IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 5, 2018, accepted November 22, 2018, date of publication December 24, 2018,
date of current version January 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2889604

3D Simulation Modeling of UAV-to-Car Communications

SEILENDRIA A. HADIWARDOYO[1], (Graduate Student Member, IEEE), CARLOS T. CALAFATE[1], JUAN-CARLOS CANO[1], YUSHENG JI[2], (Senior Member, IEEE), ENRIQUE HERNÁNDEZ-ORALLO[1], (Member, IEEE), AND PIETRO MANZONI[1], (Senior Member, IEEE)

[1] Department of Computer Engineering, Universitat Politècnica de València, 46022 Valencia, Spain
[2] Information Systems Architecture Research Division, National Institute of Informatics, Tokyo 100-0003, Japan

Corresponding author: Seilendria A. Hadiwardoyo (seiha@upv.es)

This work was supported in part by the Japan Society for the Promotion of Science KAKENHI under Grant JP16H02817 and Grant JP18KK0279, in part by the International Internship Program of the National Institute of Informatics, Japan, and in part by the Ministerio de Economía y Competividad, Programa Estatal de Investigación, Desarollo e Innovación Orientada a los Retos de la Sociedad, Proyectos I+D+I 2014, Government of Spain, under Grant TEC2014-52690-R and Grant BES-2015-075988.

ABSTRACT In this paper, we propose a realistic model for simulating communications between unmanned aerial vehicles (UAVs), or drones, and ground vehicles, which can support mobile infrastructure to broadcast alerts in emergency situations. Three-dimensional positioning features should be considered in these simulations that involve UAVs and ground vehicles since communications links are not based on a flat surface. In fact, irregular terrains in the form of hills and mountains can greatly affect the communications by acting as obstacles that block radio signals partially or totally. Hence, in this paper, we propose a simulation model that conforms to this kind of communication and that was developed in the scope of the OMNeT++ simulator. The simulation results achieved showed a great degree of similarities with those results obtained in a real testbed for different scenarios. In addition, various path loss models and elevation models were considered to improve the level of realism of the simulation model.

INDEX TERMS Intelligent transportation systems, vehicular and wireless technologies, unmanned aerial vehicles, simulation, channel models, digital elevation models.

I. INTRODUCTION

In emergency situations such as search and rescue or disaster scenarios, UAVs (Unmanned Aerial Vehicles), also known as drones, can act as supporting nodes for communications since they can benefit from a wider communications range, and can be deployed flexibly, on demand. UAVs have better line-of-sight (LOS) features compared to ground infrastructures. In addition, UAVs can relay information and can form a Flying Ad-hoc Network (FANET) [1]. Intelligent Transport Systems (ITS) applications, such as remote sensing [2] and disaster assistance operations [3], can use UAVs to optimize their deployment. Moreover, UAVs can enhance the communications between ground vehicles in scope of ITS applications [4]. UAVs can even be deployed to assist vehicular networks as store-carry-forward nodes [5].

Differently from vehicles that move on the ground, following well-known established routes, UAVs can move freely in a three-dimensional space. Hence, the mobility is not restricted to road layouts or to a two-dimensional space, whether moving randomly as a single UAV or as a UAV swarm [6]. In situations where the communications rely on both UAVs and ground vehicles, their performance can be affected by diffraction from mountains, or by blockage by hilly terrains that will hence experience signal attenuation [7]. For more details on the behavior of the communications channel between ground nodes and UAVs, please refer to [8] and [9].

To assess the quality of communications that involve UAVs and ground vehicles, simulations are performed for scaled analysis. However, most simulation frameworks only assume two-dimensional environment [10]. The real world scenario that involves vehicles are projected into a planar network. Hence, it will not be possible to consider a flying node or a UAV, which take into account its altitude when communicating with ground nodes. In addition, not all traffic environments are perfectly flat as some streets might encounter ascent or descent. Neglecting these details of vehicular environments can lead to different results of analysis of vehicular communications. Hence, elevation data containing altitude information should be considered if we want to analyze three-dimensional vehicular communications [11].

In this paper, we propose and implement a novel simulation model for UAV-to-car communications. The model takes into account the results obtained from a real testbed [12], and it was developed for the OMNeT++ simulation tool [13]. The simulation model takes into account three-dimensional communications. A comparison with two-dimensional communications is also presented in this paper. This model calculates the signal attenuation due to the presence of hilly or mountainous terrains that hinder the communications by retrieving elevation information of the Earth provided by a Digital Elevation Model (DEM) [14]. Signal attenuation is obtained from existing propagation models. The use of various DEMs and propagation models (i.e., diffraction models like Bullington and Deygout) are compared to the results obtained from our testbed experiments. The results from the simulation tests show that the model obtains results comparable to testbed experiments, thus validating their adequateness.

The organization of the paper is as follows: in the next section we provide an overview of related works. A theoretical concept to build the simulation model, as well as the proposed architecture, is explained in section 3. Details on how the simulation test is performed is detailed in section 4. Section 5 presents the results from our simulation model. Then, in section 6, we present a comparison of the simulation tests with various DEMs. In section 7, another comparison is presented with the experiments using various propagation models. Finally, we conclude the paper and discuss future works in section 8.

II. RELATED WORKS

Even though UAVs or drones can be integrated with various wireless communications, only quite recently have researchers proposed integrating networks combining UAVs with VANETs. In particular, only a few papers actually examine the communications performance between UAVs and cars through real testbed experiments. Notice that, in general, such real experiments involve a static ground node with a UAV that is moving, not the other way around [15], [16].

On the other hand, several works address the simulation of UAV-to-Car communications. Jia and Zhang [17] have conducted extensive simulations to verify the existence of an optimal UAV altitude, and a minimum number of UAVs to guarantee a target connectivity among vehicles. More efforts include creating an inter-connectivity for a disconnected group of cars using UAVs as relays [18], or analyzing the vehicle-to-drone packet delivery delay in VANETs [19]. However, all these researchers performed their experiments using a flat scenario, thus neglecting 3D communication effects.

For the sake of providing a three-dimensional communications environment model, several efforts have considered terrain or environmental settings in the scope of simulation. In particular, terrain surfaces are considered as obstacles when estimating signal power in the simulation. Filiposka *et al.* [20] include the implementation of a DEM to indicate obstacles formed by terrains in the ns-2 simulator. A DEM is used to define the signal attenuation due to the actual terrain profile. A similar approach is implemented in [21] where, using a DEM as a reference, the authors emulate the signal propagation in OMNeT++ [13]. However, these two papers emphasized using mobile nodes that move randomly on the ground. Brummer *et al.* [10] describe the work proposing a simulation framework for Vehicle-to-Vehicle (V2V) communications when considering the three-dimensional space. The simulation framework proposed uses a DEM to support the information of the vehicle's location in 3D space, as well as to identify obstacles formed by terrains. Nevertheless, the proposed method was only suitable for Vehicle-to-Vehicle (V2V) scenarios in urban settings and did not consider nodes that are not following roads (i.e., infrastructure-related restrictions).

Contrary to previous research efforts, our work emphasizes on bringing the real experiment settings into simulation settings. We achieve this by proposing a model to characterize 3D communications between UAVs and cars. Our final goal is to improve the level of realism of our experiments by extending the simulation environment. Specifically, we propose a novel module for the simulation framework that resembles realistic measurements to characterize three-dimensional communications. This 3D communications model involves UAVs in the air, and cars on the ground. Hence, it can be used to seamlessly integrate FANETs with VANETs. To undertake this type of measurement, 3D map information had to be imported into the simulator to get geographical information. Afterwards, path loss was defined based on the geographical information that provides elevation data. By using the sample parameters used in our previous real testbeds [12], [22] (see Fig. 1), our goal is to validate the model by obtaining comparable results to achieve an improved level of realism. Hence, not only the technical settings are taken into account in the simulations, but we also put our efforts on having the

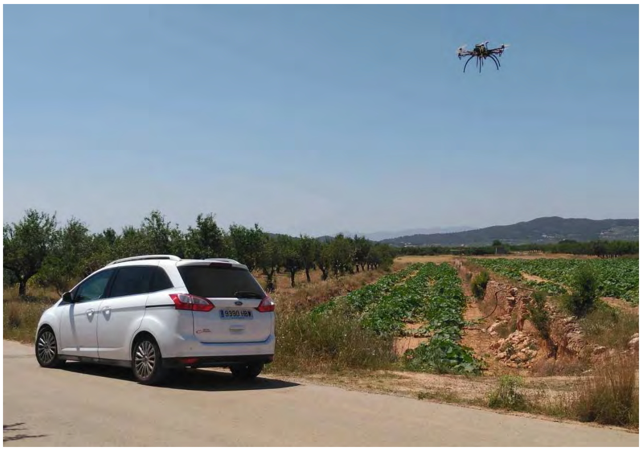


FIGURE 1. UAV-to-Car Communications Testbed Experiment [22].

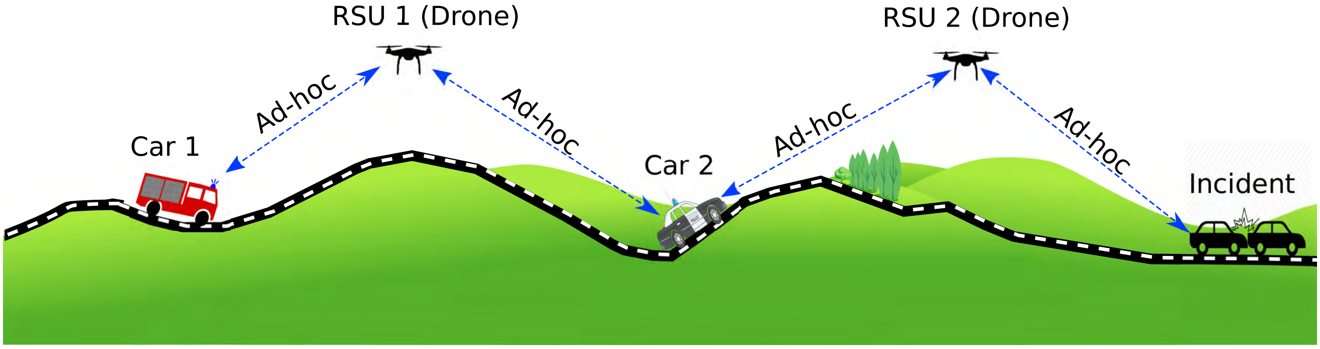


FIGURE 2. Drones as Mobile RSUs [22].

same geographical settings in the simulator as in the real testbed settings. Using OMNeT++ [13], SUMO [23], and Veins [24], we can replicate the testbed experiments in a realistic simulation framework with appropriate parameter settings, thus validating both the testbed and the simulation experiments.

III. 3D COMMUNICATIONS MODEL FOR UAV-TO-CAR

The main goal of our research work is to support the use of UAV as relays for vehicular communications. The UAV can support the communications between cars when the area of the network has limited infrastructure support. As seen in Fig. 2, UAVs are useful to detect incidents in rural areas. In this paper, we develop a simulation framework where the communications between nodes are affected by terrain irregularities that might act as obstacles. Hence, the actual 3D terrain profile is of utmost importance since it can impact the network connectivity. For UAV-to-car communications, the challenges are not the visibility range nor the interference, especially in rural areas, where infrastructures are limited and the interference was minimal. The terrain profile is the main challenge, especially when the presence of hills or mountains represents obstacles that hinder wireless communications. The detection of obstacles is thus necessary to characterize the communication when obstructed by the terrain features. This is done by extracting the elevation information of the terrain. An obstruction to communications is defined by a higher elevation. This obstruction from the elevation can cause moderate to high attenuation of radio signals.

A. ELEVATION MODEL

In order to include terrains as obstacles, such as hills or mountains, information about the elevation of the terrain is required. A Digital Elevation Model (DEM) provides information about the real-world terrain data, including its elevation. The DEM, which contains data about the Earth's surface, can be used to recognize how high the terrain is above the sea level. Based on that information, the terrain can be defined either as flat, hilly, or mountainous. The DEM also provides information about the elevation of any given points, and it can be obtained from different sources. Specifically, we used two different DEMs: the NASA's DEM, which is collected from the Shuttle Radar Topography Mission (SRTM) and the Google's DEM, accessed via the Google Maps Elevation API.

The NASA's DEM is collected from the Shuttle Radar Topography Mission (SRTM) [25], which obtained such information from capturing the Earth's surface with a radar sensor from space. The model has various resolutions that define the accuracy of the elevation information. The NASA's DEM files are in "height" format and contain global elevation data captured during NASA space missions, and are named in format of *northing* and *easting*, which are based on their coverage in geocoordinates. *Northing* means the northward measured distance when adopting the UTM system, while *easting* means the eastward-measured distance. The file then covers one degree of latitude and one degree of longitude. To include DEM into our simulation environment, a suitable file based on the location of the scenario should be selected. This DEM contains an array of points in the form of a grid and has two resolutions, one is 1-arc second, and the other one is a 3-arc second sampling. The right elevation is obtained by pinpointing the appropriate latitude and longitude into the right cell of the grid.

If we use a DEM corresponding to a 3-arc second sampling in order to obtain elevation data, we will get a grid of 1201 × 1201 cells, as illustrated in Fig. 3, in which

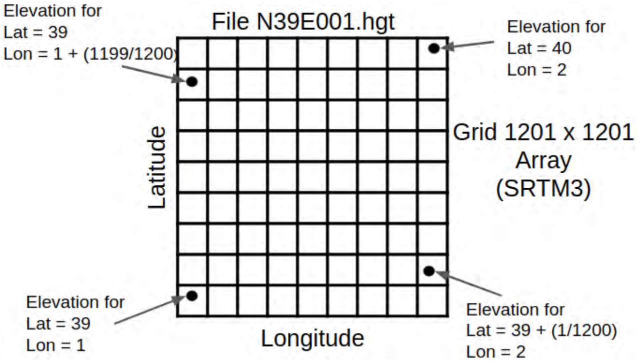


FIGURE 3. Shuttle Radar Topography Mission (SRTM) Digital Elevation Model (DEM) Data Structure.

1201 cells represent one degree of latitude or longitude. To define the elevation or the height of the terrain, we must index the target coordinate based on its northing and easting information. To pinpoint the exact cell of the grid, we can use the fraction part of the geographical coordinate and multiplying it by 1200, in the case of 3-arc second sampling. As an example, 0.5 belongs to the 600th cell.

The Google's DEM can be obtained by an API that allows us to query locations around the earth for elevation data. The Google Maps Elevation API [26] is intended for developing hiking and biking applications, positioning applications, or low-resolution surveying applications. Using this API, we can retrieve the altitude information of practically any points data on Google Earth. To retrieve the altitude of the terrain, the API requires the coordinates in longitude and latitude. As a result, it will provide the altitude in meters. The good thing about using this API is that it can access the larger database that Google has, which might have better precision as it is daily updated by both developers and users. In addition, it is easier to retrieve the elevation information, as we just need to input the coordinates in latitude and longitude and get an instant altitude of the terrains in return. The drawbacks of using this API is the exclusivity of accessing the data. The API has a limited free access, which is only up to 10 requests per second, for a maximum of 2500 requests per day. In addition, there is a maximum of 512 locations that can be retrieved per request. This means that the service is not accessible to the general public who typically have limited resources.

B. PROPAGATION MODELS

In this subsection we describe the three different propagation models used in the evaluation and how they can be used with the DEM information obtained for a particular area.

High terrains can affect the quality of wireless communications since they can represent signal obstructions. The presence of hilly or mountainous terrains can cause non-line-of-sight conditions. This means that the power of the received signals is reduced or even blocked, and so these terrains can act as signal obstructions. In some cases, diffraction effects can also occur due to high terrains. This means waves will bend or deflect when encountering hilly terrains or other types of obstructions. We can analyze the diffraction effect by approximating obstacles as knife edges. In such case, the signal transmitted in the network will be attenuated due to the knife edge diffraction effect. Knife edge diffraction loss is calculated by considering the distance between the transmitter and the receiver, the wavelength of the signal, and the height (h) of the obstacle above the line of sight line formed between the transmitter and the receiver. By retrieving the elevation information, we can obtain this height.

Calculation with a parametric equation is required in order to get the LOS line between the locations of the transmitter and the receiver. Then, detecting the terrain that acts as an obstacle and blocks the signal is done by moving along the LOS line in discrete steps. We will be able to calculate the height of the obstacle (h) or the knife at each step by getting a positive value. Otherwise, getting a negative value means that the terrain is not an obstacle to the signal.

In more detail, defining a LOS line is done by a parametric equation based on the location of the UAV, which includes its latitude (x_s), longitude (y_s), and altitude (z_s), and the vehicle position (x_r, y_r, z_r). In order to compare the detected LOS line and the terrains present in the network, we have to define an inspection point (x_p, y_p, z_p). We can obtain the inspection point and analyze it depending on how many slices the LOS line is divided into. Notice that the LOS line is divided into slices by inspection points, as shown in Fig. 4. The fraction of the LOS can be defined by specifying these slices. This fraction will be the s parameter in the following parametric equation:

$$\begin{aligned} x_p &= x_s + s \cdot (x_r - x_s) \\ y_p &= y_s + s \cdot (y_r - y_s) \\ z_p &= z_s + s \cdot (z_r - z_s) \end{aligned} \tag{1}$$

that is defined by the spacing of the resolution of the map. Slicing the LOS or spacing it should not be smaller than the resolution of the DEM file, since it can lead to unnecessary computing efforts. However, a higher spacing can lead to losing the information about potential obstacles that are present in the LOS.

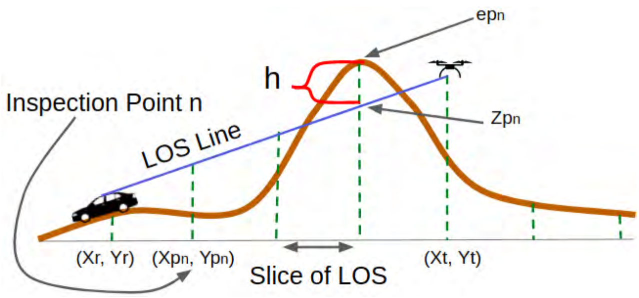


FIGURE 4. Set of slices that divide the LOS into inspection points.

The s parameter is needed to calculate the inspection point. Then, we can compare the z_p position for every inspection point. This point acts as the altitude of the LOS point. The elevation point or the terrain height (e_p) is obtained by consulting the DEM file to extract the elevation information by getting the x_p and y_p coordinates of the inspection point.

The knife or the height of the obstacle is obtained by calculating the difference between e_p and z_p, and it is denoted as h_p. When an elevation point at the inspection point is higher than the height of the LOS, or its altitude, an obstacle is detected at that particular location. The outcome of the calculation would be a positive h_p value. On the other hand, when the value of h_p is negative, LOS is not obstructed by an obstacle

Now, we describe the three propagation models used in the experiments: the first one (the *Single Knife Edge*) only considers a single edge and is based on the Fresnel-Kirchoff diffraction model; the other two consider several knife edges, assuming that there could be more than one obstacle present in the signal path. Among the most popular and widely used multiple knife edge diffraction models, we will analyze the *Bullington* [27] and the *Deygout* [28] diffraction models. The main reason behind choosing these models is that these two models were suggested in recent ITU-R recommendations when studying signal propagation effects in the presence of diffraction [29], [30]. For a general comparison between these three methods, table 1 summarizes their main features.

TABLE 1. Comparison of diffraction methods.

Features	Single Knife Edge	Deygout	Bullington
Number of Edges	Single	Multiple	Multiple
Accuracy	Low	Improved	Low
Complexity	Simpler	More Complex	Simpler
Main Edge	Edge with Maximum Height	Principal edge with largest v parameter	Equivalent knife edge between transmitter and receiver

1) SINGLE KNIFE EDGE

In the first model, a Fresnel- Kirchhoff diffraction parameter (v) is required after getting the height of the obstacle or the knife at a specific inspection point with the purpose of calculating the signal attenuation. It is obtained by taking into account the height of the obstacle (h_p), the distance between the obstacle and the sender (d_s), the distance between the obstacle and the receiver (d_r), and the wavelength (λ). Thus, the calculation of the diffraction parameter is done using the following equation:

$$v = h_p \sqrt{\frac{2(d_s + d_r)}{\lambda d_s d_r}} \tag{2}$$

We can then define the signal attenuation due to diffraction from the obstacle by obtaining the v parameter and using the Fresnel Integral $F(v)$. According to Lee [31], the Fresnel integral is determined as follows:

$$\begin{aligned}
F(v) &= 0 v \leq -1 \\
F(v) &= 0.5 - 0.62v - 1 \leq v \leq 0 \\
F(v) &= 0.5e^{-0.95v} 0 \leq v \leq 1 \\
F(v) &= 0.4 - \sqrt{0.1184 - (0.38 - 0.1v)^2} 1 \leq v \leq 2.4 \\
F(v) &= \frac{0.225}{v} v > 2.4
\end{aligned} \tag{3}$$

Once the Fresnel integral result is obtained, we can define the signal attenuation due to the diffraction using the following equation (in dBm):

$$L(dB) = 20 log\,|F(v)| \tag{4}$$

2) DEYGOUT MODEL

The Deygout model differs from the single knife edge diffraction model as it not only considers the point with the highest value of the knife along the path, but it also considers ''secondary'' obstacles that are added to the diffraction loss over the main obstacle. This is a cascaded knife edge method. The diffraction parameter calculation is the same as the previous method (single knife edge) using equation 2. The knife edge with the highest value must be found. However, to get the height of the obstacle, or h_p, the calculation was not done using the previous parametric equation. Instead, it uses the equation shown below:

$$h_p = e_p + [d_s d_r / 2r_e] - [(z_s d_r + z_r d_s)/d] \tag{5}$$

where we have to include the Earth curvature in this method; r_e corresponds to the effective Earth radius, which is about 6371 kilometers and d is the distance between the transmitter and the receiver.

The value of v for every point must be calculated referencing every knife edge. The maximum value of v indicates the principal knife edge, which is the highest obstacle. If v results in a value of more than -0.78, it means that there is no loss in the signal path; otherwise 0 dB is returned. However, if the v value is less than -0.78, then we have to keep calculating the value of v, but this time the points are not between the transmitter and the receiver, but between the transmitter and the principal knife edge, which will return the v_s value. The same process is then repeated for the points between the principal knife edge and the receiver. This time we will get the v_r value.

Once we get the knife height of the principal edge (v), the edge between the principal edge and the transmitter (v_s), and the edge between the principal edge and the receiver (v_r), we can calculate the attenuation using this method. The signal loss, in dB, is denoted as:

$$L(dB) = F(v) + T[F(v_s) + F(v_r) + C] \tag{6}$$

Calculating the Fresnel integral for Deygout is not the same as in the single knife edge diffraction model. The Fresnel integral calculation used in this method is as follows:

$$F(v) = 6.9 + 20log(\sqrt{(v - 0.1)^2 + 1} + v - 0.1) \tag{7}$$

The C value in equation 6 is an empirical correction given by the following equation:

$$C = 10.0 + 0.04D \tag{8}$$

where D is the total path length in kilometers, whereas T is a factor denoted as:

$$T = 1.0 - exp\left(-\frac{F(v)}{6.0}\right) \tag{9}$$

This way, the $L(dB)$ value obtained is the overall attenuation value for the method of using the Deygout diffraction model.

3) BULLINGTON MODEL

As recommended by ITU for propagation by diffraction [30], the Bullington model is used as a method for a generic terrestrial path. The Bullington diffraction method is used to calculate the attenuation whether the signal path is LOS or is trans-horizon. In this method, the height of the knife is defined by the slope of the line between the transmitter, the inspection point, and the receiver.

In the Bullington method, the first step to do is to get the intermediate profile point with the highest slope of the line from the transmitter to that point. The slope is obtained using equation (S_{tim}):

$$S_{tim} = max\left[\frac{e_p + 500C_e di(d - d_i) - z_s}{d_i}\right] \tag{10}$$

where C_e is the effective Earth curvature in km^{-1}, which is given by $1/r_e$. d_i is the distance between the transmitter and the inspection point i (or the point which divides the LOS into slices). After getting the slope of the line between the transmitter to the inspection point, we need to calculate the slope of the line from the transmitter to the receiver assuming a LOS path:

$$S_{tr} = \frac{z_r - z_s}{d} \tag{11}$$

After getting the two slopes, there are two cases to be considered. The first case considers that the path has LOS so the slope of the line from the transmitter to the point is less than the slope of the line from the transmitter to the receiver ($S_{tim} < S_{tr}$). In that situation, we need to find the inspection point that has the highest diffraction parameter v, obtained by:

$$v_{max} = max\left\{\left[e_p + 500C_e di(d - d_i) - \frac{z_s(d - d_i) + z_r d_i}{d}\right] \times \sqrt{\frac{0.002d}{\lambda di(d - d_i)}}\right\} \tag{12}$$

In this first case, the knife edge loss for the Bullington method is obtained by:

$$L(dB) = F(v_{max}) \tag{13}$$

using equation 7, if the v_{max} is greater than -0.78. Otherwise, if it is lower, the value returns 0 dB.

For the second case, the path is trans-horizon, which means that the slope of the line between the transmitter and the inspection point is equal or greater than the slope of the line from the transmitter and the receiver ($S_{tim} \geq S_{tr}$). Hence, for this case, we need to find the slope of the line from the receiver to the inspection point using the following equation:

$$S_{rim} = max\left[\frac{e_p + 500C_e di(d - d_i) - z_r}{d_i}\right] \tag{14}$$

Once we have obtained the slope of the line, we have to calculate the distance of the Bullington point from the transmitter using the following equation:

$$d_b = \frac{z_r - z_r + S_{rim}d}{S_{tim} + S_{rim}} \tag{15}$$

The Bullington point is required to get the Bullington diffraction parameter, denoted as v_b, which is obtained through the following equation:

$$v_b = \left\{\left[z_s + S_{tim}d_b - \frac{z_s(d - d_b) + z_r d_b}{d}\right]\sqrt{\frac{0.002d}{\lambda d_b(d - d_b)}}\right\} \tag{16}$$

Then, the Bullington diffraction parameter v_b is used to calculate the knife edge loss for the Bullington point as:

$$L(dB) = F(v_b) \tag{17}$$

using equation 7 to calculate the Fresnel integral. In both cases, whether the signal path is LOS or trans-horizon, we should calculate the final diffraction loss for the Bullington method (L_b) using the following equation:

$$L_b = L + [1 - exp(-L/6)](10 + 0.02d) \tag{18}$$

where L_b is the overall attenuation value obtained when applying the method using the Bullington diffraction model. The comparison of using these various diffraction models will be presented and evaluated in Section VII.

C. IMPLEMENTATION OF THE PROPOSED MODEL IN SIMULATION

This subsection is devoted to describing the implementation of the proposed 3D communications model in a standard simulation tool. In particular, for our experiments, we have used OMNeT++ [13] for network simulation, SUMO [23] for vehicular traffic simulation, and the Veins [24] simulator framework to connect SUMO and OMNeT++ for specific vehicular communications simulation. In particular, the effect of 3D wireless transmission has been implemented by modifying its physical layer. For our simulation experiments, we use WAVE as the MAC protocol, and the control channel is used for communications.

Our proposed model works as an extension to this simulation framework (combining OMNeT++, SUMO, and Veins) by adding a module that considers 3D space features. Traditional modules for signal propagation only consider 2D space. By modifying the physical layer in the simulation, we can simulate the effect of 3D wireless transmission. When configuring the propagation model in the simulation, we have to take into account the location of the nodes (latitude, longitude, and altitude), and the terrain spacing. In addition, other parameters that characterize the behavior of the signals, like the wavelength and the received power threshold, are configured by default in Veins. All these features are implemented in Veins, as shown in Fig. 5.

Inside the simulation tool, we have to extract the elevation information first. By considering the location of the nodes with respect to its northing and easting, the DEM data are imported. By consulting the DEM file, we can retrieve the elevation information that is accessed inside OMNeT++. The DEM file is locally stored in order to reduce computation, and thus simulation time. Thanks to the assistance of TraCI

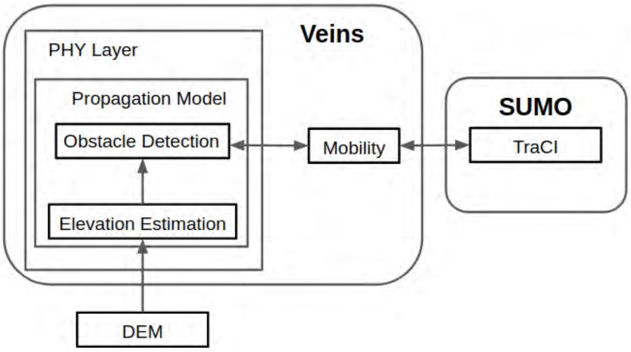


FIGURE 5. Scheme of the Simulation Architecture with the implemented modules.

in SUMO, we can obtain the exact geocoordinates (location) by connecting it to Veins, according to the mobility of the nodes.

The detection of obstacles is required after obtaining the coordinate of the vehicle from the mobility module. This functionality is executed inside the propagation module at the physical layer. The terrain profile with its elevation must be estimated in order to detect obstacles. The DEM file inside OMNeT++ is accessed in order to estimate the elevation of the terrain. By being able to detect obstacles using the knife-edge diffraction model, the propagation module can better determine the actual signal attenuation. On the receiver side, OMNeT++ can assess the signal strength, as well as determine the minimum signal strength to receive packets from this communication model.

As an alternative, if we want to use another elevation model for comparison purposes, we will include the Google Maps Elevation API; in this case, some libraries are required to be included in the simulator. Since OMNeT++ simulator is written in C++, libraries such as *libcurl* [32] and *jsoncpp* [33], both implemented in C++, provide access to the Google Maps Elevation API. The *libcurl* library is used to fetch data from remote websites, in our case the Google Maps Elevation API. In addition, since the Google Maps Elevation API would return the information in JSON format, the *jsoncpp* library is needed to decode and analyze the received information. It will parse the value of the elevation information that will be needed as information about the height of the terrain. Notice that terrain height information is the basic parameter required to assess the signal strength using the propagation module adopted.

IV. SIMULATION SETUP

In this section, we present an overview of the simulation architecture and the settings used to validate our model. Details about how to import the geographical location into the simulation, as well as using the 3D propagation module in the simulation tools for a specific scenario are provided in this section. In addition, data analysis covers how data are obtained to produce adequate results.

A. SIMULATION ARCHITECTURE

In order to validate our model, the simulation environment is modified to resemble the same environment used in our previous testbed experiments. A map from Open Street Map (OSM) is imported into SUMO. Since the experiment took place in the outskirts of Casinos, Spain, which is a rural area, the map corresponding to that particular location was imported into the SUMO. The location was far from the urban area, and so the interference in the 5 GHz band was minimal. In that particular location, free spaces like valleys or hilly terrains are present, as we can see in the aerial photo shown in Fig. 7, which evidences that the experiment's location represents a typical rural area. In particular, the road is running downhill from about 400 meters above the sea level (starting point) to about 320 meters above sea level (destination point). As presented in Fig. 6, we can trace a LOS point between the transmitter at the starting point to the receiver at the destination point. Depending on the altitude of the transmitter, the worst case line-of-sight conditions are expected when the vehicle is located about 900 meters away from the starting point, situation where signal obstructions exist. Such situation can cause communications to experience Non-Line-Of-Sight (NLOS) conditions.

The map of Casinos that was imported from OSM is used to generate the ground vehicle traffic using SUMO.

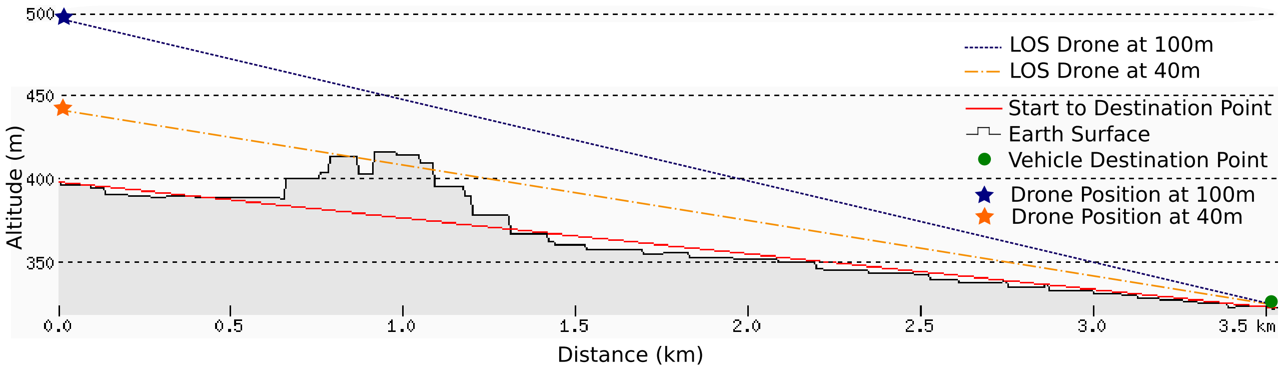


FIGURE 6. Elevation Profiles Measured from the Initial Starting Point to the Destination Point [12].

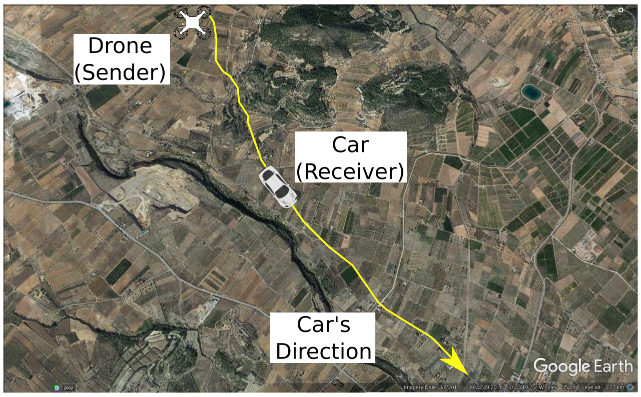


FIGURE 7. Satellite View of the target road near Casinos (Valencia, Spain) used in the real testbed experiments (see [22] for details).

The OSM files imported were modified so that the car used for testing is following the same trajectories as in the real experiments. Overall, the car's path is more than 3 kilometers long. As for the UAV, it moves freely in the air, meaning that mobility is independent of the map, and is defined directly in OMNeT++ without the use of SUMO.

In summary, a static UAV and a moving car are used in the scenario, thereby resembling the testbed experiments used as a reference.

B. SIMULATION SETTINGS

To simulate our experiments, we have used the map imported from Open Street Map (OSM) [34], which was then integrated into SUMO to manage vehicle's mobility. This way, to mimic the real mobility pattern, the car in the simulation can have the same trajectory and speed as in the real testbed. The Veins simulator is used to mimic real vehicular communications in the simulation.

The communications between UAV and car rely on the ad-hoc mode, and the UAV will act as the data source by generating UDP packets and broadcasting them into the network. The car will be moving away from the UAV that sends the packets. Both UAV and car record their own geographic location periodically.

The packets transmitted into the network, at a rate of 10 Hz, are Basic Safety Messages (BSMs) resembling the same scenario as in the testbed experiments. The parameters set in the simulation are specified in Table 2. The simulation was performed using a machine with Intel Core i7-4790 @ 3.60GHz x 8 and 8 GB RAM.

TABLE 2. Parameters used for simulation.

Parameter	Value
Transmit Power	200 mW
Antenna	5 dBi
Packet Size	1.4 kB
Message Type	BSM
Transmission Range	4.78 km
Packet Sending Rate	10 Hz

In this simulation, we include also our implemented module with the different models for path loss to provide 3D communications simulation. The module implements the diffraction of the terrain and can obtain information about the elevation of the terrain, which can then be used to calculate the signal attenuation. For comparison purposes, we will also perform an experiment that only considers 2D communications. In that case, the module that calculates the elevation as an obstruction will not be used. The comparison with these two methods will be presented in the following section.

The simulation is composed of a car moving back and forth, non-stop, following the trajectory for five consecutive times. During that period, the car will receive packets that are sent by the UAV. The packets that are sent are BSMs, which are transmitted periodically until the simulation stops. All the number of BSMs circulated in the network are recorded for statistical purposes.

While recording the flow of the packets, the simulation tool (OMNeT++) also records the location of the nodes in the network. The location that is obtained in the simulation is either in Cartesian coordinates or real geocoordinates. Such information will be useful to define the results, and for data analysis. The Cartesian coordinates correspond to the location of the node in OMNeT++ and SUMO (in x,y). The real geocoordinates correspond to the location of the node in the real world. Thanks to the TraCI module inside OMNet++, we can have the two types of coordinates seamlessly.

C. DATA ANALYSIS

From the simulation experiments, we can get the data to measure the packet delivery ratio based on distance. The simulation records the geographical location of the transmitter and the receiver whenever the receiver gets a packet. This way, the distance between the two nodes can be obtained directly by comparing the two node locations.

To calculate the packet delivery ratio, we compare the number of packets sent at the transmitter side, and the number of packets received at the receiver side within a predefined distance interval. Both endpoints are analyzed as in discrete simulation, and we can observe events occurring throughout time for every node. By considering the geographical information obtained from simulation, we are able to compare and calculate how many packets are sent and received.

In the simulation that includes the elevation profile, potential factors such as height can affect the communication's performance. Hence, we have chosen to vary the altitude of the transmitter (which in this case is the UAV). So, the LOS probability is affected by the UAV's altitude. Notice that, in general, a low altitude results in a higher probability of finding obstacles. We have considered two altitudes for the UAV: 100m and 40m. In this paper, we will differentiate these two scenarios that involve different UAV flight altitudes. For the scenario where the UAV is flying at 100m, we will name it as Scenario 1. The other scenario, that involves the UAV flying at 40m, is called Scenario 2.

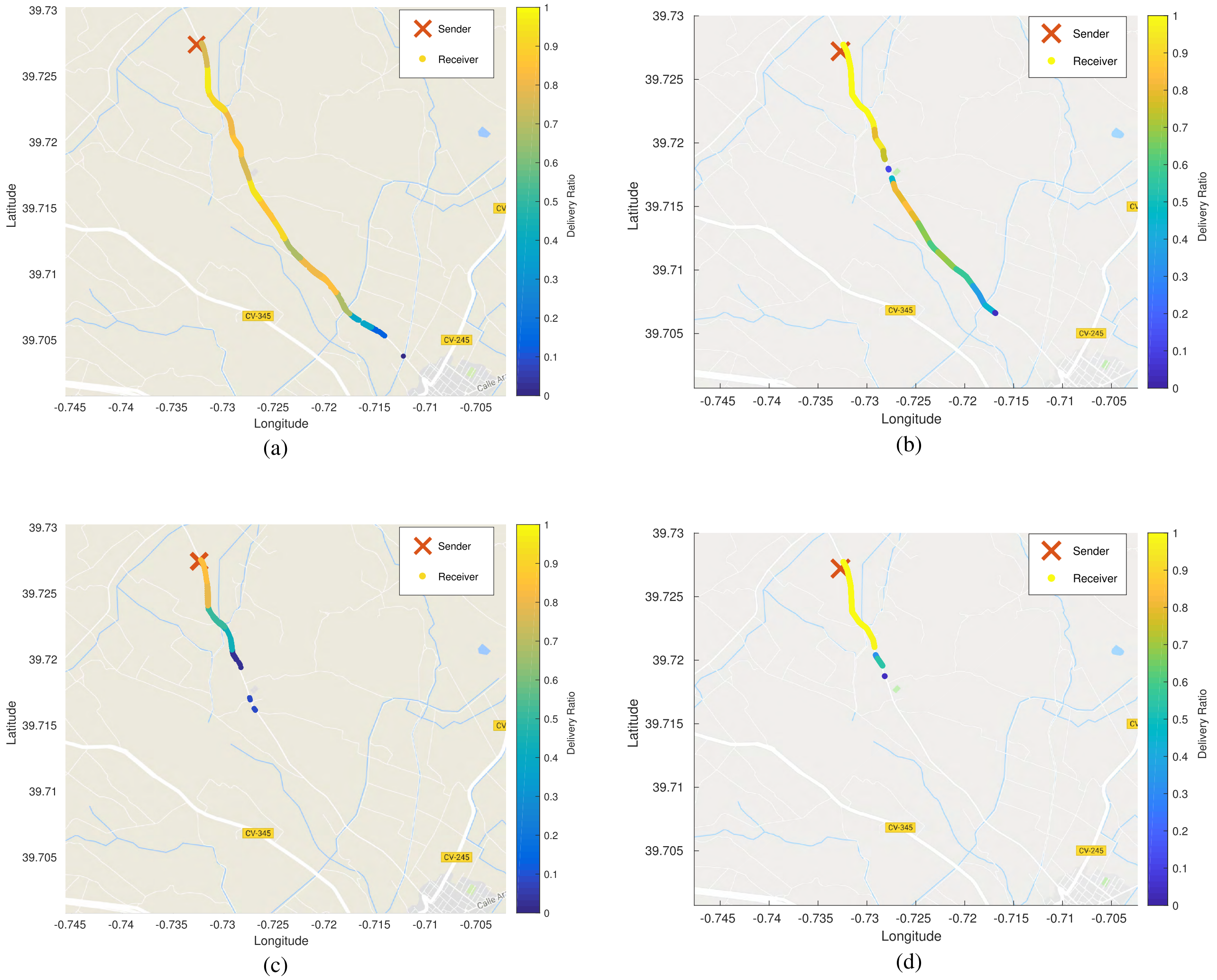


FIGURE 8. **Packet delivery ratio using heat maps from real experiments (left) and simulation scenarios (right). (a) Scenario 1. UAV Altitude: 100m, Real Experiments. (b) Scenario 1. UAV Altitude: 100m, Simulation. (c) Scenario 2. UAV Altitude: 40m, Real Experiments. (d) Scenario 2. UAV Altitude: 40m, Simulation.**

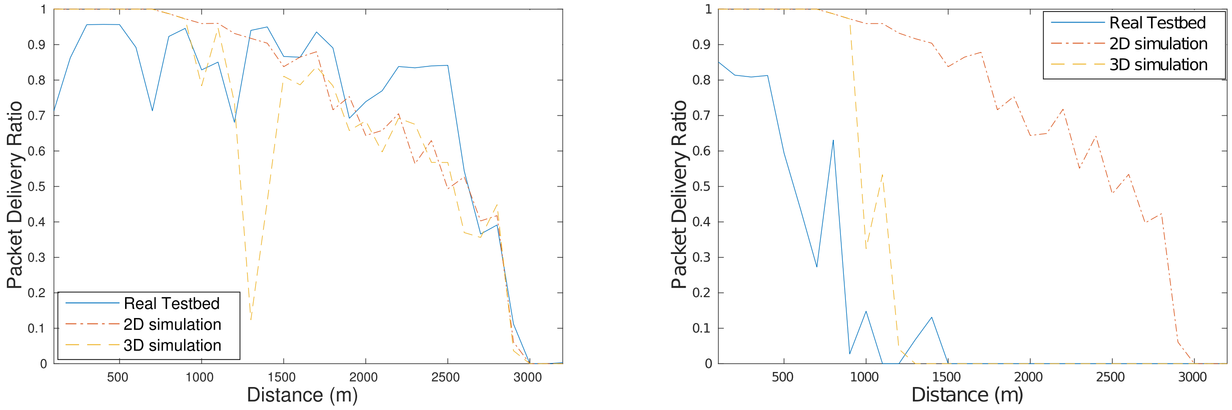
V. EVALUATION OF THE SIMULATION RESULTS

The goal of this section is to compare and validate the results obtained using the simulation architecture that includes our 3D communications model, with the results obtained from the real testbed experiments. In all the simulation experiment performed in this section we use the NASA's DEM data and the single-knife edge diffraction model. This diffraction model is used since our idea is to analyze the impact of including the 3D communications, hence the simplest diffraction model is used here. We will study the impact of using other diffraction models in Section VII.

Firstly, in Fig. 8 we can see the packet delivery ratio using heat maps for better visual description. The heat map is generated from the geographical information retrieved from the experiments. The points on the heat map indicate successful packet reception locations. Based on the packet delivery ratio, the color gradually changes from dark to light, denoting the ratio from low to high.

The location of the received packets in the heat map in Fig. 8 represent the exact location of the scenario for both testbed and simulation experiments. When comparing Fig. 8a with Fig. 8b, where the latter uses our 3D communications model, we can clearly observe that the reception event locations are quite comparable. In more detail, and for this scenario where the UAV flew at 100 meters, the heat map indicates that the points from the starting point to the destination point are associated to decaying packet delivery ratio values. In the real testbed, we can see on the heat map that some points are recorded for higher distances, despite representing very low values. In comparison, in the middle of the car's trajectory in the scenario, there is a drop in the delivery ratio in both simulation and testbed experiments, although this drop is more significant in the case of simulation.

The results when the UAV is flying lower (at an altitude of 40 meters) are also very similar. The points recorded on both simulation and real testbed show only half of its



(a)

(b)

FIGURE 9. Packet Delivery Ratio as a function of the distance between the car and the UAV. (a) Scenario 1. UAV Altitude of 100m. (b) Scenario 2. UAV Altitude of 40m.

trajectory, as seen in Fig. 8c and in Fig. 8d. The ratio recorded in the real testbed has a lower value for shorter distances. The color is darker at certain locations, as observed in the heat map. On the other hand, in the simulation the color is lighter, indicating a higher packet delivery ratio. However, both the real testbed and the simulation recorded a very low value at the end of the path, where the car can still receive packets. In that case, the LOS is starting to be obstructed at that particular location.

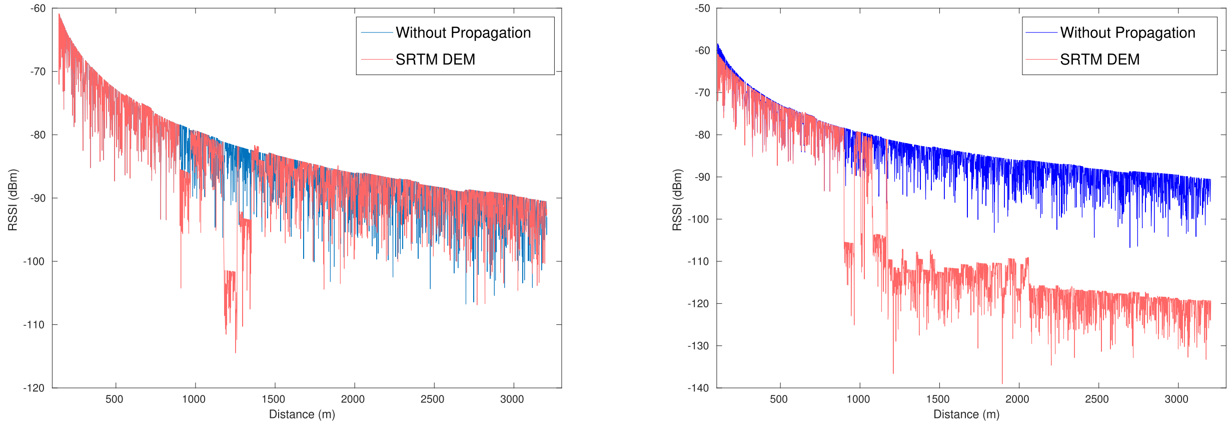
In order to have a more detailed look of the results, we represent the packet delivery ratio depending on the distance between the car and the UAV in Fig. 9. The figure shows results for both scenarios (when the drone is flying at 40m and 100m) using the same parameter settings. To have a better comparison and validate our model, we have also tested using the same parameters and the same scenario when only considering the 2D communications model (it is worth remembering that this model neglects the terrain elevation information). For each figure, the plots include the curves for the simulation taking into account the elevation information (our 3D simulation model), the simulation using only a 2D communications model (no elevation information), and also the results obtained from the real testbed experiments for the validation of the model.

In the scenario where the drone is flying at 100 meters, it attains LOS in most cases, since the drone was high enough above the terrains. Thus, the difference when including the elevation information is not noticeable. A slight difference is shown when the distance is about 1.3 kilometers. A drop of the packet delivery ratio is indicated in the 3D simulation as a result of having the DEM indicating a high elevation in that specific location. This is due to the lack of accuracy of the DEM for this specific scenario. Since the resolution of NASA's DEM is 90 meters, there might be a chance that at that particular points there is a change of elevation within 90 meters of the DEM tile. However, for the other scenario, it is found to be adequate.

The case when the UAV flew lower (at 40 meters) leads to NLOS conditions. The real impact of using the elevation information in our proposed 3D simulation is noticeable here. The outcome would not be absolutely the same as the real results from the testbed had we not included the elevation information in the propagation model. Contrarily, by performing the 3D simulation, the communications are clearly affected by the high elevation of the terrain. If we merely perform a 2D simulation, at a distance of 2 kilometers, the packet delivery ratio remains quite high (more than 60%). On the other hand, no communications were taking place at that particular distance in the real tested experiments. The simulation including 3D communications showed that, at that distance, the communication was obstructed by the terrains since it includes the elevation information. This proves that an acceptable level of realism is reached with our proposed simulation model as a result of including the terrain model proposed.

Finally, in order to have more detailed simulation results, including how much packet reception is affected by the signal quality, we have also recorded the RSSI (Received Signal Strength Indicator) value in our simulation. Figure 10 shows the RSSI values based on the distance between the transmitter (the car) and the receiver (the UAV). These figures include both the case when the UAV flew at 100 meters (Fig. 10a), and at 40 meters (Fig. 10b). In each figure, we have included the results when using 3D communications being represented by the red line. The other line, in blue, included in each figure, represents the simulation results where we did not include the elevation information to define the signal quality, or in other words, we did not consider 3D communications.

Fig. 10a, which represents the results from the simulation when the UAV flew at 100 meters, shows the reason why packets are lost at certain distances. The simulation spotted that, when the distance is between 1200 and 1300 meters, there is a drop on the signal obtained at the receiver's side. The RSSI indicates the lowest value, at about −115 dBm,



(a)

(b)

FIGURE 10. Received Signal Strength indicator (RSSI) depending on the distance between the car and the UAV. (a) Scenario 1. UAV Altitude of 100m. (b) Scenario 2. UAV Altitude of 40m.

at a distance of 1300 meters. At that particular location, the signal path was obstructed by a higher terrain, resulting in a higher loss of the signal strength. However, for distances greater than 1400, the signal strength indicator showed a similar value for both when the simulation considers the 3D communications model, and the one with only the 2D communications model.

On the other hand, the results when the UAV flew lower (40 meters) as shown in Fig. 10b, denotes that for most cases the signal path experiences NLOS conditions. In more detail, compared to the packet delivery ratio results, we can see that the RSSI is much lower when the 3D communications model is used. When the distance is about one kilometer, the signal strength drops to about −110 dBm compared to the signal strength when using the 2D communications model, where the signal remains higher at −80 dBm. Another significant difference, this time in terms of packet delivery ratio, is detected at 1300 meters, since no packets were received at the receiver's side at that location. This is also confirmed by the results regarding the RSSI. The figure shows that, when we consider 3D communications, the RSSI values range from −110 dBm to −130 dBm. Instead, when the 2D communications model is used, the RSSI ranges from −80 dBm to −105 dBm. These data further confirm that elevation information significantly affects the signal path since, at some location, the simulation spotted some terrains that act as obstructions to the signal propagation.

VI. IMPACT OF DIFFERENT DIGITAL ELEVATION MODELS ON PERFORMANCE

In this section, we compare the two elevation models included in the simulation framework. The default elevation model is the NASA's DEM. Using this DEM in the simulation we can record the altitude of the car when it follows the route based on the geocoordinates obtained from the simulation.

For the sake of validating the effectiveness of using the appropriate DEM file, we will also compare the results using the NASA's DEM with the ones obtained using Google's DEM (as described in subsection III-A). The goal is to compare its overhead in terms of execution time, and also its performance results in terms of Packet Delivery Ratio and the Received Signal Strength Indicator (RSSI).

In addition, since in our previous real experiments we have obtained the trace detailing the elevation of the terrain, we will use this real trace as a comparison to the obtained results in the simulation. In this case, the car acting as data receiver records its location, providing the geocoordinates used to retrieve the elevation value; such value will be added to the height of the car in the simulation, thereby obtaining the correct altitude. We will use three methods: first, we use the altitude recorded in the real experiment for simulation, getting the Packet Delivery Ratio and the RSSI. Second, based on the vehicle's geocoordinates, we will retrieve the elevation information using the NASA's DEM. In the third method, we will retrieve the elevation information using Google's DEM.

A. IMPACT OF DIGITAL ELEVATION MODELS ON SIMULATION TIME

As an effort to select the more efficient DEM for simulation, we have compared the execution time required by the different methods for retrieving the elevation information. In particular, we will compare five methods that run our simulation model in 3D communications.

The results of the execution time are shown in Fig. 11. The first bar (the blue one), is the real experiment itself. It represents how much time is needed to run the experiment in the field. The second one, represented by the yellow bar in the figure, is the simulation using the real trace obtained from the previous real experiment to gather information about the car's altitude as it moves along the trajectory. The third one, represented as a green bar, also works by retrieving the car's altitude, and the signal path from the UAV to the car is determined using the NASA's DEM. The fourth one,

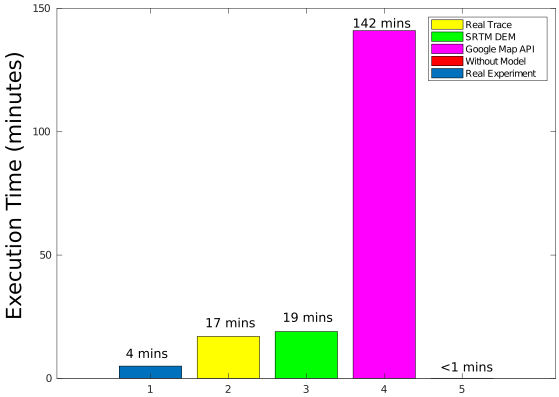


FIGURE 11. Execution Time for the different Elevation Models.

represented as a purple bar, uses the same method as the latter, but retrieves elevation data using the Google Maps Elevation API (Google's DEM). Concerning the last one, represented as a red bar, refers to the simulation experiment when 3D communications are not considered, meaning that no elevation information is included, and it is included as a reference.

Our assessment of the execution time is made by determining the time required for measuring the communications between the UAV and the car as the latter moves along the trajectory from the starting point (where the UAV is located) to the destination point, right before it makes a U-turn, in both the simulation and testbed experiments.

As shown in Fig. 11, the real experiment itself takes about four minutes, for a car speed of about 40 km/h. However, if we execute a simulation using the same parameters using the NASA's DEM for the elevation information, it takes about 19 minutes to execute. This is expectable since retrieving the elevation information for the signal path requires significant computation time. In addition, when calculating the knife edge diffraction, the machine executing the simulation also takes more time. The same thing occurs when we use the real trace as a reference for the altitude of our receiver; in this case, the execution time is slightly lower than the NASA's DEM one, as it avoids retrieving elevation information.

The most time-consuming simulation approach is using the Google Maps Elevation API. Since the elevation information is not stored locally, it has to check online the terrain height values throughout the simulation experiment. Thus, not only do we have a slower execution time associated to connecting to the Internet, but also the Google Maps Elevation API itself consumes a significant time since it represents a very large database of elevation information. Thus, in our experiment, the time consumed was more than 2 hours for one run. This evidences that using Google's DEM is quite inefficient in terms of execution time. Compared to using the NASA's DEM, the gap is very huge. Another thing to keep in mind is the exclusivity of retrieving the elevation using Google Maps Elevation API. The request is limited to the public, and it is not adequate for the simulation if this approach is adopted by a broader number of users.

Finally, and as a reference of the overhead incurred when using 3D communication, the execution time is lower than one minute when only 2D communications are considered since it does not need more computation time to retrieve the elevation information. This simulation experiment was run with the same parameters as previous ones This shows that, even though in terms of computation time 2D communications are faster, the degree of realism is far worse, as shown in the previous section.

B. ACCURACY OF THE ALTITUDES OBTAINED BY THE ELEVATIONS MODELS

In this subsection, we evaluate the accuracy of the obtained altitudes from the NASA and Google's DEMs. Particularly, based on the results of our previous testbed experiment, where we obtained the latitude, longitude, and altitude of the car (receiver), we used them for comparison with the retrieved elevation information using the other DEMs. Thus, based on the latitude and the longitude of the car, we can retrieve the elevation information using both DEMs. The height of the car is added to the elevation obtained so that it has the absolute altitude for that particular location.

In Fig. 12, we present a comparison of the altitude of the car from the starting point to the final point, which is calculated based on the distance to the sender (UAV). The blue line represents the trace of the car's altitude obtained in the real testbed. The purple line in the figure represents the car's altitude obtained via NASA's DEM, where the elevation is obtained using the latitude and the longitude of the car's trajectory of the real testbed. The yellow line represents the results using Google's DEM, uses the same method as the previous one.

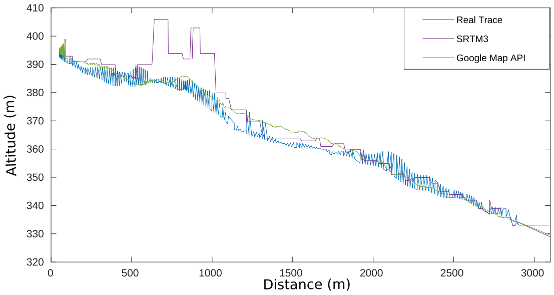


FIGURE 12. Comparison of Altitudes Recorded.

The results in the figure show that Google's DEM has the closest values to the ones obtained from the real trace, due to its higher resolution. On the contrary, the results for NASA's DEM shows a more discretized curve, not being as continuous as the line produced from Google's. This is particularly noticeable for distances between 1300 meters and 1700 meters; in this range, the line produced by the NASA's DEM is a steady line, showing that there was no change in the altitude, meaning that the Earth's surface is assumed to be flat in that location. On the other hand, the real trace showed that there is a slight drop of altitude in that slice of distance. Even though Google's DEM showed a higher value of altitude, it has the same trend as the line from the real trace, unlike the NASA's one, that has the same value of altitude (400 meters). From the figure, we can understand that the altitudes produced using Google's DEM are more accurate than NASA's ones, as they have values that better resemble the altitudes recorded in the real trace.

C. PERFORMANCE EVALUATION OF THE ELEVATION MODELS

Since we have understood the differences between using different elevation models in terms of execution time and accuracy when determining the altitude, we now proceed to compare them in terms of packet delivery ratio performance. We compare the three methods of using the different elevation models (real trace, NASA's DEM and Google's DEM). However, the main difference arises depending on the approach taken to retrieve the altitude of the receiver, in this case, the moving car. When the simulation is running, the car has to record its geolocation coordinates in order to define the signal path and the obstructions in between. For the elevation information of the signal path in the simulation, for the three experiments, we will keep using the NASA's DEM.

Fig. 13 shows the results achieved where the blue curve represents the result of running the 3D simulation using NASA's DEM. The yellow curve represents the result of running the 3D simulation using Google's DEM as a reference for obtaining the receiver's altitude. And finally, the red line represents the case when the receiver's altitude takes the data directly from the real trace.

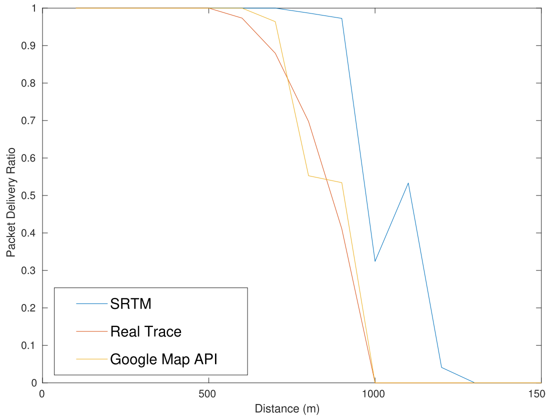


FIGURE 13. Comparison of Elevation Models on the Packet Delivery Ratio.

The figure shows that the curve produced by Google's DEM has values closer to the curve associated with the real trace. As for the curve produced by NASA's, the trend is similar to the curve from the Google's one, although that the actual values differ when compared to both the real trace and the Google's DEM. This evidences that Google's DEM, due to its greater resolution, obtains better estimation of the packet delivery rate than NASA's DEM.

For a more detailed comparison, we have also tested the simulation to get the Received Signal Strength Indicator (RSSI) value to prove the signal strength when receiving packets from the sender. We have tested the three methods using the same parameters to get the RSSI in dBm.

The results are shown in Fig. 14. The Google's DEM curve is closer to the one obtained using the real trace. We can understand that from the previous figure (Fig. 13), in terms of both RSSI and packet delivery ratio, as the values obtained are quite comparable. However, the trend is found to be more similar to the NASA's DEM. On the contrary, NASA's DEM presents the greatest difference for a distance between 1000 meters and 1300 meters. On the other hand, both the simulation using Google's DEM and the real trace show a drop of the RSSI value, which in turn results in packets losses, presenting a low packet delivery ratio.

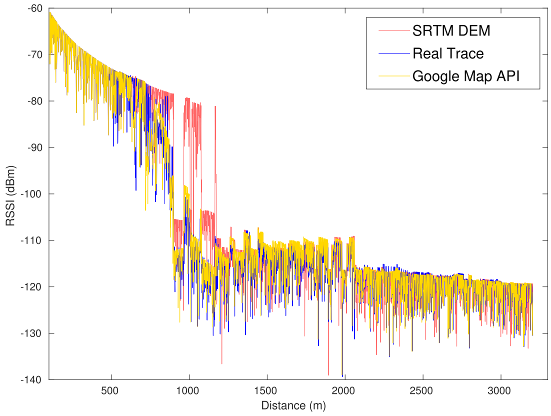


FIGURE 14. Comparison of Elevation Models on RSSI.

Summing up, although the results obtained using the Google's DEM are slightly more accurate than the ones obtained using the NASA's DEM, due to its high execution time and limited data access, currently, it is not a feasible approach to perform efficient simulations.

VII. IMPACT OF DIFFRACTION MODELS ON PERFORMANCE

In this section, we compare the different path loss models that are included in the simulation framework as described in subsection III-B. The first path loss model is a single knife edge model, where we have calculated the highest obstacle (or knife) that is present within the signal path between the transmitter and the receiver. The second path loss model is the Deygout model, and the last one is the Bullington model. The comparison is between the three models implemented in the scenario where the drone is flying at 40m, that is, the scenario with obstacles that lead to NLOS conditions.

We will compare the performance of these three models in terms of how much time is needed to execute a particular path loss model on our host. Another comparison is the packet delivery ratio obtained when running the simulation using these models. In addition, for a more detailed comparison, we have also added the RSSI values that are produced using these models.

A. PERFORMANCE OF USING DIFFERENT PATH LOSS MODELS

Firstly, we compare the different path loss models (or diffraction models) based on the packet delivery ratio at the receiver's side. That is, the ratio between the arrived packets to the receiver (the car) and the packets sent from the transmitter (the UAV). As a reference, the single knife edge diffraction model is used, as it has already been compared with the results from the real experiment. The results are shown in Fig. 15, where we can see that the outcome is basically the same. This occurs since the three diffraction models indicate the same obstacle within the signal path. However, the calculation of the signal strength might have different results for these three models.

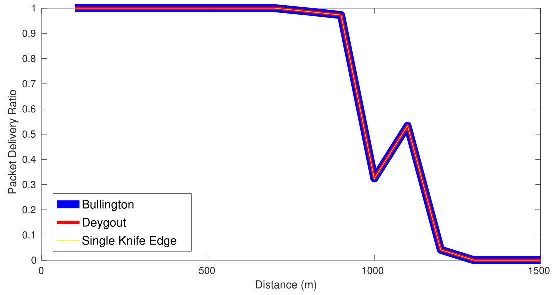


FIGURE 15. Comparison of Diffraction Models on the Packet Delivery Ratio.

Although having the same outcome in terms of packet delivery rate, it cannot justify the fact that the signal strength at the receiver's side records the same value. To prove this, we have also evaluated the RSSI for each diffraction model.

Fig. 16 represents the comparison of the RSSI values from the outcome of the simulation using the three different diffraction models, where we can see that the signal strength for the three diffraction models records a low value after 1300 meters. This is the reason why at the receiver's side, no packets are successfully delivered. The difference between the three is in fact on how low RSSI values are when detecting an obstacle. In the figure, we can understand that, when using the Deygout model, the RSSI outcome is the lowest compared to the other two. On the other hand, if we only include the single knife edge diffraction model, the RSSI value is the highest. This occurs because the signal loss is affected by a single obstacle, instead of multiple obstacles. In the Bullington and the Deygout models, since they are multiple knife edge diffraction models, we obtain lower RSSI values. Thus, the single knife approach seems to be the least adequate one as it neglects the other obstacles that might be present in the signal path, and that affect the signal quality between the transmitter and the receiver.

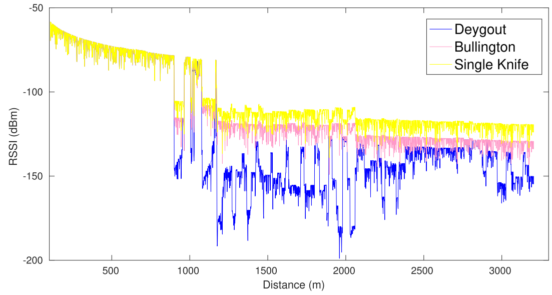


FIGURE 16. Comparison of Diffraction Models on the RSSI.

B. IMPACT OF PATH LOSS MODELS ON SIMULATION TIME

To complete our analysis, we now compare the execution time to simulate, using different path loss models. In order to have a view of which path loss model is more efficient in terms of execution time, we compare the three models using our 3D simulation framework.

Fig. 17 shows that, when using the Deygout diffraction model, the execution time is longer than for the other two (about 40 minutes). On the other hand, the simulation using the single knife edge and Bullington diffraction models show very similar values. In the case of the single knife edge, the simulation takes about 13 minutes, whereas, for the Bullington model, it takes about 14 minutes. We can conclude from the figure that the Bullington model offers the best trade-off between execution time and accuracy.

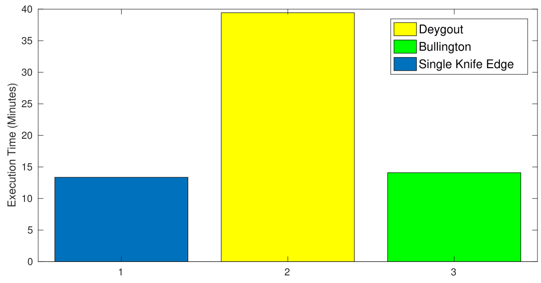


FIGURE 17. Execution time for the different diffraction models.

Summing up, the best option is to use the Bullington model, given its slightly higher accuracy and low execution time. A comparison of all the results obtained is presented in table 3 for the sake of completeness.

TABLE 3. Summary of results based on methods used in the simulation.

Diffraction Method	Elevation Model	Deployment	PDR Accuracy	RSSI Accuracy	Simulation Time
None (2D)	None (2D)	very easy	not accurate	not accurate	very fast
Single knife	SRTM	easy	less accurate	less accurate	fast
Single knife	Google Elevation API	very complex & limited	more accurate	more accurate	very slow
Deygout	SRTM	very complex	more acccurate	most accurate	slow
Bullington	SRTM	complex	more accurate	more accurate	moderate

VIII. CONCLUSIONS

In the paper, in order to make simulation experiments involving UAV-to-Car communications more realistic, we have proposed a simulation framework that includes 3D terrain profiling. In fact, when characterizing the communications involving UAVs and ground vehicles, we found that standard approaches defined by a planar space are inadequate, as communications are affected by the presence of terrains having certain height or elevation; in such cases, 3D space enabled communications should be considered instead.

We can retrieve information about the terrain profile thanks to the DEM made available by NASA and by Google. In fact, terrain information is critical to calculate the attenuation of the signal for this kind of communication. Therefore, in this paper we propose a simulation framework that provides a specific propagation model able to model diffraction effects caused by hilly terrains, and we have implemented our model in the OMNeT++ simulation tool. Our model is then validated by performing several simulation experiments to analyze the packet delivery success of UAV-to-Car communications. The simulation results obtained are then compared with those obtained from real experiments. We find that, if properly tuned, simulation experiments show results that are comparable to the ones from the real testbed experiments for both scenarios tested (varying UAV flight altitude).

As future work, we consider improving the model for urban scenarios as well, including buildings as 3D obstacles. In that case, information about the height of each building is required to properly assess its impact as an obstacle to communication when determining the associated signal attenuation. We will also perform extensive tests in urban scenarios using a greater number of vehicles. In addition, we will improve the applicability of our model, and make our extension to the simulator freely available in the future.

APPENDIX

ACRONYMS

API	Application Programming Interface
BSMs	Basic Safety Messages
DEM	Digital Elevation Model
FANET	Flying Ad-hoc Networks
ITS	Intelligent Transport Systems
ITU	International Telecommunications Union
LOS	Line-of-Sight
OSM	Open Street Map
NLOS	Non Line-of-Sight
PDR	Packet Delivery Rate
RSSI	Received Signal Strength Indicator
SRTM	Shuttle Radar Topography Mission
UAVs	Unmanned Aerial Vehicles
UTM	Universal Transverse Mercator
V2V	Vehicle-to-Vehicle

REFERENCES

[1] I. Bekmezci, O. K. Sahingoz, and Ş. Temel, "Flying ad-hoc networks (FANETs): A survey," *Ad Hoc Netw.*, vol. 11, no. 3, pp. 1254–1270, 2013.

[2] K. Daniel, B. Dusza, A. Lewandowski, and C. Wietfeld, "Airshield: A system-of-systems muav remote sensing architecture for disaster response," in *Proc. IEEE 3rd Annu. Syst. Conf.*, Mar. 2009, pp. 196–200.

[3] E. Yanmaz, M. Quaritsch, S. Yahyanejad, B. Rinner, H. Hellwagner, and C. Bettstetter, "Communication and coordination for drone networks," in *Ad Hoc Networks*. Cham, Switzerland: Springer, 2017, pp. 79–91.

[4] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "UAV-enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 22–28, Mar. 2017.

[5] W. Fawaz, R. Atallah, C. Assi, and M. Khabbaz, "Unmanned aerial vehicles as store-carry-forward nodes for vehicular networks," *IEEE Access*, vol. 5, pp. 23710–23718, 2017.

[6] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, 2nd Quart., 2016.

[7] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, May 2016.

[8] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 2898–2904.

[9] R. Sun and D. W. Matolak, "Air–ground channel characterization for unmanned aircraft systems part II: Hilly and mountainous settings," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 1913–1925, Mar. 2017.

[10] A. Brummer, R. German, and A. Djanatliev, "On the necessity of three-dimensional considerations in vehicular network simulation," in *Proc. 14th Annu. Conf. Wireless-Demand Netw. Syst. Services (WONS)*, Feb. 2018, pp. 75–82.

[11] D. Santos, J. Pinto, R. J. Rossetti, and E. Oliveira, "Adding the third dimension to urban networks for electric mobility simulation: An example for the city of porto," in *Developments and Advances in Intelligent Systems and Applications*. Cham, Switzerland: Springer, 2018, pp. 199–214.

[12] S. A. Hadiwardoyo, E. Hernández-Orallo, C. T. Calafate, J. C. Cano, and P. Manzoni, "Experimental characterization of uav-to-car communications," *Comput. Netw.*, vol. 136, pp. 105–118, May 2018.

[13] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," in *Proc. 1st Int. Conf. Simul. Tools Techn. Commun., Netw. Syst. Workshops*, 2008, p. 60.

[14] J. P. Wilson and J. C. Gallant, *Terrain Analysis: Principles and Applications*. Hoboken, NJ, USA: Wiley, 2000.

[15] E. Yanmaz, R. Kuschnig, and C. Bettstetter, "Achieving air-ground communications in 802.11 networks with three-dimensional aerial mobility," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 120–124.

[16] Y. Shi, R. Enami, J. Wensowitch, and J. Camp, "Measurement-based characterization of LOS and NLOS drone-to-ground channels," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.

[17] S. Jia and L. Zhang, "Modelling unmanned aerial vehicles base station in ground-to-air cooperative networks," *IET Commun.*, vol. 11, no. 8, pp. 1187–1194, Jun. 2017.

[18] P. Shilin, R. Kirichek, A. Paramonov, and A. Koucheryavy, "Connectivity of VANET segments using UAVS," in *Proc. Int. Conf. Next Gener. Wired/Wireless Netw.* St. Petersburg, Russia: Springer, 2016, pp. 492–500.

[19] H. Seliem, M. H. Ahmed, R. Shahidi, and M. S. Shehata, "Delay analysis for drone-based vehicular ad-hoc networks," in *Proc. 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–7.

[20] S. Filiposka, D. Trajanov, and M. Vuckovik, "Performances of clustered ad hoc networks on 3d terrains," in *Proc. 2nd Int. Conf. Simulation Tools Techn.*, 2009, p. 45.

[21] H. H. Nguyen, S. Krug, and J. Seitz, "Simulation of 3D signal propagation based on real world terrains for ad-hoc network evaluation," in *Proc. 9th IFIP Wireless Mobile Netw. Conf. (WMNC)*, Jul. 2016, pp. 131–137.

[22] S. A. Hadiwardoyo, E. Hernández-Orallo, C. T. Calafate, J. Cano, and P. Manzoni, "Evaluating UAV-to-Car communications performance: Testbed experiments," in *Proc. IEEE 32nd Int. Conf. Adv. Inf. Netw. Appl. (AINA)*. Kraków, Poland: Pedagogical Univ. of Kraków, May 2018, pp. 86–92.

[23] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO—Simulation of Urban MObility," in *Proc. 3rd Int. Conf. Adv. Syst. Simulation (SIMUL)*, Barcelona, Spain, vol. 42, 2011, pp. 63–68.

[24] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 3–15, Jan. 2010.

[25] T. G. Farr *et al.*, "The shuttle radar topography mission," *Rev. Geophys.*, vol. 45, no. 2, 2007, doi: 10.1029/2005RG000183.

[26] G. Developers. (2018). *Google Maps Elevation API*. [Online]. Available: https://developers.google.com/maps/documentation/elevation/intro

[27] K. Bullington, "Radio propagation at frequencies above 30 megacycles," *Proc. IRE*, vol. 35, no. 10, pp. 1122–1136, Oct. 1947.

[28] J. Deygout, "Multiple knife-edge diffraction of microwaves," *IEEE Trans. Antennas Propag.*, vol. AP-14, no. 4, pp. 480–489, Jul. 1966.

[29] *Propagation by Diffraction*, document P.526-11, International Telecommunication Union, 2009.

[30] *Propagation by Diffraction*, document P.526-14, International Telecommunication Union, 2018.

[31] W. C. Lee, *Mobile Communications Engineering*. New York, NY, USA: McGraw-Hill, 1982.

[32] Libcurl. (2018). *The Multiprotocol File Transfer Library*. [Online]. Available: https://curl.haxx.se/libcurl/

[33] JSONC. (2018). *A c++ Library for Interacting With JSON*. [Online]. Available: https://github.com/open-source-parsers/jsoncpp

[34] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervasive Comput.*, vol. 7, no. 4, pp. 12–18, Oct. 2008.

SEILENDRIA A. HADIWARDOYO (S'11–GS'12) received the B.Eng. degree in computer engineering from the Fakultas Teknik, Universitas Indonesia, Depok, Indonesia, the Pre-Master's Diploma degree (Maîtrise, equivalent to Pg.Dip.) in computer science from the Université de Lille 1, Villeneuve-d'Ascq, France, in 2012, the M.S. degree in computer science, with a minor in web, multimedia, and networks, from the Université de Bretagne-Sud, Vannes, France, in 2013, and the Predoctoral Diploma degree (equivalent to M.Phil.) in informatics from the Universidade do Minho, Braga, Portugal, in 2015. He is currently pursuing the Ph.D. degree with the Department of Computer Engineering, Universitat Politècnica de València, Valencia, Spain. In 2018, he was a Visiting Ph.D. Research Student with the National Institute of Informatics, Tokyo, Japan. His research interests include ad hoc and vehicular networks, mobile applications, and the implementation of intelligent transportation systems.

CARLOS T. CALAFATE received the degree (Hons.) in electrical and computer engineering from the University of Oporto, Portugal, in 2001, and the Ph.D. degree in informatics from the Universitat Politècnica de València (UPV), Spain, in 2006. He has been with UPV, since 2002, where he is currently a Full Professor with the Department of Computer Engineering. He has published more than 350 articles, several of which in journals, including the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE/ACM TRANSACTIONS ON NETWORKING, the *Elsevier Ad hoc Networks*, and the *IEEE Communications Magazine*. His research interests include ad hoc and vehicular networks, UAVs, smart cities and the IoT, QoS, network protocols, video streaming, and network security. He is a Founding Member of the IEEE SIG on Big Data with Computational Intelligence. He is an Associate Editor for several international journals. He has participated in the TPC of more than 250 international conferences.

JUAN-CARLOS CANO received the M.Sc. degree and the Ph.D. degree in computer science from the Universitat Politècnica de València (UPV), Spain, in 1994 and 2002, respectively. From 1995 to 1997, he was a Programming Analyst with the IBM's Manufacturing Division, Valencia. He is currently a Full Professor with the Department of Computer Engineering, UPV. His current research interests include vehicular networks, mobile ad hoc networks, and pervasive computing.

YUSHENG JI (M'94–SM'18) received the B.E., M.E., and D.E. degrees in electrical engineering from the University of Tokyo. She joined the National Center for Science Information Systems, Japan, in 1990. She is currently a Professor with the National Institute of Informatics, Japan, and the Graduate University for Advanced Studies. Her research interests include network architecture, resource management, and quality of service provisioning in wired and wireless communication networks. She has served as a Board Member of Trustees of the IEICE, a Steering Committee Member of the Quality Aware Internet SIG, and the Internet and Operation Technologies SIG of IPSJ. She has also served as a TPC Member of many conferences, including the IEEE INFOCOM, the ICC, the GLOBECOM, and the VTC. She is an Expert Member of the IEICE Technical Committees on Internet Architecture and on Communication Quality. She was a Symposium Co-Chair of the IEEE GLOBECOM, in 2012 and 2014. She is a Track Co-Chair of the IEEE VTC, in 2016. She was an Associate Editor of the IEICE Transactions and the IPSJ Journals, a Guest Editor-in-Chief, a Guest Editor, and a Guest Associate Editor of the Special Sections of the IEICE Transactions, and a Guest Associate Editor of the Special Issues of the IPSJ Journals. She is an Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.

ENRIQUE HERNÁNDEZ-ORALLO (M'04) received the M.Sc. and Ph.D. degrees in computer science from the Universitat Politècnica de València, Spain, in 1992 and 2001, respectively. From 1991 to 2005, he was with several companies in real-time and computer networks projects. He is currently an Associate Professor with the Department of Computer Engineering, Universitat Politècnica de València. He is also a member of the Computer Networks Research Group. He has participated in over ten Spanish and European research projects. He has authored about 100 journal and conference papers and has co-authored two successful books of C++ in the Spanish language. His areas of interest include distributed systems, performance evaluation, mobile and pervasive computing, and real-time systems. He is mainly involved in the performance evaluation of MANET and opportunistic networks.

PIETRO MANZONI (M'95–SM'17) received the master's degree in computer science from the Università degli Studi of Milan, Italy, in 1989, and the Ph.D. degree in computer science from the Politecnico di Milano, Italy, in 1995. From 1992 to 1993, he did an internship at the Bellcore Labs, Red Bank, NJ, USA. In 1994, he was a Visiting Researcher with the International Computer Science Institute, Berkeley, CA, USA. He is currently a Full Professor of computer engineering with the Universitat Politècnica de València, Spain. His research activity is related to networking and mobile systems and applied to intelligent transport systems, smart cities and the Internet of Things, and community networks (mesh). He is the Coordinator of the Computer Networks Research Group, Department of Computer Engineering.