Simulation Framework for Evaluating Video Delivery Services over Vehicular Networks

P. Pablo Garrido Abenza, Pablo Piñol Peral, Manuel P. Malumbres, O. López-Granado
Physics and Computer Architecture Dept.
Miguel Hernández University, Spain
e-mail: {pgarrido, pablop, mels, otoniel}@umh.es

***Abstract*—Vehicular Ad-hoc Networks contribute to the Intelligent Transportation Systems by providing a set of services related to traffic, mobility, safe driving, and infotainment applications. One of the most challenging applications is video delivery, since it has to deal with several hurdles typically found in wireless communications, like high node mobility, bandwidth limitations and high loss rates. In this work, we propose an integrated simulation framework that will provide a multilayer view of a particular video delivery session with a bunch of simulation results at physical (i.e., collisions), MAC (i.e., packet delay), application (i.e., % of lost frames), and user levels (i.e., perceptual video quality). With this tool, we can analyze the performance of video streaming over vehicular networks with a high level of detail, giving us the keys to better understand and, as a consequence, improve video delivery services.**

***Index Terms*—Vehicular networks, Video delivery, QoS, QoE, HEVC, OMNeT++, Veins, SUMO**

I. Introduction

Among the potential applications that may be supported by Vehicular Ad-hoc Networks (VANETs), video delivery is one of the most resources demanding. Several application scenarios may require video delivery services in both on-demand and real-time live video streaming, using unicast, multicast or broadcast communications. We may find scenarios where video delivery is required, like the ones related to accidents/rescue assistance, V2X real-time video, context-aware video broadcasts, security surveillance services, driving assistance, etc. However, multimedia streaming over VANETs is a very challenging issue mainly due to the high mobility of the vehicles, the bandwidth limitations, and the high loss rates typically found in wireless communications. In addition, video transmission requires a high bandwidth with a bounded packet delay, specially when real-time restrictions are mandatory. So, when video suffers from high packet losses and/or highly variable packet delays, the user perceived video quality is seriously reduced.

In this work, we analyze the impact of all these factors on the video streaming application performance to know how video delivery is degraded in VANET scenarios. In order to send real video data, we use the High Efficiency Video Coding (HEVC) encoder [1], as well as the corresponding decoder.

We have developed a simulation framework named Video Delivery Simulation Framework over Vehicular Networks (VDSF-VN), which will allow us to model in detail the different actors involved in a video streaming session. We have choosen the OMNeT++ simulator [2], together with the Veins (VEhicles In Network Simulation) framework [3] to conduct the network simulations, and with SUMO (Simulation of Urban MObility) [4], as the vehicular simulator. The urban maps used for the scenario in this work have been downloaded from OpenStreetMap [5]. In addition, it has been necessary to develop several software modules, such as an OMNeT++ project ('ppp_qos'), and a packetizer/depacketizer tool. Finally, a graphical application named GatcomVideo was also developed in order to improve the usability and automation of the proposed framework VDSF-VN.

The remainder of the paper is organized as follows. First, in Section II, some existing simulation frameworks are briefly described, analyzing its properties and drawbacks. Then, in Section III, our proposal is described. To show its potential and flexibility, in Section IV, we describe the setup process for a particular VANET scenario, and, in Section V, we discuss some results of the proposed experiments. Finally, in Section VI, some conclusions and future work are drawn.

II. Related Work

EvalVid [6] represents a simulation framework and toolset for quality evaluation of video transmission over a real or simulated communication network. Besides measuring the Quality-of-Service (QoS) parameters of the underlying network, such as loss rate, delay, and jitter, they support also a video quality evaluation of the received video based on the frame-by-frame Peak Signal-to-Noise Ratio (PSNR) calculation. EvalVid is a popular framework, however no network simulator is proposed (authors explain that whatever simulator may be embedded), and the video codec included is MPEG4. Several works extend EvalVid to include a particular network simulator (ns-2, ns-3, etc.) or update the MPEG4 codec with current video coding standards.

One of these works is proposed by Rosario et al. in [7]. Their simulation framework, Mobile Multi-Media Wireless Sensor Networks (M3WSN), is based on EvalVid, OMNeT++ and Castalia frameworks [8]. It supports multimedia transmission for fixed and/or mobile wireless video sensor nodes. Although it uses realistic modeling of wireless video sensor communications, it does not include flexible mobility models, like the ones demanded by VANETs, and uses the same video encoder than EvalVid.

978-1-5386-6358-5/18/$31.00 ©2018 IEEE

In [9], authors propose the design and implementation of a novel open-source tool, named QoE Monitor, which consists of a new module for the ns-3 simulator which can be used to perform Quality-of-Experience (QoE) assessments in any simulated network. Their framework is based on the EvalVid architecture, redesigning some of its software modules and integrating it with the ns-3 simulator.

Despite of the different existing video frameworks (most of them based on the initial EvalVid approach), none of them allows the transmission and quality evaluation of video sequences with the combination of the OMNeT++ simulator, the Veins framework and the SUMO traffic mobility model for vehicular networks. Some analyzed frameworks lack of an updated video codec module, being not trivial to change it with another one, because the packetizer needs to be properly adapted to the intrinsic features of the target bitstream format. Also, node mobility models are too simple in most approaches to fit with the particularities of vehicular networks (roads, streets, lanes, stops, traffic lights, etc.). And, finally, the different modules of the framework need to be completely integrated in order to launch global simulations specifying the detailed configuration of every module, and performing, on the fly, all the processes, from the video encoding at source node to the decoding process at the receiver end, passing through the network simulation of the video delivery in a realistic vehicular network scenario. These aspects motivated us to develop a complete video evaluation framework that it is especially suited for vehicular networks.

III. Simulation Framework

In Fig. 1, we show a representation of our VDSF-VN framework, where different components are depicted. On the one hand, there is a set of tools used by the video coding management (encoder, packetizer, etc). And, in the other hand, we have the main simulation engine based on OMNeT++, Veins and SUMO packages, where we have developed a new OMNeT++ project named 'ppp_qos' which empowers and cooperates with Veins in order to add video delivery services to network nodes (RSUs, cars). Finally, a graphical application, named 'GatcomVideo', acts as a front-end for the previous components, which improves the usability, shows results in a graphical way, and automatizes the whole process.

In a previous work, we presented GatcomSUMO [10], an open source tool which assists researchers with the tedious and hard task of creating vehicular networks scenarios and the necessary configuration files for the simulation process. This tool allows the deployment of both synthetic and real map scenarios (downloaded from OpenStreetMap), creating vehicle routes, placing fixed elements like Road-Side Units (RSUs), etc., with the benefit of avoiding the execution of complex command-line orders and using a GUI application. It is targeted to the triplet SUMO/OMNeT++/Veins, a free-software paradigm for the simulation of vehicular networks which is used by a great number of projects and has an increasing users base.

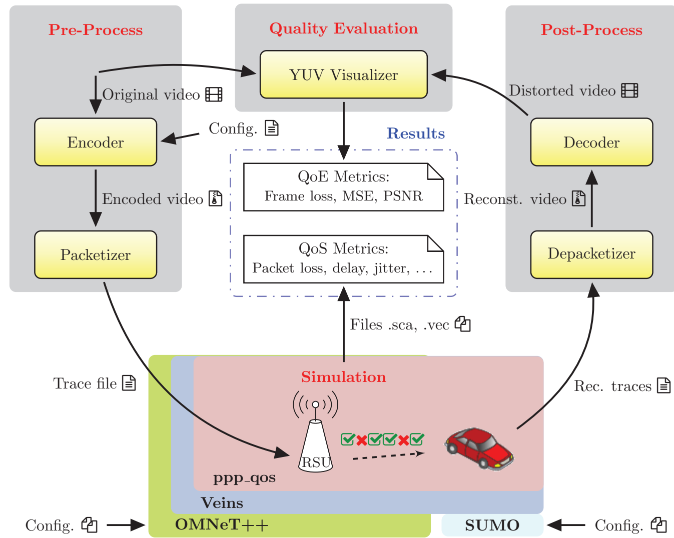

Fig. 1. Workflow of VDSF-VN

The main advantages of VDSF-VN are the following: a user-friendly GUI (avoiding scripting and command-line orders); an all-in-one environment from where the different simulation stages are managed; a framework where batch simulations can be run, taking advantage of multi-threading parallel processing when possible; a multi-platform tool which can be run in different environments (Windows, Linux, Mac OS X).

The VDSF-VN framework is structured in three main steps, as depicted in Fig. 1: (1) pre-process step, (2) simulation step, and (3) post-process step.

A. *Pre-process step*

The aim of the pre-process step is to prepare the video sequences to be used by the simulation step. For this task, we use the HEVC reference software, named HM (HEVC Test Model) [11], which encodes the selected raw video sequences. We have modified HM software to include RTP bitstream packetization [12], since an encoded frame may be larger than the network MTU, being necessary to encapsulate it into several packets.

From a specific raw video sequence a great number of possible encoded bitstreams can be generated by fixing some encoder configuration parameters in order to observe how these parameters affect the video delivery in vehicular networks. For example, we can select the desired compression mode, namely, All Intra (AI), Low-delay P (LP), Low-delay B (LB), and Random Access (RA). Another parameter is the Quantization Parameter (QP), which is used to adjust the compression level. A low QP value implies a soft quantization that will result in larger bitstreams with high video quality.

Once video encoding is done, we proceed to build a trace file from the encoded bitstream, which includes an ordered list of packets to be transmitted, where each packet is defined with the following fields: a correlative packet number, the frame type it belongs to (I, P, or B), the playback time (ms), the packet size (bytes), the frame offset of packet payload, and the total number of fragments of the current frame.

B. Simulation step

One of the contributions of the current work is the integration of video delivery in the SUMO/OMNeT++/Veins framework. The original Veins project is the base for the new OMNeT++ project named "ppp_qos", which references the original one. The modified "TraCIDemo11p" application module has been extended with many features, like statistics at application level (load, goodput, end-to-end delay) and the possibility of using video trace files. Server nodes are able to read the trace files and send the packets through the network. Client nodes write the correctly received video trace packets into a file, which is used in the post-process step. Finally, the background traffic nodes are able to send network packets at the selected bitrate, in order to define a particular background traffic. Another kind of collected statistics are related to the mobility of the vehicles, such as, the distance between selected pairs of nodes, the number of neighbors during the simulation, etc. These metrics are useful in order to check the validity of the built scenario.

The "ppp_qos" project also includes the MAC files from the Veins project, which have been extended with many statistics like the Channel Busy Ratio (CBR) or MAC queues length, per node (local) or the whole network (global), and broken down by Access Category (AC).

VDSF-VN can show the set of simulation runs existing in the OMNeT++ configuration file "omnetpp.ini", launch the SUMO server, and, finally, run in parallel the selected simulations according to the selected number of computing cores. Finally, it is possible to generate a set of graphs for any of the selected network statistics with both R and GNUplot graphing packages.

C. Post-process step

For the evaluation of the video delivery we take into consideration two kind of measurements. The network performance metrics, as explained above, and the QoE metrics, which measure the quality of the reconstructed video, giving an indication about how the user's watching experience will be. The QoE metrics considered are the Frame Loss Ratio (FLR) and the Peak Signal-to-Noise Ratio (PSNR) value of the reconstructed video, which is built from the trace files received by the video clients.

The FLR is not directly inferred from the Packet Loss Ratio (PLR) because when a fragment of an encoded frame is lost, the whole frame cannot be reconstructed. The FLR affects video quality because when a frame is lost, the decoder keeps playing the previous decoded frame. This causes a "freezing" effect in the video, which diminishes the perceived quality.

After the bitstream file is generated, we use a modified version of the HM decoder in order to get the reconstructed video sequence. The modification of the HM decoder has been mandatory because the original HM decoder crashes when any piece of information is lost, so we have strengthened the decoder to be robust against packet losses.

At last, once the video sequence is reconstructed we compute the PSNR value relating to the original video sequence, which is the most commonly used metric for measuring the video quality.

Carrier frequency	5.890 GHz
Propagation model:	
Without obstacles	SimplePathlossModel
With obstacles	SimpleObstacleShadowing
Bitrate	18 Mbps
Transmit power	20 mW
RX Sensitivity	-89 dBm
Communication range	510.87 m
MAC queues size	0 (infinite)

TABLE I
PHY/MAC PARAMETERS

IV. Simulation setup

With the proposed framework we have created a set of experiments to evaluate the video delivery performance in a particular network scenario.

A. Scenario setup

The scenario is localized in a square area (sized 2000x2000 m) of the city of Kiev (Ukraine) (see Fig. 2). Three fixed RSUs are placed along an avenue, delivering the same video sequence in a synchronized way. The parameters of the network cards are set with the values shown in Table I. The communication range is around 500m, which is the default value in Veins. The radio transmission range of the three RSUs is depicted with a blue circle in Fig. 2. During the experiment, two cars travel along the cited avenue. One of them is the video client, and the other car, which travels next to the client node, is a background traffic node, which sends packets continuously at different bitrates. The distance between both the client and the three RSUs and the number of neighbors along the entire simulation are shown in Fig. 3. The simulation time is 340 seconds, which is the time that the two cars need to travel from the beginning to the end of the avenue, at a maximum speed of 14 m/s (50 km/h). The background traffic car injects packets with a size of 512 bytes at six different rates: $\{0,125,250,500,750,1000\}$ pps.

B. Video setup

The video sequence "BasketballDrill", which belongs to the Common Test Conditions set, is transmitted in a cyclic way by all the RSUs. It has a resolution of 832x480 pixels, a length of 250 frames, and a rate of 25 frames per second (this represents 10 seconds of video). It has been encoded with the modified HM encoder, with two encoding modes: All Intra (AI) and Low-delay P (LP).

In AI mode, all the frames of the video sequence are encoded as I frames. I frames are encoded without using any other frame as reference. In LP mode, the first frame is encoded as an I frame, and the rest of the frames are encoded

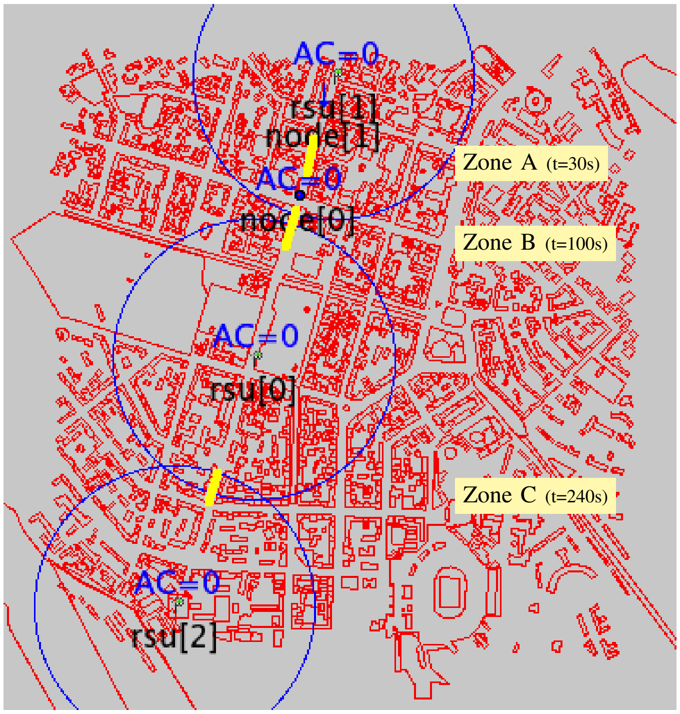

Fig. 2. City of Kiev

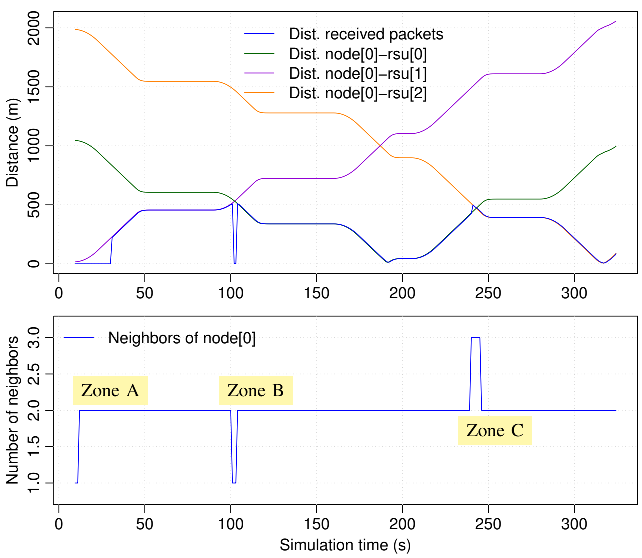

Fig. 3. Distance from client to the 3 RSUs (top) and neighbors (bottom)

as P frames. P frames use one previously encoded frame as reference. LP mode is very efficient regarding compression performance because of the use of motion estimation and compensation, but it is sensible to packet losses because of the dependencies between frames.

For our experiments, the QP value has been individually set for each one of the two modes tested, in order to get approximately the same video quality in both cases (PSNR≈36 dB). For the AI mode, a QP value of 31 is used, which produces a bitstream of 3.42 Mbps, with a PSNR value of 35.86 dB. For the LP mode, we have chosen a QP value of 28, which produces a bitstream of 0.96 Mbps, with a PSNR value of 36.16 dB. These two QP values can be obtained with the utility included in the GatcomVideo application, which automatically searches for the QP value which provides (for a certain encoding mode) a desired PSNR or bitrate value.

For the experiments we have combined the 2 encoded sequences (LP_QP28 and AI_QP31) with the different background traffic rates. These experiments are conducted without adding priorities to the video packets, that is, all the packets are sent with the same priority (AC=0). However, the framework is ready to use QoS properties by assigning different priorities to each frame depending on its type (I, P, or B).

V. Results and Discussion

From along the path traversed by the cars, we have selected three zones that represent different network situations. The first zone, Zone A, is the area where the client node has full coverage of one of the RSUs. The second one, Zone B, is a shadow area between two RSUs, where none of them are "visible" for the client node. Finally, Zone C is an area between the second and third RSU where the car is inside the coverage range of both RSUs, that is, their signals are overlapped.

Under "ideal" conditions, i.e., transmitting the encoded sequences without background traffic, as expected, we obtain a 0% PLR in zone A. In Zones B and C the % of lost packets is unmanageable. In Zone B, a revision and adjustment of the RSUs coverage should be done, whereas in Zone C, an efficient and seamless horizontal handover mechanism should be proposed.

For the tests with background traffic, due to the limited space, we only show the results in Zone A. In Fig. 4 (top), the PLR and FLR are presented for both AI and LP encoding modes, at different background traffic loads. We can see that, even though the PLR keeps under certain limits for both encoding modes (it is always lower than 12%), it produces high values of FLR, especially in the AI case. For this encoding mode, only a very low background traffic of 125 pps (0.5 Mbps) keeps the FLR around 30%, and for the rest of the traffic conditions, the FLR has very high values, in most cases around 80%. This happens because I frames are usually bigger than MTU and a high fragmentation of frames appears. This fact entails a high FLR even with a low PLR, because, as explained before, the real loss of just one network packet of the frame implies the effective loss of the whole frame.

In Fig. 4 (bottom), the PSNR values for the reconstructed video in Zone A, under different conditions of background traffic and the two evaluated encoding modes are shown. It can be seen that with no background traffic, the LP mode is more efficient, because it gets the same video quality than AI requiring a much lower bitrate (0.96 Mbps vs. 3.42 Mbps). But when some background traffic is present, LP mode obtains unacceptable low PSNR values, due to the existing dependencies between frames. For AI mode, only when a low

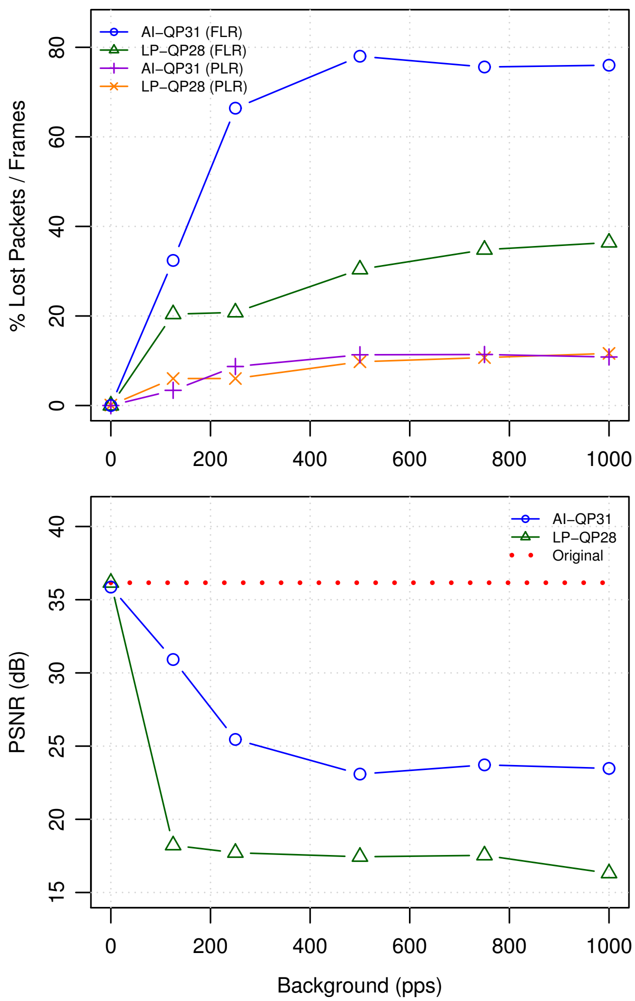

Fig. 4. QoE metrics: PLR vs. FLR (top) and PSNR (bottom)

background traffic rate is present (125 pps / 0.5 Mbps), the obtained PSNR keeps over 30 dB, which can be considered an acceptable value for the QoE of a user. This graph reveals that, even though AI mode is a robust encoding mode, it is not enough to provide protection to the video transmissions, especially with hard background traffic conditions.

VI. Conclusions and Future Works

A complete framework for the study and analysis of video delivery over vehicular networks, VDSF-VN, has been presented. It is based in the triplet of simulators formed by SUMO, OMNeT++ and Veins, and extends their capabilities with the pre-processing, network simulation and post-processing of video sequences in these scenarios. A set of experiments have been performed to show the potential of our framework. Notice, that we can define whatever scenario with any traffic configuration by means of GatcomSUMO, and the source video may be encoded with the desired coding configuration parameters to evaluate its performance under different network conditions.

As a simple example, we defined a urban scenario to analyze the behavior of two HEVC video coding modes, AI and LP, showing interesting findings through several application statistics. As we have shown, good RSU coverage should be planned (avoid shadow areas), and efficient handoff techniques are needed (collision areas). Even though, the quality of the received video is very poor due to the correlation between packet and video frame losses.

In order to improve the received video quality, we are currently working in several areas: (1) QoS at MAC level (as the one provided by IEEE 802.11p); (2) Forward Error Correction techniques; (3) protection of the video streams at encoding stage (slices, tiles, intra refresh, etc.); and (4) Error Concealment approaches. All of these techniques will be integrated in our simulation framework.

Acknowledgments

This work has been supported by the Spanish Ministry of Economy and Competitiveness under Grant TIN2015-66972-C5-4-R, co-financed by FEDER funds (MINECO/FEDER/UE).

References

[1] "High Efficiency Video Coding (HEVC)," ITU-T Recommendation H.265, 2013.
[2] A. Varga and R. Hornig, "An Overview of the OMNeT++ Simulation Environment," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, ser. Simutools '08, 2008, pp. 60:1–60:10.
[3] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2011.
[4] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent Development and Applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, 2012.
[5] M. M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.
[6] J. Klaue, B. Rathke, and A. Wolisz, *EvalVid – A Framework for Video Transmission and Quality Evaluation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 255–272.
[7] D. Rosário, Z. Zhao, C. Silva, E. Cerqueira, and T. Braun, "An OMNeT++ Framework to Evaluate Video Transmission in Mobile Wireless Multimedia Sensor Networks," in *Proceed. of the 6th International ICST Conference on Simulation Tools and Techniques*, 2013, pp. 277–284.
[8] D. Pediaditakis, Y. Tselishchev, and A. Boulis, "Performance and scalability evaluation of the Castalia wireless sensor network simulator," in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, 2010, p. 53.
[9] D. Saladino, A. Paganelli, and M. Casoni, "A tool for multimedia quality assessment in NS3: QoE Monitor," *Simulation Modelling Practice and Theory*, vol. 32, pp. 30 – 41, 2013.
[10] P. P. Garrido Abenza, M. P. Malumbres, and P. Piñol Peral, "GatcomSUMO: A Graphical Tool for VANET Simulations Using SUMO and OMNeT++," in *Proceedings of the SUMO User Conference 2017 (SUMO2017)*, vol. 31, Berlin-Adlershof, 2017, pp. 113–133.
[11] Joint Collaborative Team on Video Coding (JCT-VC), "HEVC reference software HM (HEVC Test Model) and Common Test Conditions," https://hevc.hhi.fraunhofer.de, Joint Collaborative Team on Video Coding (JCT-VC), [Accessed May 11, 2018].
[12] Y. Wang, Y. Sanchez, T. Schierl, S. Wenger, and M. Hannuksela, "RTP Payload Format for High Efficiency Video Coding," RFC 7798, 2016.