-theoretical scheme in our routing protocol 3MRP so that some frames would be forwarded with a certain probability through the best neighbor vehicle and the other frames through the second best neighbor vehicle. This scheme is inspired in our work [23] where we designed a game-theoretical forwarding scheme in a multipath routing protocol to send video-warning messages in mobile ad hoc networks (MANETs). Furthermore, we plan to design an adaptive video-streaming source able to send only I frames in case delay or losses locally grow up to a threshold.

REFERENCES

- [1] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO—Simulation of urban mobility," *Int. J. Adv. Syst. Meas.*, vol. 5, pp. 128–138, 2012.
- [2] A. Mezher, J. Oltra, L. Urquiza-Aguiar, C. Iza-Parades, C. Tripp-Barba, and M. A. Igartua, "Realistic environment for VANET simulations to detect the presence of obstacles in vehicular ad hoc networks," in *Proc. Perform. Eval. Wireless Ad Hoc, Sensor, Ubiquitous Netw.*, Nov. 2014, vol. 47, no. 11, pp. 77–84.
- [3] S. Olariu and M. Weigle, *Vehicular Networks: From Theory to Practice*. New York, NY, USA: Taylor & Francis, 2009.
- [4] H. Hartenstein and K. Laberteaux, *VANET Vehicular Applications and Inter-Networking Technologies*. Hoboken, NJ, USA: Wiley, 2009.
- [5] "Open street maps," 2006. [Online]. Available: <http://www.openstreetmap.org/>
- [6] K. Lee, U. Lee, and M. Gerla, "Survey of routing protocols in vehicular ad hoc networks," in *Advances in Vehicular Ad-Hoc Networks: Developments and Challenges*. Hershey, PA, USA: IGI Global, 2009, pp. 149–170.
- [7] H. Y. A. Wahid and D. Kim, "Unicast geographic routing protocols for inter-vehicle communications: A survey," in *Proc. 5th ACM Workshop Perform. Monit. Meas. Heterogeneous Wireless Wired Netw.*, Oct. 2010, pp. 17–24.
- [8] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. 6th Int. Conf. Mobile Comput. Netw.*, 2000, pp. 243–254.
- [9] H. Menouar, M. Lenardi, and F. Filali, "Movement prediction-based routing (MOPR) concept for position-based routing in vehicular networks," in *Proc. IEEE Veh. Technol. Conf.*, 2007, pp. 2101–2105.
- [10] D. Xiao, L. Peng, C. O. Asogwa, and L. Huang, "An improved GPSR routing protocol," *Int. J. Adv. Comput. Technol.*, vol. 3, no. 5, pp. 132–139, Jun. 2011.
- [11] C. Tripp-Barba *et al.*, "A multimetric, map-aware routing protocol for VANETs in urban areas," *Sensors Netw.*, vol. 14, pp. 2199–2224, 2014.
- [12] C. Rezende, A. Boukerche, H. Ramos, and A. Loureiro, "A reactive and scalable unicast solution for video streaming over VANETs," *IEEE Trans. Comput.*, vol. 64, no. 3, pp. 614–626, Mar. 2015.
- [13] H. Xie, A. Boukerche, and A. A. F. Loureiro, "A multipath video streaming solution for vehicular networks with link disjoint and node-disjoint," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 12, pp. 3223–3235, Dec. 2015.
- [14] C. Tripp-Barba, L. Urquiza-Aguiar, M. A. Igartua, A. M. Mezher, A. Zaldivar-Colado, and I. Guérin-Lassous, "Available bandwidth estimation in GPSR for VANETs," in *Proc. 3rd ACM Symp. Des. Anal. Intell. Veh. Netw. Appl.*, Nov. 2013, pp. 1–8.
- [15] "The network simulator, NS-2," 1995. [Online]. Available: <http://nsnam.isi.edu/nsnam/>
- [16] M. Fougue, P. Garrido, F. Martínez, J. Cano, C. Calafate, and P. Manzoni, "A realistic simulation framework for vehicular networks," in *Proc. 5th Int. ICST Conf. Simul. Tools Techn. Simutools*, Mar. 2012, pp. 37–46.
- [17] J. J. Oltra, "Design and implementation of an algorithm to analyse the presence of buildings in vehicular ad hoc networks," M.S. thesis, Ingeniería de Telecomunicaciones, Universitat Politècnica de Catalunya, Barcelona, Spain, Jul. 2014. [Online]. Available: <http://hdl.handle.net/2099.1/22391>
- [18] "REVsims, realistic environment for VANETs simulation tool," 2014. [Online]. Available: <https://drive.google.com/open?id=0B81r-ZO1x4wnbUJLa3JZRWdDVDg>
- [19] C. Quadros, A. Santos, M. Gerla, and E. Cerqueira, "QoE-driven dissemination of real-time videos over vehicular networks," *Comput. Commun.*, vol. 9192, pp. 133–147, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0140366416302651>
- [20] C. Sarr, C. Chaudet, G. Chelius, and I. Lassous, "Bandwidth estimation for IEEE 802.11-based ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 10, pp. 1228–1241, Oct. 2008.
- [21] *IEEE 802.11 standard: Part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications.*, 2003. [Online]. Available: <http://standards.ieee.org/getieee802/>
- [22] "Traffic accidents," 2015. [Online]. Available: <http://www.youtube.com/watch?v=jmZ4iRRTrzs>
- [23] A. M. Mezher *et al.*, "A multi-user game-theoretical multipath routing protocol to send video-warning messages over mobile ad hoc networks," *Sensors*, vol. 15, no. 4, pp. 9039–9077, 2015. [Online]. Available: <http://www.mdpi.com/1424-8220/15/4/9039>



Ahmad Mohamad Mezher received the M.Sc. degree in signals and systems from the Central University of Las Villas, Santa Clara, Cuba, and the Ph.D. degree in network engineering from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. He is currently a Postdoctoral Researcher with the Information Security Group, Department of Network Engineering, UPC. His research interests include data privacy, machine learning, and vehicular ad hoc networks.



Mónica Aguilar Igartua received the M.Sc. and Ph.D. degrees in telecommunication engineering from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 1995 and 2000, respectively.

Her research interests include design, analytical modeling, design of dynamic self-configured routing protocols, and performance evaluation of multimedia services over vehicular ad hoc networks (VANETs). Her current work deals with the design of privacy-aware VANETs, electric vehicle, automotive vehicle, and integration of mobility models for VANETs and

smart cities.