Evaluating the Use of QoS for Video Delivery in Vehicular Networks

P. Pablo Garrido Abenza*, Manuel P. Malumbres†, Pablo Piñol Peral‡, Otoniel López-Granado§,
Dept. of Computer Engineering
Miguel Hernández University
Elche, Spain
*pgarrido@umh.es, †mels@umh.es, ‡pablop@umh.es, §otoniel@umh.es

***Abstract*—In a near future, video transmission capabilities of intelligent vehicular networks will be essential for deploying high-demanded multimedia services for drivers and passengers. Applications and services like video on demand, iTV, context-aware video commercials, touristic information, driving assistance, multimedia e-call, etc., will be part of the common multimedia service-set of future transportation systems. However, wireless vehicular networks introduce several constraints that may seriously impact on the final quality of the video content delivery process. Factors like the shared-medium communication model, the limited bandwidth, the unconstrained delays, the signal propagation issues, and the node mobility, will be the ones that will degrade video delivery performance, so it will be a hard task to guarantee the minimum quality of service required by video applications. In this work, we will study the impact of those factors on the received video quality by using a detailed simulation model of a urban vehicular network scenario. We will apply different techniques to reduce the video quality degradation produced by the transmission impairments like (a) Intra-refresh video coding modes, (b) frame partitioning (*tiles/slices*), and (c) quality of service at MAC level. So, we will know how these techniques are able to conceal as much as possible the network impairments produced by the hostile environment typically found in vehicular network scenarios. The experiments were carried out with a simulation environment based on the OMNeT++, Veins and SUMO simulators. Results show that the combination of the proposed techniques improves the robustness of video transmission in vehicular networks.**

***Index Terms*—VANET, Video, HEVC, Quality of Service**

I. INTRODUCTION

Vehicular Ad-hoc Networks (VANETs) have a great potential since they contribute to the so-called Intelligent Transportation Systems (ITS), providing a series of services in urban and inter-urban environments, for both drivers and passengers. Among them, it is worth to mention the implementation of systems related to the improvement of driving safety, traffic information, weather forecast, Internet access, and entertainment applications (*infotainment*).

Video transmission in vehicular networks can have many applications, such as the diffusion of advertisements or tourist information according to our current location (context-aware video), real-time video transmission (video streaming), video surveillance, multimedia emergency call (eVideoCall) systems, etc. However, video transmission in this kind of networks is very problematic.

On the one hand, the typical problems of wireless communications appear, such as limited bandwidth and the use of a shared medium with other devices, which generates interference signals and leads to collisions. In addition, various phenomena occur such as signal attenuation due to distance (*path loss*) or time (*fading*), the presence of obstacles (*shadowing*), and the refraction or reflection of the signals (*multipath*). On the other hand, due to the inherent mobility of the network nodes, the network topology is continously changing, and the high speed of the vehicles limits the communication time windows. All of these issues lead to an increase in the waiting time for access to the channel (*blocking time*), and also to an increase in the average number of packets lost, reducing the performance of the wireless networks compared to the wired ones [1].

With the use of video coding, the amount of data required for storage is reduced, as well as the bandwidth required for its transmission. Various video coding standards have emerged in recent years, such as High Efficiency Video Coding (HEVC) [2], which improves the compression rates of its predecessor Advanced Video Coding (AVC/H.264) [3]. Even so, the video quality perceived by the receiver can be greatly affected by the transmission impairments, specially in the case of vehicular network scenarios.

To mitigate the above mentioned problems, several techniques have been proposed in the literature with the aim of maximizing the quality of the video perceived by the user. In works such as [4] and [5], authors group these techniques into several categories: (1) admission control and bandwidth reservation, (2) Quality of Service (QoS) at the application (APP) level, (3) traffic differentiation at the Medium Access Control (MAC) level, and (4) physical layer (PHY) link adaptation.

Error control techniques usually work at the application level. In the case of using asynchronous communication protocols such as Automatic Repeat Request (ARQ), the receiver must send an acknowledgment (ACK) in case of having correctly received a packet, so the sender will retransmit the packets without any acknowledgment. However, this is not suitable for sending a video stream in real time, due to the low latency required. Another alternative is synchronous communication protocols, such as the use of Forward Error Correction (FEC), which allows reconstructing lost data from redundant information sent in additional packets, as long as the number of lost packets does not exceed a certain limit. These mechanisms use to be non-adaptive, so bandwidth can be wasted due to the unnecessarily redundant packets when the network is not saturated, or, on the contrary, the amount of redundant packets might not be enough to restore all lost packages when the network works in high load state.

Regarding the use of QoS through traffic differentiation at MAC level, it can be done in two ways: (1) defining queue planning strategies, or (2) using different priority levels. For the particular case of the transmission of video sequences, this work focuses on prioritizing the video packets according to the type of frame to which they belong (I, P, B). Three groups of experiments have been carried out by simulations: (1) not prioritizing video packets, the default Best Effort service provided by IEEE 802.11, (2) prioritizing only packets belonging to type I frames, and (3) prioritizing all video packets (frames I, P, B) over other network traffic.

We also propose the use of other error resilient techniques that use to be found at application level (video encoder): (a) Intra-refresh coding modes to reduce the effects of error propagation, and (b) frame partitioning to fight against the frame losts due to a single lost packet. These techniques are focused on error protection and may be combined with other network aware techniques like the QoS provided at MAC level. So, our main goal in this work is to evaluate the performance of these techniques when working together in vehicular network scenarios, in order to achieve video transmissions with an acceptable video quality for the final user.

The rest of the paper is structured as follows. First, in section II, a brief review of the vehicular communication standards is presented. Next, in section III, some works in the literature related to the use of QoS techniques when delivering multimedia content in wireless networks are presented. In section IV, the setup of simulation tools, vehicular network scenario, video sequences and HEVC configuration profiles is explained. The results of the experiments are discussed in section V. Finally, in section VI, conclusions and some future work are drawn.

II. COMMUNICATION STANDARDS

In the IEEE 802.11 standard [6], a MAC sublayer is defined, as well as several physical (PHY) layers. Although IEEE 802.11 is the most widespread type of wireless network, it does not include support for QoS.

The IEEE 802.11e working group defined some extensions to the IEEE 802.11 standard to provide QoS by enabling traffic differentiation at the MAC layer. In this way, it is possible to support traffic from different applications depending on their QoS specifications, such as Voice over IP (VoIP) calls, video conferencing, video surveillance, and any other application with QoS requirements. In the IEEE 802.11e standard [7] the Hybrid Coordination Function (HCF) was introduced, which defines two new access mechanisms that replace the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF) of the IEEE 802.11 standard: the HCF Controlled Channel Access (HCCA) and the Enhanced Distributed Channel Access (EDCA). The latter, the EDCA, is the one used to manage the access to the wireless channel in ad-hoc network mode (without infrastructure). To implement the traffic differentiation, the application may assign a priority value, or User Priority (UP), to each packet to send, between 0 (the lowest priority) and 7 (the highest one). When the MAC layer receives a packet from the upper layers, it maps the priority level of the packet with one of the four Access Categories (AC) defined. The AC(3) category has the highest priority, and the AC(0) the lowest one.

Nodes that implement EDCA have four service queues at the MAC layer (one for each AC), in order to classify the packets according to their priority. If there are packets to be transmitted in two or more queues, an internal (virtual) collision occurs, selecting the queue with the highest priority. In addition, the MAC layer will also treat packets differently when accessing to the medium according to the lower and upper limits of the corresponding Contention Window (CW_{min} and CW_{max}), Arbitration Inter-Frame Space Number (AIFSN), and Transmission Opportunity (TXOP) parameters.

The AIFSN parameter is an additional time to wait before transmitting a packet once the medium is free, and its value depends on the AC category assigned to the packet. In case the channel is busy, the device will have to start a *backoff* process, in which it will have to wait for a time proportional to a random value in the range $[0..CW]$, where the initial value of the contention window is CW_{min}. If after that period the medium is still busy, the contention window will increase its value depending on the AC category assigned to the packet, until it reaches the maximum value CW_{max}. The highest priority queues have lower values for CW and AIFSN to access the channel. Finally, the TXOP parameter defines a time interval in which the node can transmit without competing with other nodes to access the channel. This allows to obtain a greater performance, which will be greater for the queue that has a greater value of TXOP, as well as an increase in the global occupation of the channel.

However, in addition to the characteristics of wireless networks, vehicular networks have the additional problem of a continuously changing topology due to the high mobility of the vehicles, which causes the communications between them not to last long enough. Therefore, the set of IEEE 1609 standards was proposed, known as the Wireless Access in Vehicular Environments (WAVE) architecture, which provides a communication protocol optimized for vehicular environments (Fig. 1). In particular, the IEEE 1609.4 standard [8] specifies the extensions to the IEEE 802.11 MAC layer, for the necessary coordination between Control CHannel (CCH) and Service CHannel (SCH), including a MAC layer for each one of them (Fig. 2). The characteristics of this MAC layer are defined in the IEEE 802.11p standard [9], which is mainly based on the IEEE 802.11e standard. Specifically, it slightly modifies the default values of the EDCA parameters of the

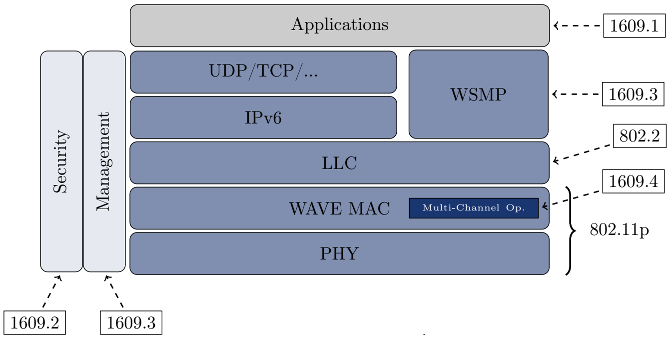


Fig. 1. WAVE reference model.

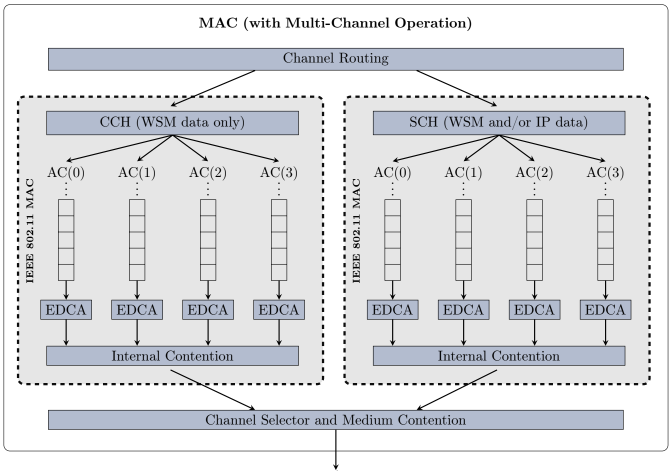


Fig. 2. MAC layer for the WAVE architecture.

TABLE I
EDCA Access Categories (AC).

AC	CW$_{min..max}$	AIFSN	TXOP$_{limit}$
AC_BK or AC(0)	15..1023	9	0 ms
AC_BE or AC(1)	15..1023	6	0 ms
AC_VI or AC(2)	7..15	3	0 ms
AC_VO or AC(3)	3..7	2	0 ms

MAC sublayer (Table I). The physical layer (PHY) of the IEEE 802.11p standard is similar to that of the IEEE 802.11a standard, but it supports a maximum transmission rate of up to 27 Mbps, instead of 54 Mbps.

WAVE supports both IP and non-IP data transmissions. Non-IP transmissions are based on the use of WAVE Short Messages (WSMs), which are defined in the WAVE Short Message Protocol (WSMP). Other standards included in the WAVE architecture are the IEEE 1609.2 one, which specifies security services, and the IEEE 1609.0 standard, which describes WAVE architecture and operations.

III. RELATED WORKS

There are different proposals in the literature that assign different priorities depending on the type of video frame, either statically [10] [11] [12], or dynamically such as the Dynamic Frame Assignment Algorithm (DFAA) [13] and others [14] [15]. For example, in [10], an application-level data partition is made (*slices*), where a priority (AC) is assigned at the IEEE 802.11e MAC level depending on the partition type, and the QoS metrics (one-way packet loss rate and packet delay). In [16] an adaptive mechanism called Adaptive Mapping Mechanism (AMM) is proposed to improve the quality of H.264/AVC video transmitted over wireless networks (WLAN) based on IEEE 802.11e, by assigning different priority based on the structure of the encoded video (frame type), the importance of the frame, and the load of each AC.

However, the works found in literature do not consider specific vehicular network scenarios, nor use the most advanced video encoding tools like HEVC video. Also, the application level error resilience techniques are not evaluated in combination with the network ones, so we can not properly assess the feasibility of video delivery over vehicular networks. In addition, most works in literature use unicast communication model, where multicast/broadcast uses to be the preferred method for vehicular networks. Using unicast or multicast/broadcast is not the same from the performance evaluation point of view, since different channel signaling approaches are used. So, in this work we will analyze the behavior of MAC level QoS provided by IEEE 802.11p working side-by-side with two error resilient techniques defined at application level (HEVC video encoder) (a) intra-refresh coding modes and (b) frame partitioning. We will use detailed simulation models to obtain accurate results that would properly assess the effectiveness of the proposed techniques.

IV. SIMULATIONS

In this section, we evaluate the proposed techniques through a simulation framework specially suited for analyzing in detail the video delivery over vehicular networks. Our goal is to accurately predict the impact of these techniques in the video delivery process, taking especially care in measuring the benefits for the final user in terms of the received video quality. So, we will describe the network scenario, the video sequences used in the tests, and the proposed experiments. Then we will analyze and discuss the main results found.

A. Network scenario

The VANET scenario used is an urban area from the city of Honolulu, Hawaii, downloaded from OpenStreetMap [17] (Fig. 3). A fixed Road Side Unit (RSU) is placed along a main avenue, which acts as a video server (`rsu[0]`), transmitting a video sequence in a cyclic way. On the other hand, several vehicles travel along the cited avenue: the first one acts as a video client (`node[0]`), and is followed by ten vehicles which act as background traffic nodes (`node[1..10]`), sending packets continuously at different bit rates for testing the scenario under different network traffic loads. Specifically, each background traffic vehicle injects packets of 512 bytes at six different rates: {0, 12, 25, 50, 75} pps, making a

Fig. 3. City of Honolulu.

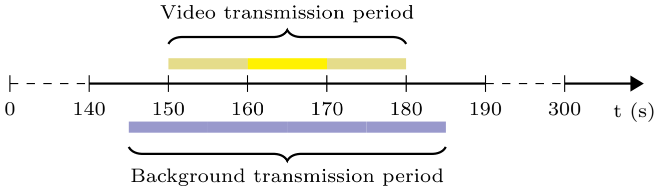


Fig. 4. Simulation timeline (in seconds).

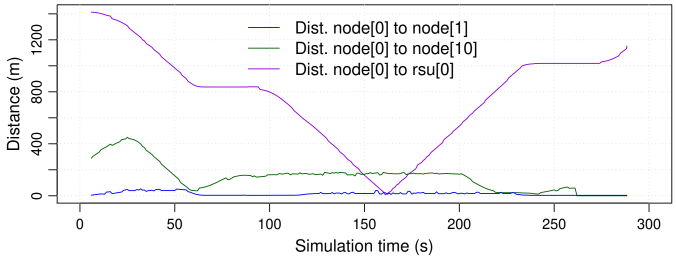


Fig. 5. Distance from client to RSU and other vehicles.

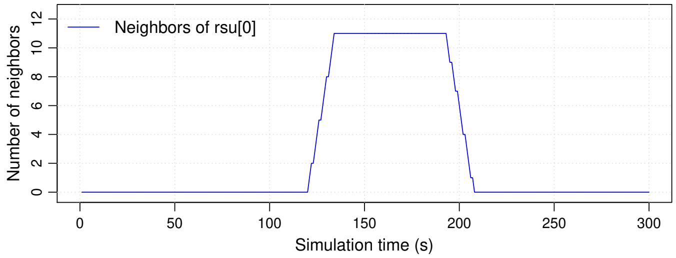


Fig. 6. Number of neighbors for the RSU.

total aggregate background traffic of {0, 0.5, 1, 2, 3} Mbps, respectively.

B. Simulation parameters

Once the scenario is defined, now we will describe the main simulation parameters we have used. All the simulations are driven by the Video Delivery Simulation Framework over Vehicular Networks (VDSF-VN) [18], which is based on the OMNeT++ v4.6 network simulator [19], together with the VEINS (VEhicles In Network Simulation) v4.4 [20] framework, and the SUMO (Simulation of Urban MObility) v0.25.0 [21] mobility simulator. With the VDSF-VN simulation tool, we may also perform the previous tasks of encoding the original video sequences (HEVC) and generating the corresponding video traces (*packetizer*) which will be used during the network simulation. This tool also carries out the analysis of the received video packets, the reconstruction of the video stream and the video decoding process to obtain the objective video quality perceived by the receiver.

The main simulation parameters are summarized in Table II. The parameters of the network cards are shown in Table III. The RSU communication range, as well as for all the vehicles, is around 500m, which is the default value used in Veins; the RSU radio transmission range is depicted with a blue circle in Fig. 3.

SUMO is a microscopic vehicular simulator, which takes into account the traffic conditions (intersections, traffic-lights, the presence of other vehicles, etc.) for adjusting the velocity of each vehicle. In the experiments, the vehicles move at a variable speed with a maximum of 14 m/s (50 km/h), and the simulations last 300s, time enough for all the vehicles to travel from the beginning to the end of the avenue. However, our interest is focused in the surroundings of the RSU, so, the results analyzed only include the interval t=[160..170]s, that is, 10s of simulation time in which the video sequence is completely received (Fig. 4).

The distance between the client vehicle (`node[0]`) and the fixed video server (`rsu[0]`), as well as the distance between the client and two of the background traffic cars (the first one and the last) along the entire simulation are shown in Fig. 5. Similarly, the number of neighbors for rsu[0] throughout the simulation is shown in Fig. 6. As can be seen, all the background vehicles are within the communication range of the client node all the time. Finally, all the vehicles are within the communication range of the video server along the region of interest.

C. Video Configuration

Two video sequences from the Common Test Conditions [22] are used in this work: BasketballDrill and RaceHorses. Both have the same resolution (832×480 pixels) and the same length (10 s). The first one, BasketballDrill, is 500 frames long at a rate of 50 frames per second (fps) in its original form, but it was sub-sampled at 25 fps with a length of 250 frames in order to reduce the required network bandwidth. The RaceHorses video sequence is 300 frames in length with a rate of 30 fps. Both video sequences were encoded with the HEVC reference software HM (HEVC Test Model) v9.0 [22] using different encoding parameters (coding mode, Intra refreshing, quantization level, frame partitioning, etc.).

By means of the Quantization Parameter (QP), the user can adjust the compression level. For our experiments, the QP value was individually adjusted for each bitstream in such a

TABLE II
Simulation parameters.

Parameter	Value
City	Honolulu
Simulation time	300s
Number of RSUs	1
Number of client vehicles	1
Number of background vehicles	10
Background traffic load	{0, 12, 25, 50, 75} pps
Max. speed of the vehicles	14 m/s (50 km/h)

TABLE III
PHY/MAC parameters.

Parameter	Value
Carrier frequency	5.890 GHz
Propagation model:	SimpleObstacleShadowing
Bitrate	18 Mbps
Transmit power	20 mW
RX Sensitivity	-89 dBm
Communication range	510.87 m
MAC queues size	0 (infinite)

TABLE IV
Bitstreams generated - 1 / 16 tiles per frame.

	Mode	QP	Bitrate (Mbps)	PSNR (dB)
BasketballDrill	AI	31	3.417 / 3.648	35.863 / 35.862
	I7P	29	1.457 / 1.587	36.071 / 36.064
	LPI4	29	1.620 / 1.766	36.045 / 36.034
	LP	28	0.959 / 1.076	36.160 / 36.193
RaceHorses	AI	31	5.802 / 6.022	36.141 / 36.133
	I7P	28	3.234 / 3.374	36.285 / 36.252
	LPI4	27	3.520 / 3.683	36.241 / 36.233
	LP	27	2.451 / 2.597	35.764 / 35.748

TABLE V
Encoding mode patterns.

Mode	Frame layout	Description
AI	IIIIIIII...	Every frame is an I frame (All Intra)
I7P	IPPPPPPP IPPPPPPP I...	An I frame followed by 7 P frames
LPI4	IPPP IPPP IPPP...	Like LP but with an I frame every 4 frames
LP	IPPPPPPPPPPPPPPP...	An I frame followed by only P frames

way so that the resulting bitstreams had the same video quality (PSNR≈36 dB), as shown in Table IV. As can be seen, the more I frames (Intra Coded) are used, the more bitrate will be necessary for achieving the same quality.

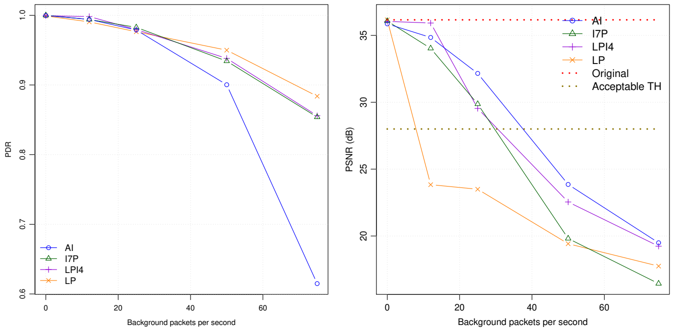
Regarding the intra-refresh coding modes, the HEVC encoder allows its configuration in a flexible way. With the All Intra (AI) coding mode, all the frames of the video sequence are encoded as I frames, which means that no frame is used as reference. As each frame is coded in an independent way, the error propagation is contained in the same frame where an error is detected, being the most robust coding mode, but requiring the highest bitrate. On the other hand, in the Low-delay P (LP) coding mode, only the first frame is coded as an I frame, followed by P frames (Predictive coded) which are not independently encoded since they require other reference frames to be encoded. This mode is very efficient regarding compression performance because of the use of motion estimation and compensation, but it is very sensible to packet losses due to the strong dependencies between frames. So, between both extremes, AI and LP, there are other coding modes that use intra-refresh by periodically inserting one I frame every group of n frames. As n increases, the error propagation effect also increases.

In previous works [23] many coding modes were analyzed, concluding that the use of a high intra-refresh degree is highly recommended for vehicular network scenarios. So, we will use the following coding modes: AI, I7P and LPI4. In addition, we will use the LP coding mode only as a reference in order to compare the improvements achieved when using the different intra-refresh degrees. All of them are summarized in Table V. The error resilience technique which we are going to use is the tile frame partitioning scheme, which proved to be very efficient to reduce the effect of single packet losses. If no partitioning is enabled, when a single packet of a frame is lost, the frame could not be decoded, resulting in the lost of the whole frame, discarding all the correctly received packets. However, the use of tiles introduces (a) an overhead in the bitstream due to the extra headers required by each tile, and (b) a lower compression efficiency. In order to study the effect of frame partitioning we are going to use different tile configurations, in particular: 1, 2, 4, 6, 8, 10 and 16 tiles per frame. These configurations are specified with a number of rows and columns, that is, in a uniform way with the following tile patterns: {1×1, 1×2, 2×2, 2×3, 2×4, 2×5, 4×4}, respectively. In this work, a total of 28 bitstreams were generated from each original raw video sequence (YUV) with different configuration parameters, using different coding modes (×4), and *tile* patterns (×7).

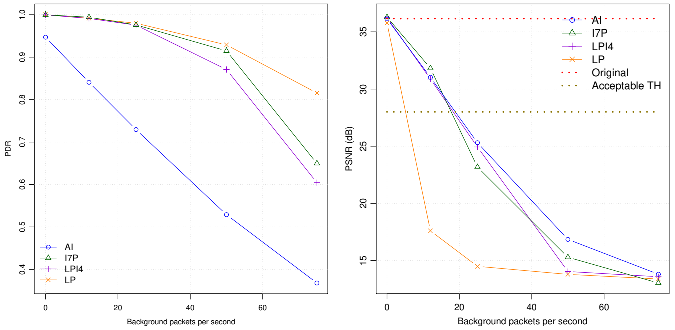
D. Experiments

The aim of this work, as mentioned before, is to evaluate the impact of using both error resilient techniques at application layer (HEVC encoder) and the available per packet QoS based on the IEEE 802.11p MAC. The evaluation is organized in several blocks, first we are going to evaluate the performance of intra-refresh coding modes to reduce the error propagation in the reconstructed video quality. So, we have compared three intra-refresh coding modes, AI, I7P and LPI4, with respect to the LP coding mode in which no intra-refresh is used, to analyze their behavior at different network loads. After the intra-refresh evaluation, we will proceed with the frame partitioning strategy by means of different tile partition setups. Here, we will measure the achieved video quality improvements of tile partitioning with respect to the non-partitioning default option (only one tile per frame). Also, we will determine the bitrate penalty of the frame partitioning scheme as the number of tiles grows, in order to find the number of tiles that maximizes the ratio between the video quality improvement and the bitrate increase. Finally, the last evaluation block will be dedicated to the use of IEEE

802.11p MAC QoS. We propose to use QoS with the video streams previously protected by means of intra-refresh coding modes and frame partitioning schemes. So, we will use the best performing protection schemes, in particular the AI and LPI4 coding modes with the 6 tiles per frame layout. Several configurations are defined: (1) assigning higher priority (AC video) to video packets belonging to I frames and the lowest priority (AC background) for the rest of video packets, just the same priority that the one used for background traffic, (2) assigning higher priority (AC video) to all video packets. For both configurations we may introduce a percentage of QoS protection to the affected video packets, P={0, 25, 50, 75, 100}%, in order to determine how the non-priority traffic is affected when competing for network resources with higher priority traffic. The motivation for carrying out the first group of experiments was the fact that the loss of an I frame is more important than the loss of any other type of frames, due to the inter-dependencies between frames.



(a) PDR (b) PSNR (dB)



(c) PDR (d) PSNR (dB)

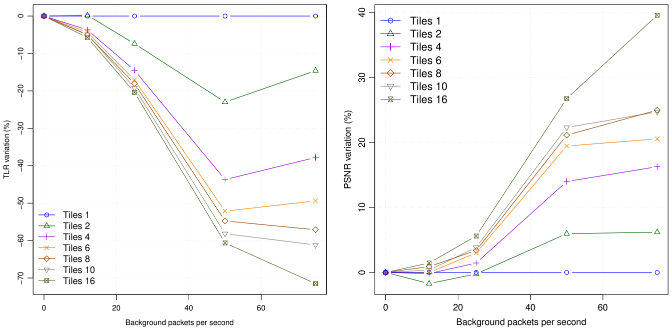
Fig. 7. Intra-refresh: BasketballDrill (top) and RaceHorses (bottom).

V. RESULTS

In this section, the results of the different experiments described above are discussed. The corresponding simulations for each experiment were run, collecting many statistics at application level (APP), Medium Access Control (MAC) and physical (PHY) levels. Some of these statistics are: the transmitted packets (Load), the received packets (Goodput or Throughput at APP level), the lost or dropped packets, the ratio between sent and received packets or Packet Delivery Ratio (PDR), the End-to-End Delay (EED) and its variation (*jitter*), size of the MAC queues for each Access Category (AC), ratio of busy channel, and number of collisions in the medium, among others. Apart from these network metrics, other metrics are computed in order to evaluate the objective quality such as the Tile Loss Ratio (TLR), and Peak Signal-to-Noise Ratio (PSNR). All the above mentioned statistics are collected for the overall simulation time, and for the region of interest; the latter are the ones presented next.

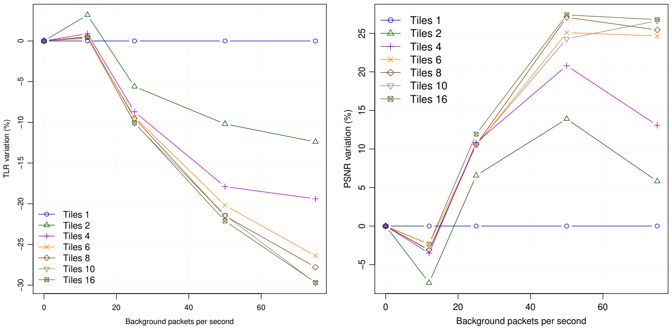
Firstly, the effect of using intra-refresh is analyzed under different background traffic loads. As expected, when no background traffic is used (0 pps), all the video packets arrive to their destination since there are no collisions. Therefore all the coding modes should achieve the maximum PDR (1.0) for all the encoded bitstreams of the first video sequence (BasketballDrill), as shown in Fig. 7a. However, for the second one (RaceHorses), the achieved PDR is slighly lower for the AI coding mode (Fig. 7c) due to the high bitrate used for that case (5.802 Mbps), as shown in Table IV, indicating that the network is close to saturation even without background traffic. As it can be seen, the LP coding mode is the most efficient regarding compression performance, as it achieves the same final video quality with the lowest bitrate (3.56 and 2.37 times less than the AI coding mode for both video sequences, respectively). However, as the background traffic grows, all the encoding modes are affected proportionally to their bitrate, being the AI coding mode the most affected one. While one might think that AI mode would reach the worst quality, the results show the opposite. From Fig. 7b, the AI coding mode achieves the best video quality in terms of PSNR for both video sequences nearly for all background traffic loads. On the other extreme is the LP mode which, despite being the mode with the lowest packet loss, it always shows PSNR values lower than the minimum threshold (28 dB). This is due to the inter-dependencies between frames of the bitstream, as the loss of a single packet belonging to a frame entails that this frame can not be decoded, as well as all those frames that reference it. The other coding modes use some level of intra-refresh, which allows to stop the temporal error propagation. This is especially important in an environment prone to packet loss such as vehicular networks. For BasketballDrill video sequence, the rest of coding modes keep acceptable values for low and medium traffic background levels. Regarding to RaceHorses (Fig. 7d), only for low traffic background levels the PSNR values are acceptable due to the greater bitrate, as said before. These results show that, despite the fact that the use of intra-refresh increases the bitrate of the bitstreams, this technique definitely improves the final video quality of the reconstructed video. Therefore, only those coding modes with a high intra-refresh were chosen for the following experiments, specifically, the AI and LPI4 ones.

Next, the benefits of using tile partitioning are evaluated with the selected coding modes in order to determine the best coding parameters (coding mode and number of tiles per frame) with the aim of getting the most robust bitstream. As will be shown below, the intra-refresh and tile partitioning



(a) TLR (Δ)

(b) PSNR (Δ)



(c) TLR (Δ)

(d) PSNR (Δ)

Fig. 8. Tile partitioning: BasketballDrill: AI (top), LPI4 (bottom).

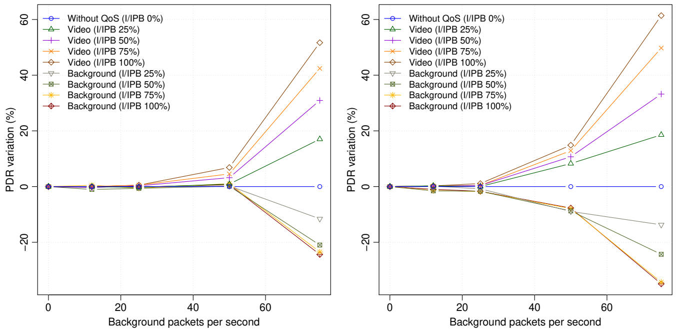
TABLE VI
PSNR VALUES FOR BASKETBALLDRILL: AI (TOP) AND LPI4 (BOTTOM).

	BGT (pps)	AI mode - Tiles per frame						
		1	**2**	**4**	**6**	**8**	**10**	**16**
No QoS	0	35.86	35.87	35.86	35.87	35.86	35.86	35.86
	12	34.84	34.24	34.78	34.84	35.15	34.97	35.35
	25	32.16	32.08	32.62	33.13	33.24	33.40	33.96
	50	23.85	25.28	27.19	28.50	28.90	29.18	30.24
	75	19.49	20.70	22.67	23.50	24.36	24.31	27.21
QoS (I/IPB)	0	35.86	35.87	35.86	35.87	35.86	35.86	35.86
	12	34.85	34.54	35.13	35.20	35.25	35.20	35.32
	25	33.12	33.80	34.05	34.13	34.34	34.09	34.53
	50	30.07	31.17	31.31	31.68	32.90	31.81	32.13
	75	25.86	27.06	29.40	29.05	30.52	29.83	31.03

	BGT (pps)	LPI4 - Tiles per frame						
		1	**2**	**4**	**6**	**8**	**10**	**16**
No QoS	0	36.04	36.05	36.05	36.05	36.05	36.05	36.03
	12	35.92	33.28	34.66	35.09	34.81	34.79	35.08
	25	29.54	31.47	32.73	32.62	32.66	32.66	33.06
	50	22.54	25.68	27.23	28.20	28.65	28.02	28.72
	75	19.24	20.36	21.75	23.98	24.13	24.36	24.39
QoS (I)	0	36.04	36.05	36.05	36.05	36.05	36.05	36.03
	12	34.05	34.64	34.74	35.28	35.59	34.93	35.51
	25	32.67	33.84	34.16	33.40	33.28	33.83	32.86
	50	28.89	29.59	30.62	30.91	30.45	28.97	30.59
	75	23.26	25.17	27.42	28.28	26.68	26.31	27.51
QoS (IPB)	0	36.04	36.05	36.05	36.05	36.05	36.05	36.03
	12	34.75	35.37	35.24	35.50	35.57	35.47	35.32
	25	33.35	35.02	33.92	34.56	34.25	34.38	34.70
	50	29.84	30.07	32.01	32.97	32.29	31.39	31.75
	75	26.97	28.74	30.40	30.16	30.33	29.58	29.30

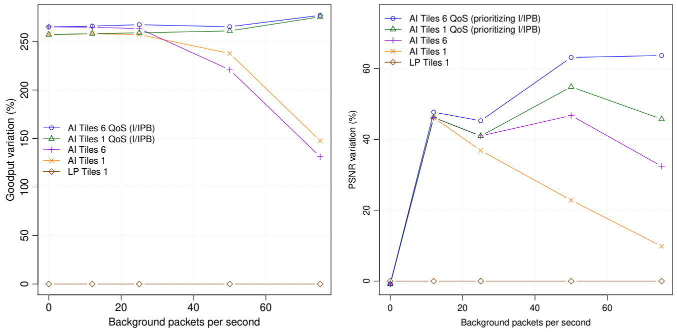
are both error resilience techniques than can be combined together in a very efficient way. As a result of the packetization process, a frame is encapsulated into several network packets before transmitting it through the network. In order to decode a frame, all its network packets should be received. The loss of just one single packet prevents the decoding of the entire frame. On the other hand, frame partitioning allows the fragmentation of each frame into independently encoded tiles, that also can be independently decoded. In this way, the loss of one packet would affect to one of the tiles only, instead of the whole frame. However, the use of tile partitioning also has the negative effect of increasing the overall bitrate (Table IV). For example, for the BasketballDrill video sequence, the bitrate is incremented by 6.76% for the AI coding mode when using 16 tiles, and by 9% for LPI4. Analyzing the TLR variation when using the different tile patterns with respect to not using tile partitioning (i.e., when using 1 tile per frame), we may conclude that the percentage of lost tiles is lower as more tiles per frame are used (see Figs. 8a and 8c). In this way, when using tile partitioning, although the network packet error rate is the same or even a bit greater due to the slightly increment in bitrate, the final video quality is improved considerably, since frames are not completely lost. For example, for BasketballDrill and the highest level of background traffic (75 pps) and 16 tiles per frame, the TLR is improved more than 70% for AI (Fig. 8a) and nearly 30% for LPI4 (Fig. 8c). As a consequence, the objective video quality of the received video is much better. Specifically, the PSNR improvement reaches nearly 40% for AI (Fig. 8b) and more than 25% for LPI4 (Fig. 8d). With these improvements, the absolute values for the PSNR are over the acceptable threshold up to 50 pps when using 6 tiles per frame. Regarding to RaceHorses, the TLR improvements achieved more than 130% for AI and nearly 70% for LPI4 (not shown). For this video sequence and the highest level of background traffic (75 pps), the AI mode achieves acceptable PSNR values from 4 tiles per frame onward, whereas the LPI4 mode, despite the improvements achieved, only reaches the acceptable threshold for low background traffic load.

Despite the fact of the improvement achieved by using tile partitioning, it is not enough for moderate to high background traffic levels. In order to achieve more acceptable video quality levels, the use of QoS is explored next. The experiments were repeated prioritizing a proportion of the video packets, from 0% (without QoS) to 100%, either those corresponding to I frames only (“I”), or all of them regardless of the frame type (“IPB”). Table VI summarizes the obtained PSNR values for the BasketballDrill video sequence for both AI and LPI4 coding modes. Only the extreme cases are shown, that is, when P=0% (“No QoS”), and P=100% (“I” or “IPB”). Each table combines all the background traffic loads (×5) with all the tile layouts used (×7). In order to make the interpretation of these tables easier, a 4-color gradient scale is used as the background color of each cell. Taking into account that 36 dB is the PSNR value for the original video sequence, four ranges are defined: above 32 dB it is considered a very good

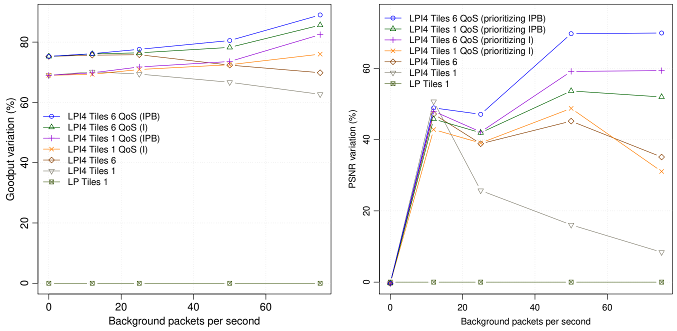


(a) PDR (Δ) - 1 tile/frame (b) PDR (Δ) - 6 tiles/frame

Fig. 9. Background traffic evaluation: BasketballDrill - AI mode



(a) Goodput (Δ) (b) PSNR (Δ)



(c) Goodput (Δ) (d) PSNR (Δ)

Fig. 10. QoS evaluation - Variation for AI (top) and LPI4 (bottom)

value (shown in green), acceptable above the 28 dB threshold (yellow), low above 24 dB (orange), and very low for values below that value (red). The first thing to be highlighted is that the use of QoS has almost no effect when the background traffic level is low, and its effect is greater when the network is more saturated. It is clearly visible that both coding modes improve the quality of the received video when QoS is used, specially when all the video packets are prioritized ("IPB"). In addition, it can be observed that a better video quality is obtained by using tile partitioning, but above a certain threshold the improvement obtained is not significant, if any. Therefore, in order to not unnecessarily increase the bitrate, the use of an intermediate value such as 6 tiles per frame may be reasonable. For example, when the background traffic load is high (75 pps), the AI coding mode experiences an approximate increase of 4 to 6 dB when using QoS. Regarding to LPI4, the increment ranges from 3 to 4 dB for the "I" experiment, and up to 8 dB for the "IPB" one.

Assigning more priority to video packets will make their access to the medium easier to the detriment of other non-priority background traffic. This effect is quantified with the BasketballDrill video sequence and the AI coding mode. The variation of PDR for both kind of traffic (video and background) with different percentage of protected video packets (25%, 50%, 75%, 100%) is shown in Fig. 9. Again, it is clear that the use of QoS has almost no effect when the background traffic level is low; otherwise, the greater the number of video packets are prioritized, the greater the number that reaches their destination, whereas the less background traffic is received. For example, when no tile partitioning is used and P=100% (all video packets are prioritized), video packets received experiences a 51.7% rise, whereas background traffic suffers a fall of approximately 24.4% only. As can be seen, other proportion of prioritized packets are intermediate cases. Therefore, a better use of the wireless channel takes place without damaging the background traffic too much (especially at low and moderated network loads). This is due to the shorter waiting times necessary to access to the channel that AC Video has (see Table I). Results for RaceHorses show a similar trend but, due to its high bitrate, the improvements are more noticeable for low network loads. Furthermore, when combining the previous experiments with the use of tile partitioning, results show a similar trend, but with a greater improvement due to the greater bitrate. In particular, Fig. 9b shows the case when using 6 tiles per frame; as can be seen, received video packets experiment a 61.4% rise, whereas background traffic suffers a fall of 35.0% at the highest traffic load (close to saturation).

Fig. 10 shows a global review for BasketballDrill for all the techniques to determine the benefits they provide. The graphs correspond to the goodput and PSNR variation for both the AI and LPI4 coding modes using as reference the LP coding mode with 1 tiles per frame and no QoS (P=0%). The goodput variation is high when intra-refresh is used for both coding modes, about 260% and 70%, respectively. This is due to the increased bitrate, specially for AI. Although this is not exploited when the network is unloaded, it allows to achieve clearly a higher quality when there is some background traffic. This means an increase in video quality (PSNR) of nearly 50% (about 11 dBs) for both coding modes at low background traffic loads (12 pps). Above this point, it can be seen that the use of intra-refresh alone is not enough to protect the video transmission as it is quite affected when higher values of background traffic exists. When using tile partitioning, in particular 6 tiles per frame, video traffic robustness improves significantly even for moderate background traffic values, at the cost of a small increase in bitrate. The use of any kind of QoS further improves the results, specially when the background traffic is moderate to high, whether in combination with the use of tile partitioning (6 tiles) or not (1 tile only). The goodput improvement is present when using any kind of QoS, either prioritizing video packets within I frames only or all of them regardless the type of frame. When using 6 tiles per frame and for the maximum background traffic load (75 pps), the achieved PSNR increment is more than 63% for AI (11 dB), and almost 70% for LPI4 (12 dB).

VI. CONCLUSION AND FUTURE WORK

Several experiments were carried out combining various techniques in order to improve the quality of the video transmitted in vehicular networks, such as the use of intra-refresh, tile partitioning, and QoS. Regarding intra-refresh, several coding modes with different proportion of I frames were compared with LP, concluding that this is essential in order to mitigate the temporal error propagation, as in the AI and LPI4 coding modes. Regarding tile partitioning, the use of a greater number of tiles per frame increases the robustness for video transmission and the quality of the reconstructed video, being a value of 6 the one that achieves the best trade off between video quality and bitrate increase. The above mentioned techniques were combined with the use of QoS by prioritizing the video packets at the MAC level according to the type of frame to which they belong. The results of the experiments showed that protecting all video packets is the best approach to achieve the highest video quality in all cases. Also, the non protected background traffic slightly reduces their network resources at moderated/high network loads, avoiding starvation from high priority traffic. We can conclude that the use of AI or LPI4 modes, with 6 tiles per frame, and prioritizing all the video frames is the alternative that provides the best results.

As future work, we are planning to combine the above experiments with other error protection techniques such as Forward Error-Correction (FEC). An adaptive scheme could also be designed that takes into account the saturation level of the network, or the size of the different MAC queues.

ACKNOWLEDGMENT

This work has been supported by the Spanish Ministry of Ciencia, Innovación y Universidades under Grant RTI2018-098156-B-C54, co-financed by FEDER funds.

REFERENCES

[1] A. Tufail, S. Ali Khayam, M. Taqi Raza, A. Ali, and K.-H. Kim, "An Enhanced Backbone-Assisted Reliable Framework for Wireless Sensor Networks," *Sensors (Basel, Switzerland)*, vol. 10, pp. 1619–51, 03 2010.

[2] "High Efficiency Video Coding (HEVC)," ITU-T Recommendation H.265, 2013.

[3] "Advanced Video Coding (AVC) for generic audiovisual services," ITU-T Recommendation H.264, 2003.

[4] D. B. Rawat, D. C. Popescu, and M. Song, "Performance enhancement of EDCA access mechanism of IEEE 802.11e wireless LAN," in *2008 IEEE Radio and Wireless Symposium*, Jan 2008, pp. 507–510.

[5] H. Zhu, M. Li, I. Chlamtac, and B. Prabhakaran, "A survey of quality of service in IEEE 802.11 networks," *IEEE Wireless Communications*, vol. 11, no. 4, pp. 6–14, Aug 2004.

[6] "IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2016*, pp. 1–3534, Dec 2016.

[7] "IEEE Standards for Information Technology – IEEE Std 802.11e – Draft Supplement to Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Quality of Service Enhancements," online, 2005.

[8] "IEEE Standard for Wireless Access in Vehicular Environments (WAVE) – Multi-Channel Operation," *IEEE Std 1609.4-2016 (Revision of IEEE Std 1609.4-2010)*, pp. 1–94, March 2016.

[9] "IEEE Standard for Information technology – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments," *IEEE Std 802.11p-2010*, pp. 1–51, July 2010.

[10] A. Ksentini, M. Naimi, and A. Gueroui, "Toward an improvement of H.264 video transmission over IEEE 802.11e through a cross-layer architecture," *IEEE Communications Magazine*, vol. 44, no. 1, pp. 107–114, Jan 2006.

[11] R. MacKenzie, D. Hands, and T. O'Farrell, "QoS of Video Delivered over 802.11e WLANs," in *2009 IEEE International Conference on Communications*, June 2009, pp. 1–5.

[12] I. Ali, M. Fleury, S. Moiron, and M. Ghanbari, "Enhanced prioritization for video streaming over wireless home networks with IEEE 802.11e," in *2011 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, June 2011, pp. 1–6.

[13] Zheng Wan, N. Xiong, N. Ghani, Min Peng, A. V. Vasilakos, and Liang Zhou, "Adaptive scheduling for wireless video transmission in high-speed networks," in *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2011, pp. 180–185.

[14] N. Chilamkurti, S. Zeadally, R. Soni, and G. Giambene, "Wireless multimedia delivery over 802.11e with cross-layer optimization techniques," *Multimedia Tools and Applications*, vol. 47, no. 1, pp. 189–205, Mar 2010. [Online]. Available: https://doi.org/10.1007/s11042-009-0413-6

[15] C.-H. Lin, C.-K. Shieh, C.-H. Ke, N. K. Chilamkurti, and S. Zeadally, "An adaptive cross-layer mapping algorithm for MPEG-4 video transmission over IEEE 802.11e WLAN," *Telecommunication Systems*, vol. 42, no. 3, p. 223, Jul 2009. [Online]. Available: https://doi.org/10.1007/s11235-009-9182-9

[16] X.-W. Yao, W.-L. Wang, S.-H. Yang, Y.-F. Cen, X.-M. Yao, and T.-Q. Pan, "IPB-frame Adaptive Mapping Mechanism for Video Transmission over IEEE 802.11e WLANs," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 5–12, Apr. 2014. [Online]. Available: http://doi.acm.org/10.1145/2602204.2602206

[17] M. M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[18] P. P. Garrido Abenza, P. Piñol Peral, M. P. Malumbres, and O. López-Granado, "Simulation framework for evaluating video delivery services over vehicular networks," in *IEEE 88th Vehicular Technology Conference (VTC-Fall)*, August 2018, pp. 1–5.

[19] 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.

[20] 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.

[21] 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.

[22] Joint Collaborative Team on Video Coding (JCT-VC), "HEVC reference software HM (HEVC Test Model) and Common Test Conditions," Available online: https://hevc.hhi.fraunhofer.de (accessed on 20 August 2018), Joint Collaborative Team on Video Coding (JCT-VC).

[23] P. P. G. Abenza, M. P. Malumbres, P. Piñol, and O. López-Granado, "Source Coding Options to Improve HEVC Video Streaming in Vehicular Networks," *Sensors*, vol. 18, no. 9, 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/9/3107