

Goal-Oriented Source Coding and Filtering for Vehicular Communications

Jose Manuel Gimenez-Guzman¹, Israel Leyva-Mayorga², *Member, IEEE*, and Petar Popovski³, *Fellow, IEEE*

Abstract—Vehicle-to-everything (V2X) networks will constitute a prominent application in future generations of cellular networks, definitely transforming our conception of transportation systems. A major challenge in V2X networks is the vast amount of data generated by the large number of sensors in the vehicles, which saturates the wireless links. As it is not possible to meet the throughput, timing, and reliability requirements for the total bulk of generated data, one needs to filter out data based on the actual communication goal. In this article, we present an architecture and diverse options to implement filtering and source coding for goal-oriented (GO) vehicular communications. We illustrate how filtering and source coding contribute to meeting the strict delay requirements while maintaining energy-efficient operation. Our results show that GO communications, performed as the combination of Bloom filtering and GO source coding, can greatly contribute not only to reduce the energy consumption by up to 30% but also to decrease delay, which in turn increases the supported amount of delay-sensitive traffic by up to 819.2%.

Index Terms—Bloom filter (BF), discontinuous reception (DRX), goal-oriented (GO) communications, intelligent transportation systems (ITSs), source coding, vehicle-to-everything (V2X).

I. INTRODUCTION

VEHICLE-TO-EVERYTHING (V2X) networks will drastically change the way we understand transportation systems, eliciting the introduction of reliable autonomous driving systems, holographic systems, or immersive infotainment services. To do so, V2X networks will agglutinate different modes for communication, including vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-pedestrians (V2P), vehicle-to-cloud networks (V2N), vehicle-to-roadside barriers (V2B), or vehicle-to-vulnerable road users (V2VRUs).

Manuscript received 30 November 2023; revised 31 January 2024; accepted 15 February 2024. Date of publication 19 February 2024; date of current version 23 May 2024. This work was supported in part by the MCIN/AEI/10.13039/501100011033 and the European Union “NextGenerationEU”/PRTR under Project TED2021-131387B-I00; in part by the MCIN/AEI/10.13039/501100011033/FEDER, UE under Project PID2021-123168NB-I00; in part by the Spanish Ministry of Science and Innovation under Project PID2019-104855RB-I00/AEI/10.13039/501100011033; in part by the SNS JU Project 6G-GOALS through the EU’s Horizon Program under Grant 101139232; and in part by the Villum Investigator Grant “WATER” from the Velux Foundation, Denmark. Funding for open access charge: CRUE-Universitat Politècnica de València. (Corresponding author: Jose Manuel Gimenez-Guzman.)

Jose Manuel Gimenez-Guzman is with the Department of Communications, Universitat Politècnica de València, 46022 Valencia, Spain (e-mail: jmgimenez@upv.es).

Israel Leyva-Mayorga and Petar Popovski are with the Department of Electronic Systems, Aalborg University, 9220 Aalborg, Denmark (e-mail: ilm@es.aau.dk; petarp@es.aau.dk).

Digital Object Identifier 10.1109/JIOT.2024.3367438

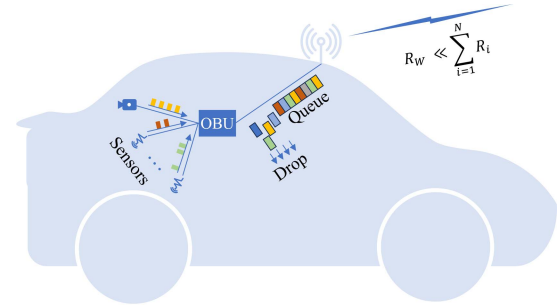


Fig. 1. Problem statement in V2X networks.

To support the plethora of advanced applications that will build up the future intelligent transportation systems (ITSs) we must design future wireless networks to support ultra reliable and low-latency communications (URLLCs). Although offering services with very low delays is mandatory for the successful deployment of ITS, achieving energy-efficient V2X networks is, for the sake of sustainability, a research problem of utmost importance. In fact, energy consumption is already one of the major concerns in 5G and one of the key performance indicators (KPIs) in the context of 6G networks. However, to fulfill these highly strict requirements over a shared and limited capacity spectrum constitutes a great challenge for the scientific community. In addition, this situation is even more challenging as a consequence of the spectrum regulation proposed by Federal Communications Commission (FCC) in [1]. In this regulation, the 5.9-GHz frequency band was splitted in two parts. While originally the 75-MHz bandwidth of the frequency band was fully assigned to ITS, the new regulation authorized the lower 45 MHz (from 5.850 to 5.895 GHz) for unlicensed use, leaving only 30 MHz to be used for V2X communications, even when the regulation admits that ITS proponents preferred to allocate the whole frequency band to ITS. Following this bandwidth reduction, the challenge of offering URLLC services is even greater and can become the bottleneck of future ITS deployments. Therefore, the problem, illustrated in Fig. 1, is evident: each vehicle has a large number of sensors N that continuously generate information with rate R_i , where $i \in \{1, 2, \dots, N\}$, that is gathered by the on-board unit (OBU) and needs to be transmitted through the V2X access network but the available capacity R_W is not sufficient and, therefore, some random packets are dropped due to congestion. In this setup, simply reducing the sampling frequency of the sensors is

not a viable option as it would negatively affect the reaction time of the network. Therefore, to address (or at least to alleviate) the problem of bandwidth shortage and, at the same time, increase network performance, we must act in many simultaneous fronts. For example, some proposals intend to enhance the spectrum efficiency [2]. In fact, 3GPP has evolved 5G NR through different Releases, making it more suited to support ITS. Unfortunately, 5G NR-based V2X networks will not be able to fulfill the strict requirements of future V2X services [3], empowering the design of 6G.

In this work, we address the problem of having vehicles that generate huge amounts of data and limited wireless communication resources from a different point of view. In a nutshell, we study how traditional and goal-oriented (GO) source coding can contribute to the feasibility of future ITS, operating either in 5G NR or 6G networks. Namely, source coding reduces the amount of information to be transmitted in the network, so the traffic congestion in the wireless link is alleviated, which reduces communication delay. However, source coding can be computationally intensive, so it incurs in an extra delay due to processing. This raises the question of which processing and compressing strategies are adequate for delay-constrained systems [4]. Accordingly, we will consider two types of complementary source coding algorithms: 1) lossy and 2) lossless. While the loss of information is tolerated in lossy source coding, lossless source coding ensures all the information can be recovered at destination by means of source decoding.

To understand the purpose of GO communications it is convenient to put them in context. The original ideas of GO communications were already defined by Shannon and Weaver in their seminal work [5], when laying the foundations of communications theory. Shannon and Weaver stated that the broad area of communication theory could be structured into three levels or problems. Undoubtedly, the most thoroughly studied problem so far is the first level, known as the technical problem, and whose purpose is to reliably transmit data or, more accurately, to study the accuracy for transmitting symbols. Notwithstanding, with the recent advent of machine/deep learning (ML/DL) techniques, we dispose of the tools required to bring the second and third level into fruition. In the second level, called the semantic problem, the transmitter aims to convey the semantic meaning of data [6] or, in other words, it consists of studying how accurately the transmitted symbols convey the meaning of the original data. However, Shannon and Weaver also envisioned a third level, the so-called effectiveness problem, which emphasizes the action that is expected from the receiver after the communication, laying the foundations of GO communications. Thus, while semantic communications is focused on correctly interpreting, at the destination, the concept associated to a message sent by the source, GO communications focuses on the correct accomplishment of the goal within a given time constraint and using a given amount of resources, such as energy [7].

In GO communications, there is an application (i.e., a high-level entity) that is responsible for fulfilling a specific task, which typically involves some form of response or actuation. Thus, in GO communications, the data source only needs to

transmit the data required to fulfill the specific task effectively and the rest of the data can be discarded. More specifically, in the context of GO communications we propose the use of two complementary techniques: 1) filtering and 2) source coding. First, we consider that some of the messages generated by the sensors are not necessary to be transmitted, so it is possible to filter them out. We have chosen Bloom filters (BFs) as the method for filtering messages and have considered that the filtering rules that drive BF are installed as the result of an artificial intelligence (AI) procedure guided by ML/DL techniques. In fact, there is a consensus in the research community that ML/DL and AI will play a key role [7] in GO communications and 6G networks, as these provide the mechanisms to tailor the selection of the important features of the generated data to the goal of the underlying application [8]. Note that the GO approach is very different from simply using a lower sampling rate for the sensors, which would decrease the reaction time to important events and can lead to the loss of crucial safety information if the sampling rate is too low. Moreover, GO filtering is an enabler for considering the information from different sensors in a unified view, so, for example, when the information of two sensors have the same meaning for the receiver, it is only necessary to transmit one of them. Second, we consider the possibility of applying GO source coding to the filtered flow of messages, as, for each message, we only need to transmit the data necessary for the destination to perform the right inference.

In this work, we investigate diverse techniques to reduce the amount of transmitted data in V2X networks, along with the tradeoffs between performance and energy-efficiency from a GO perspective [9]. There is a vast literature on ML/DL and Big Data algorithms for ITS, enabling a wide variety of applications, such as safety or incident detection and prediction, among many others [10], [11]. Examples of such algorithms include convolutional neural networks (CNNs), support vector machines (SVMs), random forests (RFs), or k -nearest neighbors (k -NNs). The choice of an algorithm will highly depend on the application and the source of data. For example, using CNNs and SVMs for image recognition with data coming from a camera is a common practice. Conversely, SVMs, RF, and k -NN are commonly used for selecting critical features from massive data [12]. Moreover, there are ongoing efforts on reducing the computational cost of ML/DL algorithms. For example, a reduction of computational complexity for neural networks can be found in [13], while for SVM and k -NN in [14].

In this article, we formulate a general framework with the objective of studying to which extent ML/DL methods can be useful to enable GO communications in V2X networks depending on their complexity. Thus, instead of following the general trend of selecting a subset of specific ML/DL algorithms for ITS, we formulate a model to characterize these algorithms based on their computational complexity and summarization capacity and evaluate their applicability on ITS systems. This work can also contribute to evaluating the impact of such a complexity reduction when ML/DL techniques are used for GO communications.

The main contributions of this work are as follows.

- 1) We present a hybrid model for the performance evaluation of V2X communications where analytical models are supplied with simulation data when no closed-form expression can be obtained.
- 2) We present a V2X communications architecture that is compliant with the European Telecommunication Standards Institute (ETSI) ITS architecture.
- 3) We study the benefits of traditional compression of the data locally at the vehicles.
- 4) We propose the use of BF to effectively perform lossy GO source coding in the context of ITS.
- 5) We show how ML and DL-based algorithms may contribute to GO communications to make future V2X networks feasible. These algorithms are characterized by their computational complexity and, due to the many possibilities for training, we consider only their inference phase.

The remainder of this article follows the next structure. In Section II, we present the main works related to the studied topic, while the V2X network traffic model proposed is thoroughly described in Section III. In Section IV, we present the system communication model, while in Section V we study how source coding and GO communications can contribute to the feasibility of future ITS. In Section VI, we propose how our proposals could be integrated in ETSI V2X architecture. Finally, in Section VII, we present the performance evaluation of the proposals made in the manuscript, and Section VIII summarizes the main conclusions of the work.

II. RELATED WORK

The design of future ITS with strict service requirements is a hot topic for the scientific community that involves a number of research fields [15] of major interest for both the academia and private companies. There are two main wireless technologies that are candidates to evolve for providing such requirements in V2X communications: 1) dedicated short-range communications (DSRCs) and 2) cellular V2X (C-V2X). Initially, DSRC was the only technology suited for vehicular communications. However, due to its unbounded channel access latency and limited coverage, 3GPP leveraged cellular technologies to develop C-V2X [16]. In fact, 3GPP introduced the possibility of direct communications between terminal equipments in Release 14 in 2017, with a clear focus in vehicular communications. Another key milestone regarding the suitability of these two wireless technologies happened at 2021, when US FCC changed its decision on the use of the reserved spectrum for ITS (the 5.9 GHz frequency band) from DSRC to C-V2X [1].

The scarcity of wireless resources to be used by V2X networks has elicited a number of works that propose data compression to alleviate congestion. Su et al. [17] proposed a discrete cosine transform (DCT) lossy compression technique to improve the throughput of the fronthaul vehicular 5G network. Guo et al. [18] made a survey of compressed sensing in vehicular infotainment systems, where compressed sensing changes the traditional vision of sampling and coding into a single step. Another interesting work related to ours

is [19], where authors study the compression gains that can be obtained with lossless compressions for different types of V2X messages, being cooperative awareness, collective perception, and maneuver coordination messages (see Section III-A for a better description of these messages). However, none of these works is focused on evaluating the energy consumption required to perform the lossless or lossy compression, which may be nonnegligible. Moreover, although not in the context of ITS but in Internet of Things (IoT), it is worth citing [20], where authors propose a framework that uses both lossless and lossy compression and compares both, concluding that lossy compression consumes less energy than its lossless counterpart, with the disadvantage of information loss.

The role of task and goal-oriented and semantic communications in 6G networks is expected to be one of the enablers of a plethora of new services to be offered [21], [22]. In the same works, authors also present ITS as one of the key use cases in the context of semantic communications, stating that DL-based semantic communications may help to compress and extract semantic information and thus reduce delay. Some works that use DL to transmit images and text semantically are [23] and [24], respectively, although these two works are not specifically designed for ITS. However, other works that propose DL semantic communications in the context of ITS are [25], [26]. Zhu et al. [25] proposed a semantic resource allocation procedure to transmit video, being this procedure based on a multiagent deep Q -network. Additionally, an image coding mechanism that preserves the semantic content has been proposed in [27]. On the other hand, Yang et al. [26] disclosed that the most straightforward application of semantic communications in the ITS domain is to extract semantic information from the whole bunch of sensors in vehicles, such as vehicle kinematic information, road conditions, or traffic signs. Also in the context of ITS, a cooperative semantic-aware architecture for multiuser communications in V2X networks that is able to reduce data traffic significantly has been proposed in [28]. Naturally, the suitability of goal-oriented and semantic communications is not restricted to 6G and ITS, as it has been also applied to the system design and resource optimization in general IoT networks [29].

With respect to the energy-efficiency perspective, [30] already identifies, in the context of connected autonomous vehicles, energy efficiency among the main features of 6G, in addition to agility, reliability, and ultralow delay. Later, Hussein et al. [15] showed that the efficiency in V2X networks will be crucial because of the growing number of connected nodes, the high communication and computation demands and the increasing energy in the adoption of new frequency bands in 6G, as all these issues will highly increase the energy demand. Moreover, authors also show that the strict quality of service requirements together with the AI procedures based on big data will create new challenges in dealing with energy efficiency improvements. A key work to understand the importance and scope of energy-efficiency in future vehicular communications is [31], where authors thoroughly present the main considerations about green V2X networks from five scenarios: 1) communication; 2) computation; 3) traffic management; 4) energy management for

electric vehicles; and 5) energy harvesting. More specifically, to develop energy-efficient V2X networks, Huang et al. [32] have considered the integration of mobile edge computing into V2X, extending the computing capability to the V2X edge in close proximity of connected vehicles. On the other hand, some works have proposed to increase V2X networks energy efficiency by optimizing interface selection [33], power allocation [34], [35], [36], or resource allocation [37]. Finally, and although not focused on vehicular environments but very related to our work, GO communications have been combined with the information bottleneck to send only the information relevant to perform an inference task, ultimately minimizing energy consumption [38].

III. DATA TRAFFIC AND ENERGY-EFFICIENT OPERATION IN C-V2X

In this section, we describe the types of messages that are transmitted in C-V2X together with a theoretical model for its temporal characterization. In addition, we present the main features of the discontinuous reception (DRX) mechanism present in 5G NR, intended to reduce energy consumption at the user terminals, along with the impact of the mechanism in the data traffic profile.

A. Traffic Model

Although there are other standardization initiatives in the field of V2X networks like SAE International (in USA) or CSAE (in China), we focus our attention on the initiative by the ETSI for V2X, which defines four types of V2X messages: 1) cooperative awareness message (CAM); 2) collective perception message (CPM); 3) decentralized environmental message (DENM); and 4) MCM. First, CAM messages contain information about location and the kinematic state of the transmitter. However, as not every object has V2X communication capabilities (for example, pedestrians, obstacles, animals, or non-V2X vehicles) or even having it they can be out of a certain vehicle range, ETSI has defined the Collective Perception Service to complement the Context Awareness Service [39]. CPM messages enable to share information about other road users or obstacles, so that the vehicles can extend their awareness even further than their sensing capabilities. Third, DENM messages contain information related to an abnormal traffic condition or a road hazard, including its type and localization. Finally, MCM messages are used to coordinate maneuvers between vehicles.

Inspired by [40], proposed by 3GPP to evaluate V2X in NR, we consider the following traffic model. For the uplink, we assume that vehicles send packets (being each packet the aggregation of a number of V2X messages) following two independent renewal processes $\{\mathbf{A}_p^{ul}(t), t \geq 0\}$ and $\{\mathbf{A}_{ap}^{ul}(t), t \geq 0\}$, whose interarrival times are defined by $\mathbf{X}_p^{ul} = (X_{1,p}^{ul}, X_{2,p}^{ul}, \dots)$ and $\mathbf{X}_{ap}^{ul} = (X_{1,ap}^{ul}, X_{2,ap}^{ul}, \dots)$, respectively. On the one hand, renewal process $\mathbf{A}_p^{ul}(t)$ represents the information that is sent periodically by the vehicle, such as position, speed, and vehicle type. With respect to the amount of information (in bytes) that is generated to be transmitted at each arrival ($D_{i,p}^{ul}$), it can be described by a random uniform

distribution $\mathcal{U}_{[D_p^{\min}, D_p^{\max}]}$ with a quantization step of $Q_{ul,p}$ bytes, averaging $(D_p^{\max} + D_p^{\min})/2$ in each arrival. On the other hand, renewal process $\mathbf{A}_{ap}^{ul}(t)$ represents event-triggered situations, such as a change in direction or a sudden braking. For the aperiodic traffic, we also make use of the high traffic intensity defined in [40]. Therefore, $X_{i,ap}^{ul} = C + \text{Exp}(\lambda'_{ul,ap})$, so the interarrival time is the sum of a constant value C and an exponentially distributed random variable with rate $\lambda'_{ul,ap}$. At each arrival, the amount of information ($D_{i,ap}^{ul}$) is defined by $\mathcal{U}_{[D_{ap}^{\min}, D_{ap}^{\max}]}$ with a quantization step of $Q_{ul,ap}$ bytes. It is important to note that the traffic model presented in [40] defines the periodic traffic and aperiodic traffic as two different options. However, in this work, we superpose both options to operate together as we assume that the information provided by car sensors is produced both periodically and event-triggered, so the uplink traffic is defined as the sum of both periodic and aperiodic renewal processes, with rates $\lambda_{ul,p}$ and $\lambda_{ul,ap}$, respectively. This assumption is based on the nature of the V2X messages, as some of the messages have a periodic nature (e.g., CAM) and other messages are event-driven (e.g., DENM) [16]. Moreover, to study the effect of different uplink traffic loads, we introduce ϱ as the factor by which we multiply $\lambda_{ul,p}$ and $\lambda_{ul,ap}$, so the uplink traffic packet rate is defined as $\lambda_{ul} = \varrho(\lambda_{ul,p} + \lambda_{ul,ap})$.

Assuming that the number of devices transmitting in a V2X environment is high, we can assume that the receiving process that defines the downlink reception is the superposition of a large number of independent random processes, so this superposition process converges to a Poisson process with rate λ_{dl} . Moreover, in each arrival, the amount of information will be defined by $\mathcal{U}_{[\min_{dl}, \max_{dl}]}$.

B. Discontinuous Reception Procedure

With the purpose of saving energy in 5G NR, in addition to the well-known from LTE RRC_CONNECTED and RRC_IDLE states, a user equipment (UE) can be in a new state called RRC_INACTIVE. Even though RRC_IDLE and RRC_INACTIVE states are highly energy-efficient, their impact on energy efficiency in V2X networks is expected to be minimal due to the huge amounts of data to be transmitted. Instead, the vehicles will stay most of the time in the RRC_CONNECTED state during operation, and so we focus on the RRC_CONNECTED state.

Monitoring the physical downlink control channel (PDCCH) is one of the most energy-consuming procedures for UEs in RRC_CONNECTED state, as it involves the search space blind decoding procedure. However, it is necessary to monitor the PDCCH to receive downlink packet indications. The main purpose of the DRX technique, inherited from 4G LTE, is to reduce the frequency of PDCCH monitoring. Next, we briefly describe DRX procedure but, for a more detailed description, refer to [41]. When equipped with DRX, a UE in RRC_CONNECTED state monitors the PDCCH periodically and only during a predefined ON period (T_{ON}), which is configured with the *drx-onDuration* field of the RRC message. After the ON period, if there is no uplink or downlink packet, the UE enters into an energy-saving state (DRX) turning off

its RF chain, a state where the energy consumption decreases. However, if while in T_{ON} the UE either receives a downlink packet from the gNB or an uplink data packet from the upper layer from its local protocol stack, that packet can be immediately queued for transmission. When the UE is in DRX state, it is important to note that uplink data, as received locally from upper layers, can *awake* UE from DRX and try to transmit that data immediately, but downlink data received from gNB will experience a delay until the next ON period starts. However, as energy consumption is becoming of paramount importance, manufacturers are currently implementing discontinuous transmission (DTX), so uplink transmissions are aligned also with the DRX cycles, with the consequence of further energy savings [42]. As in our work energy consumption plays a central role in network design, we consider both DRX and DTX, although for simplicity in notation we will shorten the notation from DRX/DTX to DRX. It is important to highlight that using DTX implies a relation between uplink or downlink traffic because any of them can awaken the UE, so both traffic flows cannot be studied in isolation.

As energy saving has become a crucial aspect in communications, 3GPP proposed in Release 16 an additional power-saving mechanism called power saving signal (PSS). The motivation for this proposal is that it is still energy-consuming to periodically enter into the ON period to monitor PDCCH, especially if the traffic is sporadic so that the probability of the ON period to finish without having received traffic is high. To the readers familiarized with WLAN technologies, it is worth noting that the central concept of PSS is similar to wake-up radio (WUR) in IEEE 802.11ba, as it is based on informing about when to turn on the main radio. In our case, PSS is a technique to inform the UE whether to start or not the next ON duration. However, IEEE 802.11ba WUR operates by including a ultralow power chip while 3GPP proposal is based on a low power-consuming indication (the so-called PSS). The operation of PSS is as follows [41]. 3GPP has defined a new control message, called downlink control information of power saving (DCP), to indicate if a UE can skip the next ON period and therefore, avoid the energy-consuming blind decoding procedure of monitoring PDCCH. It must be highlighted that DCP is also transmitted over the PDCCH. However, the UE does not perform blind decoding to receive DCP as it is scrambled with a specific power saving-radio network temporary identity (PS-RNTI).

IV. SYSTEM MODEL

In this section, we present the model for data transmission, analyzing its behavior in terms of delay and energy consumption. The components to be considered, due to having an impact on the total delay, are the transmission itself, the queuing of the packets before transmission, and the added delay due to the DRX procedure.

A. Transmission

We consider a V2X wireless environment where the channel conditions vary at two different time scales due to path loss,

large-scale, and small-scale fading. Specifically, the signal-to-noise ratio (SNR) varies with time due to the distance between vehicles d , resulting in changes in the free space path loss, and due to block Rayleigh fading. Consequently, we consider that both the distance between vehicles and the channel power gain $g[i]$ remain constant during the transmission of a given packet i but vary over time as a result of multipath fading and/or shadowing.

Since we consider a Rayleigh fading channel, $g[i]$ for each packet is an exponentially distributed random variable $g[i] \sim \text{Exp}(1)$. Let P_{tx} to be the transmission power over a channel with a bandwidth B . Then, the SNR γ for the i th packet at a receiver at a distance d from the transmitter can be computed as

$$\gamma[i] = \frac{K_0 P_{tx}}{d^\ell N_0 B} g[i] \quad (1)$$

where K_0 is the Friis equation parameter, N_0 is the noise spectral density, and ℓ is the path loss exponent. Since $(K_0 P_{tx}/d^\ell N_0 B)$ is a constant and we consider a Rayleigh channel, the SNR $\gamma[i]$ also follows an exponential distribution. Hence, the average SNR is

$$\bar{\gamma} = \frac{K_0 P_{tx}}{d^\ell N_0 B}. \quad (2)$$

So (1) can be rewritten as

$$\gamma[i] = \bar{\gamma} g[i]. \quad (3)$$

Outage occurs when the received γ is below a threshold γ_{thr} , when bits received cannot be correctly decoded. Then, $P_{out} \triangleq \Pr\{\gamma < \gamma_{thr}\}$. Using (3) and because $g[i]$ is an exponentially distributed random variable, P_{out} can be rewritten as

$$P_{out} = \Pr\left(g[i] < \frac{\gamma_{thr}}{\bar{\gamma}}\right) = 1 - e^{-\gamma_{thr}/\bar{\gamma}}. \quad (4)$$

As the outage probability is exponentially distributed (and also the probability that the received SNR is below a threshold), the number of transmissions for successfully sending a packet \mathcal{N} is geometrically distributed, being its probability mass function

$$\Pr(\mathcal{N} = n) = P_{out}^{n-1} (1 - P_{out}), \quad n \geq 1 \quad (5)$$

whose mean is $1/(1-P_{out})$. Note that we have assumed that the time to acknowledge the reception is negligible in comparison to the data transmission time. With that, the mean transmission time for packets whose average size is $\bar{\sigma}$ is given by

$$T^{trx} = \frac{1}{1 - P_{out}} \frac{\bar{\sigma}}{R}. \quad (6)$$

Finally, the throughput provided by the Rayleigh fading channel for a given SNR threshold γ_{thr} and target outage probability P_{out} can be computed by

$$R = (1 - P_{out}) B \log_2(1 + \gamma_{thr}) \quad (7)$$

that, using (4), can be rewritten as

$$R = (1 - P_{out}) B \log_2\left(1 + \bar{\gamma} \log\left(\frac{1}{1 - P_{out}}\right)\right). \quad (8)$$

On the other hand, the energy consumed by a UE to transmit a packet with length σ over a Rayleigh fading channel can be computed as

$$E_{\text{UE}}^{\text{trx}} = \frac{1}{1 - P_{\text{out}}} \frac{\sigma P_{\text{trx}}}{R}. \quad (9)$$

B. Queueing

Next, we present the model used to characterize the queueing delay experienced by the data in the network. This queueing delay depends on the arrival process and the service time process, which also define the system utilization. As both the arrival and service times follow complex distributions due to the nature of incoming traffic and the randomness of the wireless channel, for studying the queueing delay we model the wireless channel as a G/G/1 based on the Allen-Cunneen approximation [43], [44]. Differently from what we are used to in other simpler queues like M/M/1 or M/D/1 there is not a closed-form formula for the queueing delay in G/G/1, but an upper-bound. However, our traffic and service times are far from being exponential, so we cannot resort to simpler queues. Let ρ to be system utilization, \mathcal{A} the random variable that defines the interarrival times of transactions to the wireless link, and \mathcal{S} the random variable that defines the service times. The average queueing delay can be upper bounded by

$$T_q \leq E[\mathcal{S}] \frac{\rho}{1 - \rho} \frac{CV(\mathcal{A})^2 + CV(\mathcal{S})^2}{2} \quad (10)$$

where $CV(X)$ represents the coefficient of variation of random variable X , defined as the quotient between the standard deviation and the mean of random variable X , i.e., $CV(X) \triangleq \sigma_x/E[X]$. It is interesting to notice how (10) is simplified for exponential distributions, where $CV(X) = 1$. If both interarrival and service times are exponential we obtain the well-known mean waiting time for the M/M/1 queue.

From the channel model, we can compute $CV(\mathcal{S})$ because the service time for a successful data transmission is governed by a geometric distribution (5). Then, its mean time is represented by (6) and the variance, for packets with mean size $\bar{\sigma}$, by

$$\sigma_{\mathcal{S}}^2 = \frac{P_{\text{out}}}{(1 - P_{\text{out}})^2} \left(\frac{\bar{\sigma}}{R} \right)^2. \quad (11)$$

So, we can easily compute the coefficient of variation of the service time as $CV(\mathcal{S}) = \sqrt{P_{\text{out}}}$. On the other hand, due to the complex nature of incoming messages (see Section III-A), $CV(\mathcal{A})$ must be empirically computed, as it does not have a closed-form formula.

C. DRX Procedure

Let P_{PDCCH} be the power required by the UE to monitor the PDCCH for possible data transmission. If DRX is not implemented, the UE will consume P_{PDCCH} continuously. On the other hand, the energy consumption for a UE implementing the DRX mechanism is

$$E_{\text{UE}}^{\text{DRX}} = \left(P_{\text{PDCCH}}(1 - PSF) + P_{\text{sleep}}PSF \right) T \quad (12)$$

where P_{PDCCH} stands for the power required by UE to monitor PDCCH for possible data transmission, P_{sleep} is the power consumption of the UE when it is in the sleep state, PSF is the power saving factor, defined as the fraction of time where the UE is in the energy-saving state, and T is the total time where the energy consumption is being measured.

For computing the additional delay introduced to uplink messages due to the DRX procedure (T^{DRX}), we cannot use closed-form formulas due to the complexity of the procedure and the incoming traffic and also because downlink traffic also influences the DRX state. Therefore, T^{DRX} will be estimated using discrete-event simulations.

V. SOURCE CODING IN V2X NETWORKS

There is a unanimous consensus in research community [45] that V2X will generate huge amounts of data that will be a challenge to be dealt with. For that reason, it is convenient or even in some cases mandatory to resort to some kind of source coding to compress data and make it more manageable and, when necessary, to make it possible to be transmitted without overloading the communication channel. In fact, if we overload the wireless access network, the packet delivery ratio will increase and we can lose packets randomly, which can be fatal in case the most important messages are lost. Of course, reducing the data volume will also let us to decrease the energy consumption for transmitting data, but it will also incur into an extra delay and energy consumption for the source coding computing procedure. For that reason, we start by considering the use of traditional source coding techniques. In addition to traditional source coding, data traffic has some features that makes it feasible to use other type of GO lossy source coding. It is important to understand that data from different sensors or even from the same sensor can be redundant, specially in consecutive transmissions. Also, that this information has value in a specific and short period of time, so it is a delay-sensitive information, whose freshness for the receiver is important. In a nutshell, massive data arrives to the OBU from a number of sensors as the sum of multiple data streams and we must process it immediately or it is lost forever. We assume that it is not possible to store it because usually the storage in the OBU is limited and, specially, because the fast expiration of the value of data makes it useless to store it even when technically possible. Due to all these features, we consider the use of the techniques useful for mining data streams [46, Ch. 4]. To deal with reducing data streams produced by sensors from vehicles we propose the use of filtering, and more specifically, the use of BF, using ML/DL techniques to install the proper rules to filter all those messages whose meaning is not important to the receiver. Finally, and in addition to GO filtering, we explore the role of GO source coding in the context of ITS.

A. Traditional Source Coding

Traditional source coding methods can be categorized in two main types: 1) lossless and 2) lossy. The main idea of lossless source coding is to compress the data making use of its statistical redundancy to recover the information without

errors at the destination. Obviously, the efficiency of this procedure mostly depends on the redundancy of the data and the nonuniform repetition of sequences in the data. In fact, it is well-known from the pigeonhole principle that, provided a lossless algorithm compresses some input sequences, it will also expand others. Formally, a lossless source coder can be defined as a couple of functions C and D that take bit sequences as inputs and return also bit sequences, where, for any $x \in \mathcal{B}$, $D(C(x)) = x$, being x the data which we want to compress and \mathcal{B} an ordered finite set of sequences of bits. Note that this expression represents the fact that lossless source coders must be uniquely decodable.

We define the source coder gain for a source X of a statistical compressor as

$$G(X) = 1 - \frac{H(X)}{\log_2(K)} \quad (13)$$

being $H(X)$ the entropy of source X and K the alphabet size. Note that the entropy tells us about how surprised we will be on average by the outcome of a source. Stated more rigorously, entropy is defined by

$$H(X) \triangleq - \sum_{i=1}^K \Pr(X = s_i) \log_2(\Pr(X = s_i)) \quad (14)$$

where $\Pr(X = s_i)$ represents the probability of occurrence of symbol s_i .

Now we focus our attention on the computational aspects of lossless source encoding. The computational effort, expressed in CPU cycles per bit, to compress a single bit has a stochastic behavior due to the high number of running tasks in each CPU. In fact, the number of cycles to compress a bit \mathcal{W}_{LL} can be modeled by a Gamma distribution with parameters α_{LL} (shape) and β_{LL} (scale) [4], [47], whose PDF is

$$f_{\mathcal{W}_{LL}}(x; \alpha_{LL}, \beta_{LL}) = \frac{1}{\beta_{LL}^{\alpha_{LL}} \Gamma(\alpha_{LL})} x^{\alpha_{LL}-1} e^{-x/\beta_{LL}} \quad (15)$$

where $\Gamma(s) = \int_0^{\infty} t^{s-1} e^{-t} dt$ is the Gamma function. Then, $E[\mathcal{W}_{LL}] = \alpha_{LL} \cdot \beta_{LL}$. Parameters of the Gamma distribution are related to the mean ($E[\mathcal{W}_{LL}]$) and variance ($\sigma_{\mathcal{W}_{LL}}^2$) of the distribution by: $\alpha_{LL} = E[\mathcal{W}_{LL}]^2 / \sigma_{\mathcal{W}_{LL}}^2$ and $\beta_{LL} = \sigma_{\mathcal{W}_{LL}}^2 / E[\mathcal{W}_{LL}]$. Therefore, $E[\mathcal{W}_{LL}] = \alpha_{LL} \beta_{LL}$.

The average time required to compress a sample x with size σ with a lossless source coder can be computed by

$$T_{LL(x)} = \frac{\sigma \cdot \alpha_{LL} \cdot \beta_{LL}}{N_{\text{CPU}} \cdot f_{LL}(x)} \quad (16)$$

where N_{CPU} is the number of CPU cores, and $f_{LL}(x)$ is the clock frequency used by the CPU for the compression task of sample x .

On the other hand, there is a tradeoff between the compression gain that can be achieved using source coding and the computation time and energy required for such task [19]. To account for that tradeoff, we model the mean number of cycles required by the source coder to compress one bit of raw data as [20], [48]

$$E[\mathcal{W}_{LL}] = \alpha_{LL} \beta_{LL} = e^{\epsilon/(1-G(X))} - e^{\epsilon} \quad (17)$$

where ϵ is a positive value that depends on the compression algorithm.

Additionally, the energy consumption of the source coder is

$$E_{LL(x)} = \frac{\sigma P_{\text{proc}}(f_{\text{CPU}})}{f_{\text{CPU}}^3} \left(e^{\epsilon/(1-G(X))} - e^{\epsilon} \right) f_{LL}^2(x) \quad (18)$$

being $P_{\text{proc}}(f_{\text{CPU}})$ the power consumption of the CPU operating at its maximum clock frequency (f_{CPU}).

The second type of traditional source coders are lossy ones, that are able to reduce the amount of data admitting some loss of noncrucial information to preserve the best approximation of the raw data. One of the most well-known techniques to implement lossy source coders is by means of DCT [17], due to its strong energy compaction capability. Regarding the delay and energy models of lossy source coding it is the same as for its lossless counterpart, but the computational effort for lossy source coding is lower, and so the values of ϵ .

B. Goal-Oriented Coding

GO coding represents one of the crucial applications where ML/DL techniques can make future ITS feasible. As the GO techniques we are considering are lossy, there is a tradeoff between the probability of filtering out a critical message and the compression ratio, so we cannot ensure that a critical message is not lost, especially for very high compression ratios. Notwithstanding, the main purpose of the deployed AI techniques must be to understand the impact of every message so as to minimize the probability of losing critical information. As it has been aforementioned, we propose to deploy GO communications in the vehicular environment by the use of two complementary techniques: 1) filtering and 2) source coding. Note that GO communications are assumed to have an inherent summarization capacity. This is mainly because the concept focuses on only transmitting the information that is strictly relevant for the destination to effectively perform the right goal, assuming that the consumer of the data is an application at the end receiver, which interprets the data and, typically, generates a reaction to it. This is reflected in our approach where GO communications are used to summarize the huge amount of data arriving to the OBU as, otherwise, it cannot be transmitted due the limited amount of wireless resources. In fact, also in [7], it is shown that, due to the reduction of the entropy when performing GO coding, the number of bits when using GO communications can be significantly decreased.

For the computation tasks required by ML/DL techniques, we have considered that they are performed locally at the CPU in the OBU. Unfortunately, in the studied setting it is not possible to resort to edge or cloud computing as the huge amount of generated data in connected vehicles makes it infeasible to send it for being transmitted wirelessly over limited-bandwidth networks. However, there are some research efforts to perform local computations at the GPU instead of the CPU for the optimization of ML/DL techniques [49], so our work could be extended to GPU considering a performance and energy consumption model such as the ones proposed in [50] and [51].

1) *Bloom Filters*: In this section, we propose the use of BFs [52] to efficiently discard the individual messages within the packets that do not possess predefined characteristics. Note that the definition of these characteristics will be guided by ML/DL techniques to account for complex relations among messages and their possible interest for eliciting a response from the receivers. As BF can be considered a type of lossy source coders, it is important to distinguish BF from the above-mentioned DCT-based traditional proposals, as the way of reducing data volume of BF is completely different, because it is based on filtering some messages instead of a compression of all of them.

However, BFs are not the only data structure designed for checking an approximate membership that, in our case, results in discarding all those packets that do not have some features. Other filters that could also be used are Cuckoo filters [53] or the more recently proposed Xor filters [54]. Despite both BFs and Cuckoo filters have a zero false negative probability, we have considered BFs because the number of elements that can be inserted in a Cuckoo filter is limited and the insertion may fail when the occupation is high. BFs do not suffer from this problem since the insertion of elements is always possible and the only negative effect is a potential increase in the false positive rate [55], which is not critical and only results in a minor decrease in efficiency. On the other hand, we have not used Xor filters because their construction is twice slower than BFs and their performance is based on the computation of three hash functions [54] while in our case we will resort to BFs using only one hash function (as we show in Section VII-C).

A BF consists of an array of n bits to compactly represent a set of m key values, $\mathcal{K} = \{k_0, k_1, \dots, k_{m-1}\}$ and a collection of κ hash functions $\{h_1, h_2, \dots, h_\kappa\}$, so that each hash function maps values of $k_i \in \mathcal{K}$ to uniformly distributed integer number in the set $\{0, 1, \dots, n-1\}$. Furthermore, the required length of the output of the hash function is typically small, as it only requires $l \geq \log_2(n)$ bits. For example, to have a BF of length 8192 we will require that $l \geq 13$, while the shortest length in the hash functions we are considering is 32 bits. Finally, BF do not increase the complexity of the receiver, as there is no additional delay nor energy consumption due to a decoding stage.

BF are well suited for our desired task as, although with a low probability will let pass some undesired messages (false positive probability), a BF will never block a desired message (false negative probability is equal to zero). That property of BFs guarantees that none of the most crucial security-related V2X messages is lost.

The operation of the BF consists of two stages. The first one is for populating it with the information of the \mathcal{K} criteria that will let to pass the filter, as shown Algorithm 1. It is important to note that this stage will be only executed once or when the ML/DL entity wants to change the filtering criteria. Moreover, we can use different BF to manipulate the amount of data that is discarded depending on external factors like traffic congestion.

The second stage (Algorithm 2) is the operation of the filter, that consists of deciding if a new message x must pass

Algorithm 1: Populating the BF

```

Input:  $\mathcal{K} = \{k_0, k_2, \dots, k_{m-1}\}$ 
Output:  $B$ 
 $B(s) \leftarrow 0, \forall s \in \{0, \dots, n-1\}$ ;
for  $i \in \{0, \dots, m-1\}$  do
  for  $j \in \{1, \dots, \kappa\}$  do
     $z \leftarrow h_j(k_i) \bmod n$ ;
    if  $B(z) == 0$  then
       $B(z) \leftarrow 1$ ;
    end
  end
end
return  $B$ ;

```

Algorithm 2: BF Operation

```

Input:  $B, x$ 
Output: {Pass, Filter}
 $action \leftarrow \text{Pass}$ ;
 $i \leftarrow 1$ ;
while  $action == \text{Pass} \ \&\& \ i \leq \kappa$  do
   $z \leftarrow h_i(x) \bmod n$ ;
  if  $B(z) == 0$  then
     $action \leftarrow \text{Filter}$ ;
  end
   $i \leftarrow i + 1$ ;
end
return  $action$ ;

```

through the filter (if it meets one of the criteria) or must be discarded. Note that a message is discarded when the value of the BF ($B(z)$) in the position corresponding to the result of the hash function modulo n ($z = h_i(x) \bmod n$) equals to 0, as there is not any match criterium to let it pass through the filter. An example of criteria can be one specific type of ETSI V2X message or V2X messages that contains a specific container. It is important to note that the number of criteria m can be very large and that BFs avoids us to test them one by one. The computation cost of this second stage for a message x is bounded to a maximum of computing κ hash functions. Note that the only requisite for the hash function is that its possible outputs are uniformly distributed, a usual feature of the most common hash functions. In fact, as we require κ hash functions, they can even be computed with different techniques, such as CRC32, MD5, SHA-1, SHA-256, or SHA-512.

Finally, as it has been above-mentioned, Bloom filtering will not discard any desired important message but, with a certain probability, will let some messages that should be filtered. This false positive probability [52] will depend on the number of hash functions κ , the length of the BF n and the number of elements in the filter m

$$P_{fp} = \left(1 - \left(1 - \frac{1}{n}\right)^{\kappa m}\right)^\kappa \quad (19)$$

which can be approximated by

$$P_{fp} \approx \left(1 - e^{-\frac{\kappa m}{n}}\right)^\kappa. \quad (20)$$

Assuming that the computation cost of the operation of BF is mainly due to the computation of hash functions, and so

its time complexity is $O(\kappa)$, the delay introduced to test if a sample x with length σ bits is filtered or not by a BF that uses a collection of $\{h_1, h_2, \dots, h_\kappa\}$ hash functions is upper bounded by

$$T_{BF(x)} \leq \frac{\sigma}{N_{\text{CPU}}} \sum_{i=1}^{\kappa} \frac{E[\mathcal{W}_{h_i}]}{f_{h_i(x)}} \quad (21)$$

being N_{CPU} the number of CPUs, $E[\mathcal{W}_{h_i}]$ the average number of CPU cycles required for computing $h_i(x)$ for each bit and $f_{h_i(x)}$ is the frequency used by the processor to execute the computation of the i th hash function. The number of cycles to process a bit by the i th hash function \mathcal{W}_{h_i} can be modeled by a Gamma distribution with parameters α_{h_i} (shape) and β_{h_i} (scale) [47], whose PDF is

$$f_{\mathcal{W}}(x; \alpha_{h_i}, \beta_{h_i}) = \frac{1}{\beta_{h_i}^{\alpha_{h_i}} \Gamma(\alpha_{h_i})} x^{\alpha_{h_i}-1} e^{-x/\beta_{h_i}}. \quad (22)$$

Therefore, $E[\mathcal{W}_{h_i}] = \alpha_{h_i} \beta_{h_i}$ and

$$T_{BF(x)} \leq \frac{\sigma}{N_{\text{CPU}}} \sum_{i=1}^{\kappa} \frac{\alpha_{h_i} \beta_{h_i}}{f_{h_i(x)}}. \quad (23)$$

Space complexity of BF is not an issue, as the amount of memory required is equal to $n/(8 \cdot 2^{10})$ kbytes. On the other hand, the energy consumption to test if a sample x is filtered by the BF or not is

$$E_{BF(x)} \leq \frac{\sigma P_{\text{proc}}(f_{\text{CPU}})}{f_{\text{CPU}}^3} \sum_{i=1}^{\kappa} \alpha_{h_i} \beta_{h_i} f_{h_i(x)}^2 \quad (24)$$

where $P_{\text{proc}}(f_{\text{CPU}})$ is the power consumption of the CPU operating at its maximum clock frequency (f_{CPU}).

As a conclusion, the effect of using Bloom filtering in V2X is a reduced amount of traffic but at the cost of increasing the delay and energy consumption due to the computation required by the filter. For such an evaluation we introduce the Bloom filtering factor (BFF) of a certain BF defined as

$$\text{BFF} = 1 - \frac{E[\hat{\sigma}]}{E[\sigma]} \quad (25)$$

where $E[\hat{\sigma}]$ is the average length of packets after being filtered. Recall that the packets to be transmitted contain multiple ETSI V2X messages, so, from this definition, the BFF is a function of the number of ETSI V2X messages that are discarded within the packets. Although we can configure more strict BF to get higher values of BFF, the obtained BFF after filtering will highly depend on the degree of redundancy of data traffic.

2) *Goal-Oriented Source Coding*: In V2X, the goal of the communication is not usually to reconstruct the received messages as they were originally sent, but to let the receiver of a message to decide the right action after the meaning of a certain message. In other words, instead of transmitting a message, in V2X networks vehicles are interested in communicating the goal of such their messages so the destinations make the right inference. So GO communications can be considered a type of source coding where the objective is to infer the objective of each message.

To perform the GO source encoding we must resort to complex algorithms, mostly based on machine learning [56], [57] and, more recently, deep learning [58], [59] (from now on, GO-ML and GO-DL, respectively). And, of course, these algorithms are usually computation-intensive, so we incur in a nonnegligible and extra delay and energy consumption. For the sake of generality, as we do not focus on a specific GO-ML/DL algorithm but in a general one, we consider different usual algorithmic complexities in the area of ML/DL, being $O(\mathcal{F})$, $O(\mathcal{F} \cdot \log(\mathcal{F}))$ and $O(\mathcal{F}^2)$, with \mathcal{F} the number of features, that we assume to be proportional to the bulk data size (σ). For example, in the case of GO-DL, feature extraction is automatic, so it is expected that, as the size of bulk data increases (probably due to the result of coming from a higher number of vehicle sensors), so does the number of features. Then, complexity can be expressed in terms of σ . Then, for several algorithmic complexities we define an equivalent packet size σ_e , that depends on σ but also incorporates the complexity of the ML/DL algorithm. To account for the abovementioned GO-ML/DL algorithmic complexities, we define \mathbb{S} to be set of values of σ_e we are considering, being $\mathbb{S} = \{\sigma, \sigma \cdot \log(\sigma), \sigma^2\}$.

The choice of using the Big O notation to consider different computational complexities for ML/DL techniques is inspired by the field of computer vision, where the computational complexity has a dependence on the input size. In this research field, we can find techniques that use neural networks and vision transformers for image classification, object detection and semantic segmentation, which play a key role in semantic and GO communications. For example, Wan et al. [60] proposed SeaFormer to perform image semantic segmentation using vision transformers and show that its computational complexity is quadratic with the input size. Due to the increasing importance of developing computer vision techniques that operate in real-time in mobile edge nodes, there is an intense research activity in the area of lightweight techniques for reducing the computational complexity. For example, Dong et al. [61] proposed a semantic segmentation technique based on both CNNs and vision transformers to reduce the computational complexity from quadratic to linear complexity with the input size. Note that in our work we consider both quadratic and linear computational complexities, along with a loglinear complexity as an intermediate value between them.

Let $A(\sigma_e, \mathcal{W}_{GO})$ to be the representation of a task to be run by the GO-ML/DL, where σ_e is expressed in bits and \mathcal{W}_{GO} represents the computation workload/intensity for that task, expressed in CPU cycles per bit. These parameters, and in particular \mathcal{W}_{GO} , can be estimated using task profilers [62]. As \mathcal{W}_{GO} can be modeled by a Gamma distribution with parameters α_{GO} and β_{GO} , we have $E[\mathcal{W}_{GO}] = \alpha_{GO} \cdot \beta_{GO}$.

To compute the time that is required to finish a task A_i , we assume nonpreemptive CPU allocation so, once assigned, the CPU executes task A_i until it is completed. The execution time for completing a GO-ML/DL task A_i can be defined as

$$T_{A_i}^{\text{proc}} = \frac{\sigma_e \cdot \mathcal{W}_{GO}}{N_{\text{CPU}} \cdot f_{A_i}} = \frac{\sigma_e \cdot \alpha_{GO} \cdot \beta_{GO}}{N_{\text{CPU}} \cdot f_{A_i}} \quad (26)$$

where N_{CPU} is the number of cores of the CPU. Note also that it is considered that the CPU cores do not necessarily work at their highest clock frequency f_{CPU} , but we consider that they can adapt its CPU clock frequency for each task (f_{A_i}) through dynamic voltage and frequency scaling (DVFS) technique, that permits to step down (or up) the CPU cycle frequency or voltage, to decrease (increase) energy consumption while increasing (decreasing) the processing time.

After defining the execution time of GO source encoding tasks, we now model the energy consumption of such executions. For such an issue, we make use of a model that captures the main features of CPU chipsets [63], [64], [65], [66]. CPU energy consumption is due to factors, such as short-circuit, leakage, and dynamic energy consumption, predominating the effect of this last one against the others, so we restrict our energy analysis to dynamic energy consumption. More specifically, Burd and Brodersen [63] showed that dynamic energy consumption of a CPU cycle is proportional to V_{DD}^2 , being V_{DD} the circuit supplied voltage. However, the voltage is approximately linear to the operating frequency of the CPU, so we can compute the energy consumption per clock cycle when task A_i is being processed as

$$E_{\text{cycle}}^{\text{proc}}(f_{A_i}) = C_{\text{eff}} \cdot f_{A_i}^2 = \frac{P_{\text{proc}}(f_{\text{CPU}})}{f_{\text{CPU}}^3} f_{A_i}^2 \quad (27)$$

being C_{eff} the effective switched capacitance coefficient, that depends on the chip architecture (see [66] for a number of different values of C_{eff} proposed in the literature) and $P_{\text{proc}}(f_{\text{CPU}})$ the power consumption at maximum CPU clock frequency (f_{CPU}).

As a consequence, the energy consumption to complete a GO task A_i can be easily determined by

$$E_{A_i}^{\text{proc}} = \sigma_e \cdot \mathcal{W}_{GO} \cdot E_{\text{cycle}}^{\text{proc}}(f_{A_i}) = \sigma_e \cdot \alpha_{GO} \cdot \beta_{GO} \cdot E_{\text{cycle}}^{\text{proc}}(f_{A_i}). \quad (28)$$

VI. PROPOSED ARCHITECTURE

In this section, we show how source coding can be deployed in future V2X deployments. In the proposed architecture, GO communications are performed by using both BF and GO source coding. Both techniques are complementary and serve for the same purpose, but operate differently. On the one hand, BF is focused on filtering out complete messages that are not necessary for the destination to effectively fulfil the expected goal. Note that to contribute to an effective GO communication, filtering rules of BF are developed as the result of an AI procedure. On the other hand, GO source coding is centered on reducing the size of the data to the minimum needed for the destination to perform the right goal. From that, using both BF and GO source coding together will be able to achieve more effective GO communications.

In Fig. 2, we show a simplified view of how the different complementary techniques can be put together to contribute to fulfill the different strict requirements that ITS will require from future V2X networks. Note that inside the block ‘‘Source encoding’’ we can implement one or more of the source coders explained in Section V. In the case that are all present, we propose to perform Bloom filtering in the first step followed by

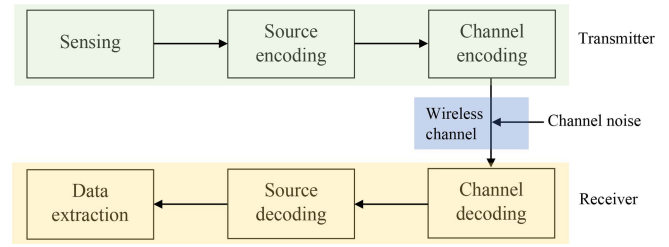


Fig. 2. V2X communication model including source coding.

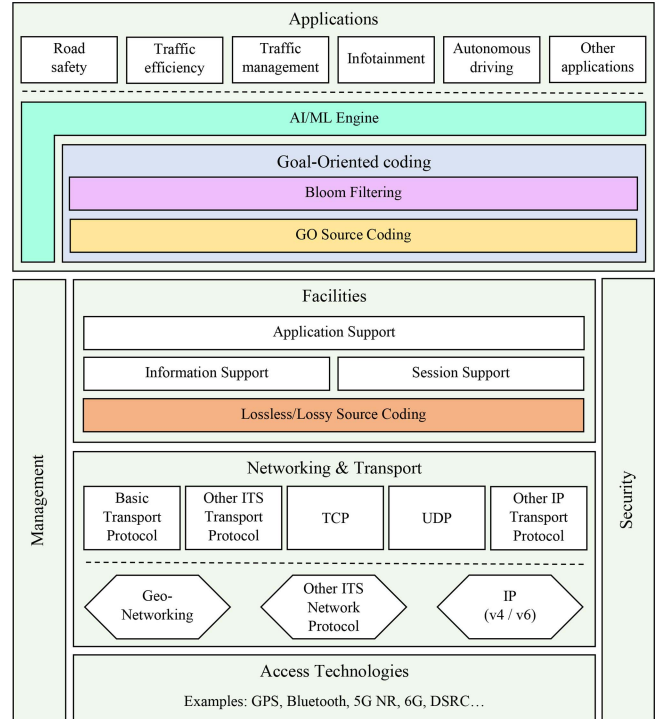


Fig. 3. Proposed architecture.

GO source coding and ending with traditional source coders. Note that in the receiver we must perform the equivalent decoding steps except for the Bloom filtering, as it has not a corresponding decoding stage.

Additionally, we also provide in Fig. 3 a proposal of how our solutions could be integrated into the communications architecture for ITS defined in ETSI EN 302 665 [67] and in ISO 21217 [68]. First, it is important to highlight that the proposal does not affect the network deployment, as it is independent of the access technology and the network and transport layers, so network routers will not be aware of the newly added layers. The main change in the protocol stack is the introduction of the lossless/lossy source coders in the transmitter (and decoder at receiver) as an added facility just above the transport layer, that would let an easy future standardization. Although network routers will not be affected by this new sublayer, it is true that the facilities layer is included into V2X gateways. As lossless source coding is composed by well-known and de facto standardized algorithms and most of lossy ones are DCT-based, it should be easy to patch gateways to include this new sublayer.

TABLE I
PARAMETER SETTINGS FOR PERFORMANCE EVALUATION

Data traffic		DRX		Transmission		Computation	
$\lambda_{ul,p}^{-1}$	30 ms	P_{sleep}	45 mW	f_c	5.9 GHz	f_{CPU}	2 GHz
(D_p^{\max}, D_p^{\min})	(30000, 60000) bytes	P_{PDCCH}	100 mW	G_l	1	N_{CPU}	4
$Q_{ul,p}$	10000 bytes	T_{ON}	1 ms	ℓ	2	$P_{\text{proc}}(f_{\text{CPU}})$	10 W
$\lambda_{ul,ap}^{-1}$	$10 + \text{Exp}(1/10)$ ms	T_i	2 ms	N_0	-174 dB	f_{LL}, f_h, f_A	1 GHz
$(D_{ap}^{\max}, D_{ap}^{\min})$	(10000, 30000) bytes	T_{SDRX}	10 ms	B	20 MHz	ϵ	4
$Q_{ul,ap}$	4000 bytes	T_{LDRX}	20 ms	P_{trx}	26 dBm	$E[\mathcal{W}_{h_i}]$	6.8
λ_{dl}^{-1}	1 ms	N_{SDRX}^{\max}	20 ms	γ_{thr}	3 dB	$E[\mathcal{W}_{GO}]$	5
				d	20 m		

Furthermore, we propose to include all the AI and ML/DL techniques within the application layer, just below the applications. This will elicit researchers to propose and test more GO-ML/DL techniques as they are independent from the network architecture. Moreover, it will permit to easily have several ML/DL GO techniques running in the same network. The consideration of a single AI/ML engine installed in the OBU to account for the configuration of all the deployed GO coders is motivated by the capacity shown by ML/DL techniques to aggregate data in massive IoT networks [69] and also because the correlations between related data streams coming from different sensors have been shown to be useful to compress data [70], [71]. Note that the AI/ML engine operates both vertically and horizontally in the protocol stack, as it processes data from/to applications but simultaneously is the managing intelligence to setup both bloom filtering and GO source coding sublayers. It is also important to note that both Bloom filtering and GO source coding sublayers are optional, so the AI/ML engine may decide when using only one of them, both or even any of them because in some cases or applications it is not required the use of GO communications.

VII. PERFORMANCE EVALUATION

In this section, we evaluate the different proposals addressed in the manuscript to alleviate congestion and, at the same time, to control the delay and energy consumption. To conduct the evaluation, we first define the parameter settings. We consider a free-space path loss model for communication, so the path loss exponent is $\ell = 2$ and calculate the Friis equation parameter as

$$K_0 = \left[\frac{\sqrt{G_l} \cdot c}{4\pi f_c} \right]^2 \quad (29)$$

where $\sqrt{G_l}$ is the product of the transmitter and receiver antenna field radiation patterns in the LOS direction, c is the light speed and f_c is the carrier frequency. Using omnidirectional antennas ($G_l = 1$) and operating in the 5.9-GHz frequency band, we obtain $K_0 = -47.86$ dB. Moreover, we have used $N_0 = -174$ dB, $B = 20$ MHz, $P_{\text{trx}} = 26$ dBm, $\gamma_{\text{thr}} = 3$ dB and $d = 20$ m, so we have $P_{\text{out}} \approx 0.01$, a typical target for the outage probability in wireless network designs [72].

The values used for the energy consumption of the different DRX states are guided by [73] where reference relative power values are defined for two different frequency ranges, FR1 and

FR2. Moreover, in 3GPP Release-17 there are three different sleep states, depending on the components of the RF chain that are shut down, being micro, light, and deep sleep. Assuming as a reference that deep sleep state power is 1 mW [74] and that micro sleep is used, for FR1 we have $P_{\text{sleep}} = 45$ mW and $P_{\text{PDCCH}} = 100$ mW. The parameter settings for the various periods defined for DRX have been chosen from the supported values in the 3GPP standards [75]. A summary of the different parameter settings used for the performance evaluation is provided in Table I.

Although most of the performance measures are analytically computed, the complex patterns of the incoming traffic coupled with the DRX procedure at the terminals makes it difficult, if not infeasible, to obtain closed-form solutions. Namely, DRX has a complex behavior and the departure process of the DRX process represents the incoming traffic for the queueing and transmission steps. For that reason, the DRX behavior has been obtained using a discrete event simulator and the results from the simulator were used as input to the models described in Sections IV and V.

A. Baseline Scenario

We first conduct the performance evaluation of a baseline scenario without any of source coding techniques, which will be used as a reference for the scenarios with source coding. Therefore, in this baseline scenario, all the information gathered by sensors is transmitted. As it has been mentioned, the nature of V2X networks imposes strict requirements in terms of delay. Moreover, we also claim that energy consumption is another key performance parameter that must necessarily be considered in the design of future wireless networks for sustainability reasons. Obviously there are also other interesting KPIs, but our work mainly deals with the performance of V2X networks in terms of delay and energy consumption. The total delay experienced by a packet since it is received from sensing until it is fully transmitted is

$$T^{\text{base}} = T^{\text{DRX}}(\mathcal{A}, T_{\text{ON}}, T_i, T_{\text{SDRX}}, N_{\text{SDRX}}^{\max}, T_{\text{LDRX}}) + T^q(\mathcal{A}', \mathcal{S}, \rho) + T^{\text{trx}}(P_{\text{out}}, \sigma, R) \quad (30)$$

where the arrival process \mathcal{A} is the departure process of the DRX procedure. As the queueing delay calculated in (10) is an upper bound, we are considering the worst case in terms of delay.

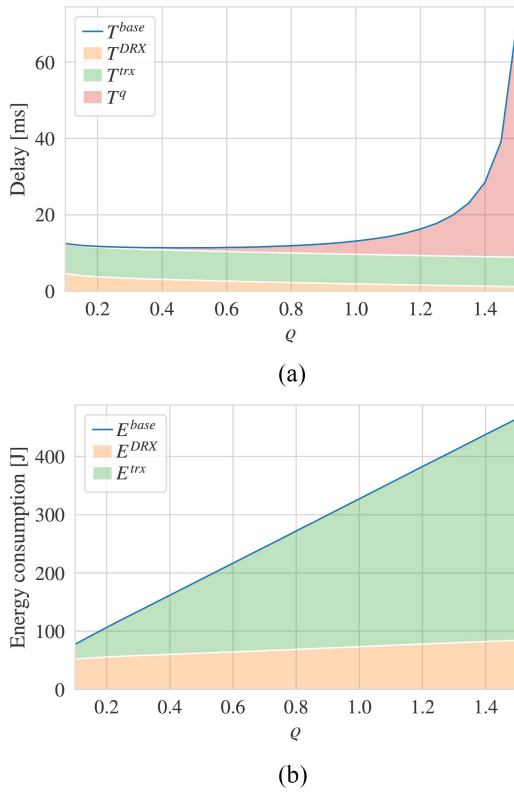


Fig. 4. Baseline setting. (a) Average message delay. (b) Total energy consumption.

On the other hand, energy consumption over a time T in the baseline scenario can be computed as

$$E_T^{\text{base}} = E_T^{\text{DRX}}(P_{\text{trx}}, \Sigma_T \sigma_i, R) + E_T^{\text{trx}}(P_{\text{PDCCH}}, P_{\text{sleep}}, \text{PSF}, T). \quad (31)$$

In Fig. 4, we show how T^{base} and E_T^{base} and their components change with the uplink load ρ . To obtain such values and due to the stochastic nature of results, we have averaged the results obtained with ten simulation runs, being each run 1000 s long. To account for the statistical significance of results, we have also measured the 95% confidence intervals, which are lower than 2% for all cases. So, for the remainder of this article, simulations of length 1000 s were used to evaluate the behavior of the DRX procedure in each specific model. According to Fig. 4(a) and, as expected, T^{trx} is independent from ρ , as it only depends on the message size σ , outage probability P_{out} and throughput R . However, the queuing delay greatly depends on ρ , constituting the main component of T^{base} for high loads. Finally, it is interesting to notice that T^{DRX} decreases as ρ increases, which is due to longer periods of monitoring the PDCCH. That is, the probability of waiting when a new packet arrives decreases as ρ increases and so does the delay due to DRX. Fig. 4(b) shows that the energy consumption for transmission increases more rapidly than the energy consumption due to DRX. As a consequence, the transmission power is, in most cases except for the lowest values of ρ , considerably higher than the energy consumed in the DRX procedure.

B. Lossless Source Coding

In this section, we evaluate the performance of including lossless compression into ITS. Note that we do not consider traditional lossy source coding because reducing the quality of the messages may be unacceptable or even dangerous for the case of security-related V2X messages. First, we evaluate in Fig. 5(a) how the lossless compression gain $G(X)$ changes with the alphabet size K for different values of data entropy $H(X)$. As expected, the best gains are obtained for low values of $H(X)$ and high values of K . It is important to note that K is a design parameter of the lossless source encoder but $H(X)$ depends on the features of the data traffic. However, it is also necessary to highlight that there is a relation between $H(X)$ and K , so we cannot choose arbitrarily large values of K to obtain arbitrarily high compression gains because as K increases, $H(X)$ also tends to increase, as shown in [19].

Probably, the most challenging issue when configuring a lossless source encoder is to define the alphabet size. From Fig. 5(a) we can observe that using larger values of K always leads to higher compression gains. However, Fig. 5(b) shows that increasing K leads to a major increase in delay.

For the remainder of the analyses, and according to [20], we have used $\epsilon = 4$. As we are interested in configuring K to obtain high compression gains but we have bounds on the admissible delay, we can compute the required value of K to have a delay $T_{LL}(x)$ lower than a threshold as

$$K = 2^{\left\lfloor \frac{H(X)}{\epsilon} \log \left(\frac{T_{LL}(x) N_{\text{CPU}} f_{LL}(x)}{\sigma} + e^\epsilon \right) \right\rfloor}. \quad (32)$$

Fig. 6 shows that if the application requires delays lower than 0.5 ms, the gains of the lossless compressor are very low. If higher delays are allowed, a higher compression gain can be achieved.

Once we know how to configure a lossless encoder in terms of K to have a bounded delay and the compression gain we can expect, we evaluate the effect of the source encoder in the V2X network when compared to the baseline scenario without source coding. Fig. 7 shows that the main advantage of using lossless encoding is that it increases the supported load by reducing the amount of transmitted data and avoiding link congestion for high values of ρ (a detailed comparison of the maximum supported load for different source coding schemes is provided in Section VII-F). On the downside, the delay is worse for low values of ρ and lossless encoding noticeably increases the energy consumption with respect to the baseline. Additionally, implementing lossless source encoding requires to perform its correspondent decoding that, although is usually less computationally intensive than the encoding part, can also require a noticeable amount of time and energy for computation. In conclusion, the use of lossless encoding in V2X should be limited to very specific situations.

C. Bloom Filtering

To have a first insight in the configuration of BF, in Fig. 8 we show how false positive probability changes with the length of the filter n , the number of hash functions κ , and the number of elements in the filter m . As it can be seen,

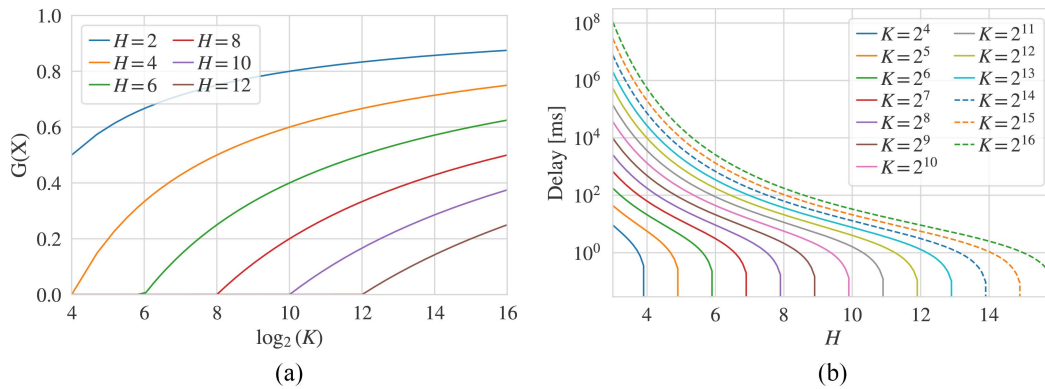


Fig. 5. Effect of alphabet size and data entropy in lossless source coding. (a) Lossless source coder gain. (b) Average message delay.

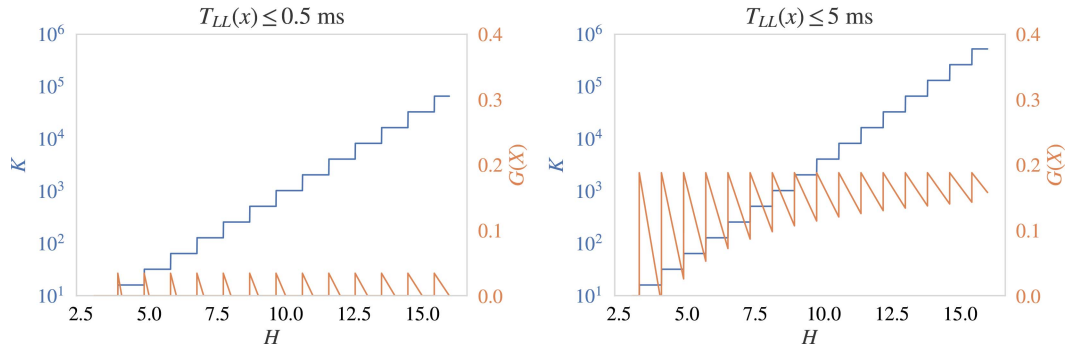


Fig. 6. Configuration of the alphabet size and compression gains that can be achieved in lossless source coding with a maximum delay of 0.5 and 5 ms.

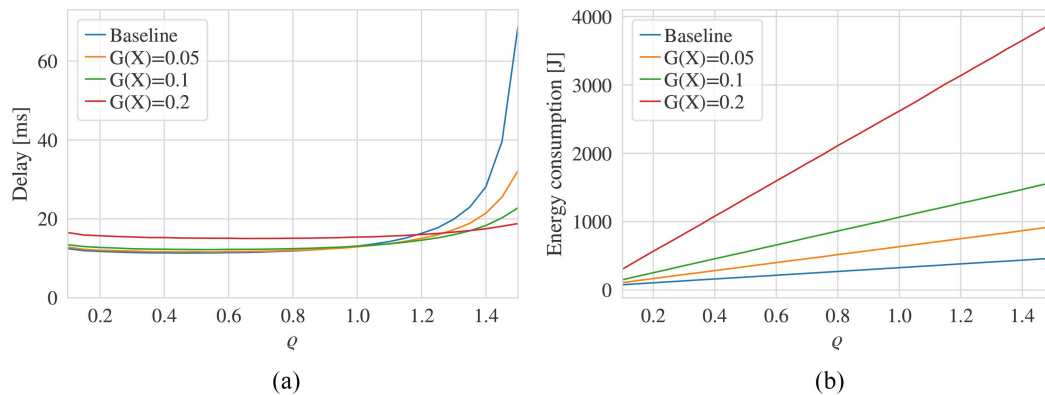


Fig. 7. Performance evaluation of lossless source coding. (a) Average message delay. (b) Total energy consumption.

P_{fp} decreases with n and, for low values of P_{fp} , also with κ . However, when m increases, also P_{fp} increases.

To select appropriate values of κ and n for a certain number of filtering criteria m , we must first analyze the computational cost for the operation of a BF, which depends on the hash function used. Assuming that the most common hash functions provide a uniformly random distribution suited for our proposal, since false positives are not a severe issue further than the traffic is not reduced as expected, the choice of the hash function will be guided by its computational complexity. This is because a lower complexity leads to a lower delay and energy consumption. Thus, and according to [76], we choose to use MD5 as it is the hash function that has the lowest computational complexity among the considered ones.

Based on the results of Fig. 8 and (23) and (24), the value of n does not affect the energy consumption nor the delay, but these are considerably increased when increasing κ . Therefore, although there are proposals that permit using $\kappa > 1$ with the computation of just one hash function [77], we have chosen $\kappa = 1$ and recommend to tune n depending on the number of filtering criteria of the BF to keep a low false positive probability.

In Fig. 9(a), we show the delay and energy consumption with BFs for different BFFs, along with the results in the baseline scenario for comparison. As it can be seen, the delay with BF is always lower than the delay with the baseline, as the computational delay introduced in the operation of the BF is offset by the decrease in delay due to the reduction in network traffic. In fact, even for low values of BFF, we can

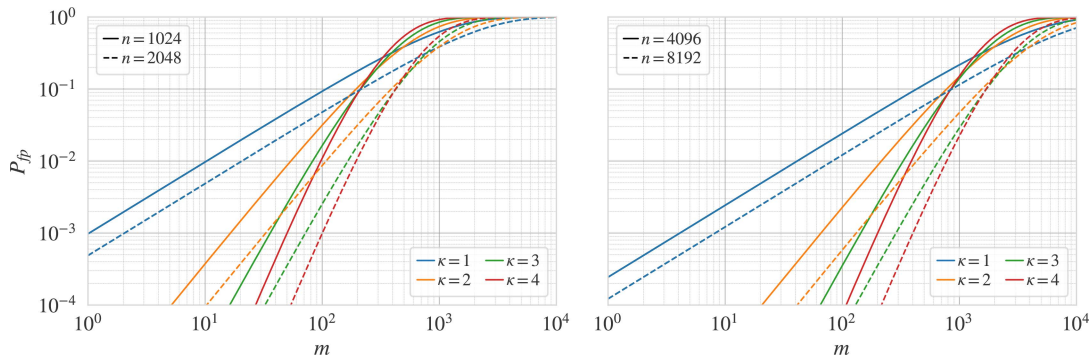


Fig. 8. False positive probability in BFs.

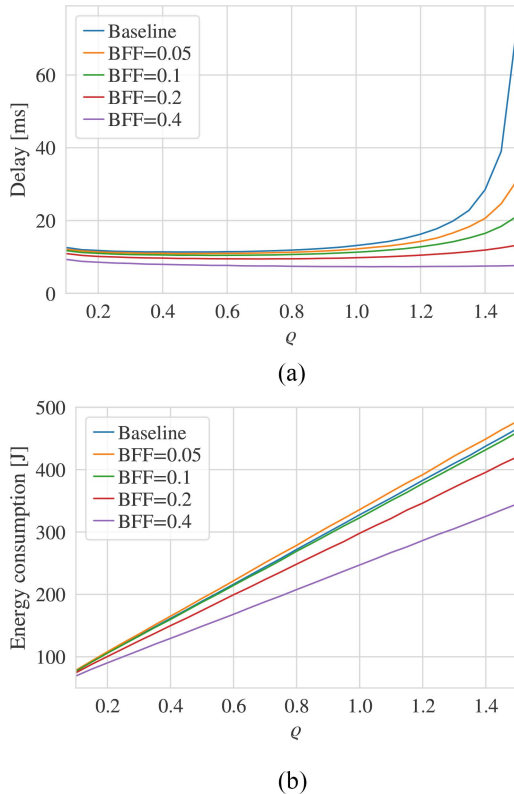


Fig. 9. Performance evaluation of Bloom filtering. (a) Average message delay. (b) Total energy consumption.

obtain a noticeable decrease in delay, along with an increase in the supported load, identified by the maximum value of ρ for a maximum admissible delay (see Section VII-F).

In terms of energy consumption, Fig. 9(b) shows that the use of BFs with low values of BFF can slightly increase the energy consumption with respect to the baseline. However, the energy consumption is drastically reduced with high values of BFF. For that reason, if a large fraction of the V2X traffic is noncrucial and/or redundant for the application and can be filtered out, BFs are a powerful tool to reduce delay and, at the same time, to save energy.

D. Goal-Oriented Source Coding

In this section, we evaluate the cost of GO source encoding in terms of added delay and energy consumption. For this, we

have defined the GO summarization capacity (GOSC) as

$$\text{GOSC} = 1 - \xi \quad (33)$$

where ξ is the length of the information extracted by the GO encoder with respect to the raw data. As GO source coding is expected to greatly summarize V2X messages, by default and unless otherwise indicated, we use $\text{GOSC} = 0.8$. The choice of this value is guided by recent works, such as [78], that show that high summarization capacities can be achieved in the context of semantic communications for image transmission in vehicular environments. Although not specifically for vehicular networks but with the use of DL, similar conclusions have been obtained in [59] and [79] for image and video transmission. Notwithstanding, as the variability of the gains can be very high depending on the desired degree of semantic compression, in this section we consider a wide range values of GOSC to analyze its impact.

In Fig. 10(a), we show the introduced delay of the ML/DL GO techniques for two cases of their asymptotic complexity and also with respect to the case where GO source encoding is not used. First, it is necessary to justify the a priori counter-intuitive behavior regarding the decrease of delay as ρ increases. Although this behavior has also appeared in previous experiments, it is especially noticeable when using GO source encoding with $O(\mathcal{F})$. This behavior is due to the DRX mechanism: as the load ρ increases the UE spends less time sleeping and, therefore, the waiting time until the next ON period decreases. Second, results clearly show that ML/DL techniques with complexity in the order of $O(\mathcal{F}^2)$ are not useful for V2X communications. However, algorithms with complexity $O(\mathcal{F} \cdot \log(\mathcal{F}))$ and especially $O(\mathcal{F})$ are a powerful method to greatly decrease the added delay. Additionally, the shape of the different curves shows that GO source coding can greatly increase the maximum supported load, due to its high degree of data summarization.

If we consider that V2X communications have very strict requirements of delay, we can conclude that GO source encoding can become a key enabler for future V2X networks. Unfortunately, if we inspect energy consumption [Fig. 10(b)] we notice that, for this KPI point of view, the use of GO encoding is only advantageous if the ML/DL algorithm has a complexity in the order of $O(\mathcal{F})$.

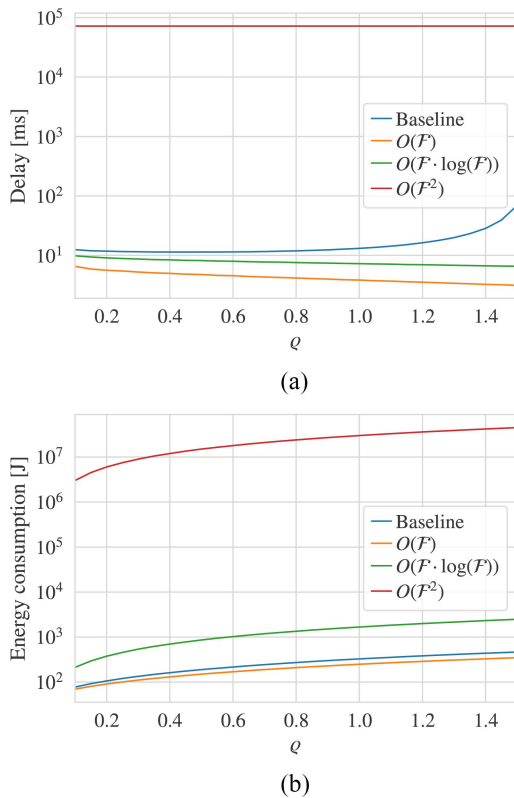


Fig. 10. Performance evaluation of GO encoding. (a) Average message delay. (b) Total energy consumption.

Next, we set $\rho = 1$ and show in Fig. 11 that even for very low values of GOSC, using GO source encoders with complexity in the order of $O(\mathcal{F})$ is beneficial. However, for using algorithms with complexity $O(\mathcal{F} \cdot \log(\mathcal{F}))$ we need values of GOSC higher than 0.22 to obtain advantage of the use of GO source encoding. Note that we have not shown the performance of complexities in the order of $O(\mathcal{F}^2)$ as they are not useful in this setting. On the other hand, when we study energy consumption, we require GOSC values higher than 0.5 to obtain a gain for $O(\mathcal{F})$ algorithms. On the other hand, $O(\mathcal{F} \cdot \log(\mathcal{F}))$ algorithms never let us to improve the energy consumption with respect to not use GO source coding.

E. Combination of Source Coding Mechanisms

In this section, we directly compare two different combinations of the source coding methods according to the architecture shown in Fig. 2. These methods include lossless, Bloom filtering, and GO source encoding, and are labeled as LL, BF and GO, respectively. In the first one, we jointly consider the use of GO communications with BFs and GO source encoding together with lossless encoding (BF+GO+LL). In the second one, we omit the use of lossless encoding (BF+GO), so the evaluation is for the full GO communication proposal, including filtering and source coding. In Fig. 12, we show the delay and energy consumption for both combinations and also include the baseline scenario with no source coding. For the BF we have used MD5 and $BFF = 0.2$, for the lossless encoder we have used $G(X) = 0.1$ and for the GO encoder we have used $GOSC = 0.8$ and an algorithm with linear

complexity $O(\mathcal{F})$. As it can be concluded, the use of lossless coding does not represent an advantage in terms of delay but its effect on energy consumption is harmful. However, we notice that the joint use of Bloom filtering together with GO source coding allows us to increase the maximum supported load and, at the same time, to get high reductions in delay and energy consumption.

In Fig. 13, we cumulatively show the different components that made up the joint result of using filtering and GO source coding. As we can see, in terms of delay, the main component is due to the data transmission (including DRX, queuing, and the transmission) but in terms of energy consumption is more evenly distributed specially for GO source coding and transmission.

Now we evaluate the effect of source coding for different distances between the transmitter and the receiver to analyze if the gains of source coding have a strong dependence on the distance between communicating vehicles. In Fig. 14, we show, for $\rho = 1$, how delay and energy consumption change with the distance between the transmitter and receiver when using and not using source coding. Results reveal another beneficial aspect of using source coding, as the degradation of performance when using long distances is much lower respect to the baseline.

F. Maximum Supported Load

One of the advantages of using source coding is the increase in the supported load, i.e., the wireless link will be able to transport a higher amount of raw data because of the traffic reduction after the source coding mechanisms. In this section, we evaluate the supported load for all the above-studied settings, analyzing the maximum value of uplink load (Q_{\max}) to obtain a packet delay lower than 20 ms. The choice of this value is guided by the shape of the baseline delay curve, because with values of ρ higher than 1.3 the average delay per packet rapidly increases from 20 ms to unstable situations where queues tend to infinity, and so does delay. Fig. 15 shows the maximum supported load to obtain an average packet delay lower than 20 ms. In all cases, the supported load increases when using source coding with respect to the baseline setting (marked as a dashed line). However, the improvements in Q_{\max} when using BF and GO source encoding are very noticeable. For example, when we use a BF that gets a $BFF = 0.4$, we can increase the supported load up to 83.8%. When using GO encoding, we have measured that this value is increased up to 587.7% for $GOSC = 0.8$. Finally, Fig. 15 also includes the maximum supported load when we use several source coding techniques at a time. With the joint use of all the studied source coding techniques (in the settings described in Section VII-E), we have obtained improvements in the maximum supported load of 819.2%.

VIII. CONCLUSION

The scarcity of wireless network resources that can be used for ITS together with the increasing data flows generated by in-vehicle sensors and that must be transmitted jeopardize the successful deployment of ITS services. In this article, we

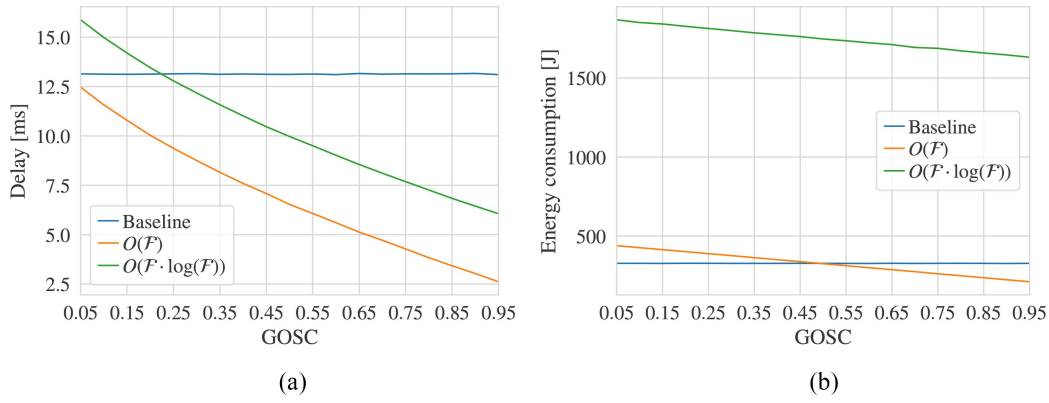


Fig. 11. Performance evaluation of GO coding as a function of its summarization capacity. (a) Average message delay. (b) Total energy consumption.

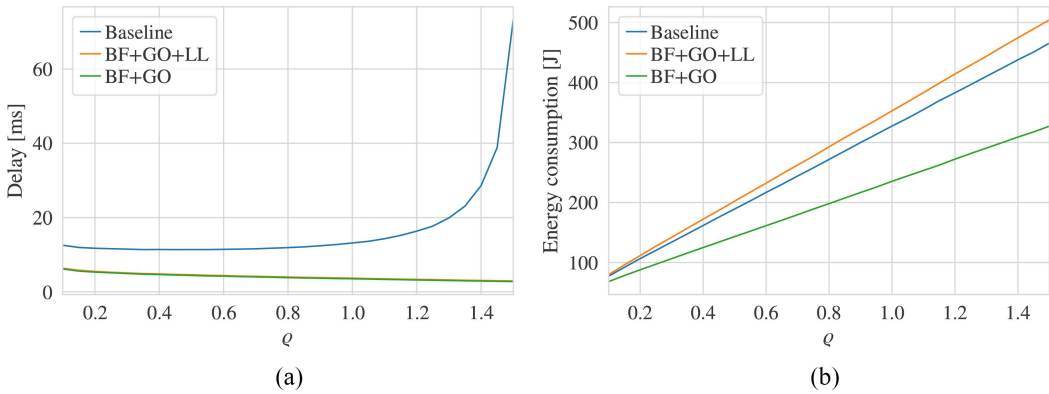


Fig. 12. Global comparison. (a) Average message delay. (b) Total energy consumption.

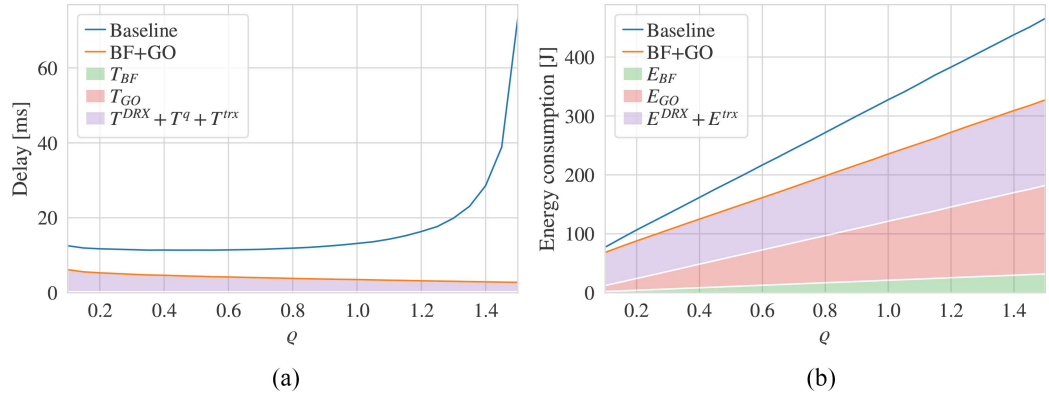


Fig. 13. Analysis of the different components when using Bloom filtering and GO encoding. (a) Average message delay. (b) Total energy consumption.

have studied how source coding and GO communications can contribute to the successful development of future energy-efficient V2X networks. With respect to source coding, we have proposed the use of traditional lossy and lossless source encoding. For GO communications, implemented by machine or deep learning techniques, we have proposed two complementary alternatives. First, we have proposed the use of BFs and we have shown that they are very suited to this purpose as they can include a very high number of filtering rules and will never filter important V2X messages. Second, for GO source coding we have analyzed the algorithmic complexities that these techniques must have to be suitable in the context of V2X networks.

To evaluate the potential benefits of using source coding and GO communications networks to achieve URLLC and energy-efficient ITS, we have focused on two main KPIs: delay and energy consumption, analyzing how those two KPIs can be computed for all the communication and computation tasks that must be performed. Results show that the use of BFs and GO source coding can become key enabling techniques that make future ITS viable. However, we have shown that the use of lossless source coding is usually harmful in the context of V2X networks, because of the huge energy demand it requires. In this article, we have also shown how the use of BFs can greatly reduce both delay and energy consumption. The benefits of BFs are at the cost of some

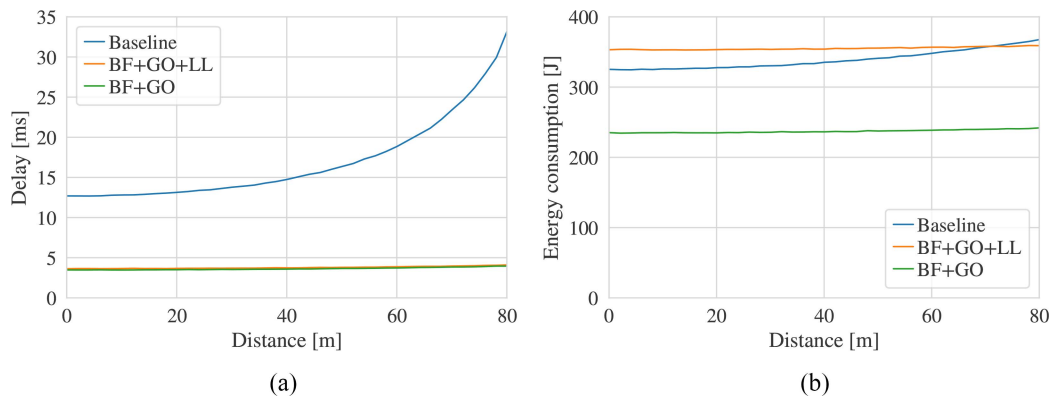


Fig. 14. Effect of distance in gains achieved by source coding. (a) Average message delay. (b) Total energy consumption.

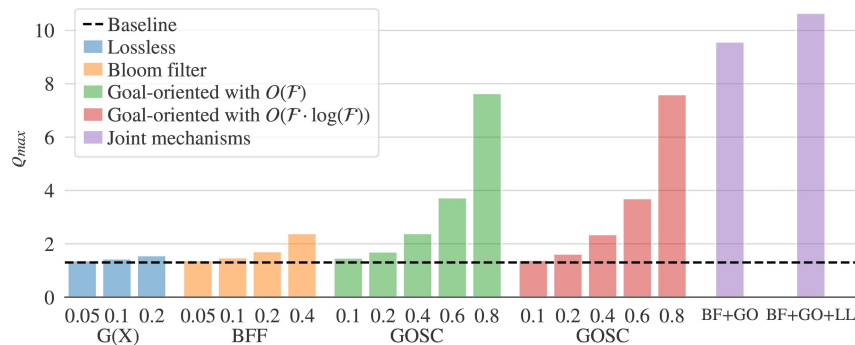


Fig. 15. Maximum supported load to obtain a packet delay lower than 20 ms.

loss of information, but V2X data traffic is expected to be redundant (and if it is not, the impact and also the benefit of BFs is negligible). On the other hand, we show that GO source coding algorithms can become beneficial in terms of delay when having linear or logarithmic complexity. However, in terms of energy consumption, as machine and deep learning techniques are computationally intensive, GO source coding is often not beneficial, unless we use a linear complexity algorithm. Finally, we have evaluated both BFs and GO source coding operating together, showing that both techniques can significantly decrease delay and energy consumption, so we highly recommend their use.

As a future work, we plan to consider the use of GPU instead of CPU for ML/DL computations and, specially, we are interested in the use of a CPU-GPU heterogeneous computing framework to gather the best capabilities of both, as it has been recently successfully proposed in [80] for deep learning inference in mobile devices.

REFERENCES

- [1] "Federal register: Use of the 5.850-5.925 GHz band," Fed. Commun. Comm., Washington, DC, USA, document 86 FR 23323, May 2021.
- [2] B. Liu, J. Liu, and N. Kato, "Optimal beamformer design for millimeter wave dual-functional radar-communication based V2X systems," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 10, pp. 2980–2993, Oct. 2022.
- [3] M. Noor-A-Rahim et al., "6G for vehicle-to-everything (V2X) communications: Enabling technologies, challenges, and opportunities," *Proc. IEEE*, vol. 110, no. 6, pp. 712–734, Jun. 2022.
- [4] S. Suman, Č. Stefanović, S. Došen, and P. Popovski, "Analysis and optimization of the latency budget in wireless systems with mobile edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2022, pp. 5029–5034.
- [5] C. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Champaign, IL, USA: Univ. Illinois Press, 1963.
- [6] P. Popovski, O. Simeone, F. Boccardi, D. Gündüz, and O. Sahin, "Semantic-effectiveness filtering and control for post-5G wireless connectivity," *J. Indian Inst. Sci.*, vol. 100, pp. 435–443, May 2020.
- [7] E. C. Strinati and S. Barbarossa, "6G networks: Beyond Shannon towards semantic and goal-oriented communications," *Comput. Netw.*, vol. 190, May 2021, Art. no. 107930.
- [8] A. Mekrache, A. Bradai, E. Moulay, and S. Dawaliby, "Deep reinforcement learning techniques for vehicular networks: Recent advances and future trends towards 6G," *Veh. Commun.*, vol. 33, Jan. 2022, Art. no. 100398.
- [9] S. Zhang, Q. Wu, S. Xu, and G. Y. Li, "Fundamental green trade-offs: Progresses, challenges, and impacts on 5G networks," *IEEE IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 33–56, 1st Quart., 2017.
- [10] T. Yuan, W. Da Rocha Neto, C. E. Rothenberg, K. Obraczka, C. Barakat, and T. Turetli, "Machine learning for next-generation intelligent transportation systems: A survey," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 4, 2022, Art. no. e4427.
- [11] S. Kaffash, A. T. Nguyen, and J. Zhu, "Big data algorithms and applications in intelligent transportation system: A review and bibliometric analysis," *Int. J. Prod. Econ.*, vol. 231, Jan. 2021, Art. no. 107868.
- [12] R.-C. Chen, C. Dewi, S.-W. Huang, and R. E. Caraka, "Selecting critical features for data classification based on machine learning methods," *J. Big Data*, vol. 7, no. 1, p. 52, 2020.
- [13] P. J. Freire et al., "Reducing computational complexity of neural networks in optical channel equalization: From concepts to implementation," *J. Lightw. Technol.*, vol. 41, no. 14, pp. 4557–4581, Jul. 5, 2023.
- [14] S. Ray, "An analysis of computational complexity and accuracy of two supervised machine learning algorithms—K-nearest neighbor and support vector machine," in *Proc. Data Manag. Anal. Innovat. (ICDMAI)*, vol. 1, 2021, pp. 335–347.
- [15] N. H. Hussein, C. T. Yaw, S. P. Koh, S. K. Tiong, and K. H. Chong, "A comprehensive survey on vehicular networking: Communications, applications, challenges, and upcoming research directions," *IEEE Access*, vol. 10, pp. 86127–86180, 2022.

- [16] I. Soto, M. Calderon, O. Amador, and M. Urueña, "A survey on road safety and traffic efficiency vehicular applications based on C-V2X technologies," *Veh. Commun.*, vol. 33, Jan. 2022, Art. no. 100428.
- [17] Y. Su, X. Lu, L. Huang, X. Du, and M. Guizani, "A novel DCT-based compression scheme for 5G vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10872–10881, Nov. 2019.
- [18] J. Guo, B. Song, Y. He, F. R. Yu, and M. Sookhak, "A survey on compressed sensing in vehicular infotainment systems," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2662–2680, 4th Quart., 2017.
- [19] M. Sepulcre, J. Gozalvez, G. Thandavarayan, B. Coll-Perales, J. Schindler, and M. Rondinone, "On the potential of V2X message compression for vehicular networks," *IEEE Access*, vol. 8, pp. 214254–214268, 2020.
- [20] X. Li, C. You, S. Andreev, Y. Gong, and K. Huang, "Wirelessly powered crowd sensing: Joint power transfer, sensing, compression, and transmission," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 2, pp. 391–406, Feb. 2019.
- [21] X. Luo, H.-H. Chen, and Q. Guo, "Semantic communications: Overview, open issues, and future research directions," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 210–219, Feb. 2022.
- [22] Y. Shi, Y. Zhou, D. Wen, Y. Wu, C. Jiang, and K. B. Letaief, "Task-oriented communications for 6G: Vision, principles, and technologies," 2023, *arXiv:2303.10920*.
- [23] H. Zhang, S. Shao, M. Tao, X. Bi, and K. B. Letaief, "Deep learning-enabled semantic communication systems with task-unaware transmitter and dynamic data," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 170–185, Jan. 2023.
- [24] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Trans. Signal Process.*, vol. 69, pp. 2663–2675, 2021.
- [25] M. Zhu, C. Feng, J. Chen, C. Guo, and X. Gao, "Video semantics based resource allocation algorithm for spectrum multiplexing scenarios in vehicular networks," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC Workshops)*, 2021, pp. 31–36.
- [26] W. Yang et al., "Semantic communications for future Internet: Fundamentals, applications, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 213–250, 1st Quart., 2023.
- [27] F. Pezone, O. Musa, G. Caire, and S. Barbarossa, "Semantic-preserving image coding based on conditional diffusion models," 2023, *arXiv:2310.15737*.
- [28] W. Xu, Y. Zhang, F. Wang, Z. Qin, C. Liu, and P. Zhang, "Semantic communication for the Internet of Vehicles: A multiuser cooperative approach," *IEEE Veh. Technol. Mag.*, vol. 18, no. 1, pp. 100–109, Mar. 2023.
- [29] P. Di Lorenzo et al., "Goal-oriented communications for the IoT: System design and adaptive resource optimization," *IEEE Internet Things Mag.*, vol. 6, no. 4, pp. 26–32, Dec. 2023.
- [30] J. He, K. Yang, and H.-H. Chen, "6G cellular networks and connected autonomous vehicles," *IEEE Netw.*, vol. 35, no. 4, pp. 255–261, Jul./Aug. 2021.
- [31] J. Wang, K. Zhu, and E. Hossain, "Green Internet of Vehicles (IoV) in the 6G era: Toward sustainable vehicular communications and networking," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 391–423, Mar. 2022.
- [32] X. Huang, R. Yu, D. Ye, L. Shu, and S. Xie, "Efficient workload allocation and user-centric utility maximization for task scheduling in collaborative vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3773–3787, Apr. 2021.
- [33] M. Saimler and S. C. Ergen, "Power efficient communication interface selection in cellular vehicle to everything networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2019, pp. 1–6.
- [34] H. Xiao, D. Zhu, and A. T. Chronopoulos, "Power allocation with energy efficiency optimization in cellular D2D-based V2X communication network," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 12, pp. 4947–4957, Dec. 2020.
- [35] F. Jameel, W. U. Khan, N. Kumar, and R. Jäntti, "Efficient power-splitting and resource allocation for cellular V2X communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3547–3556, Jun. 2021.
- [36] A. Ihsan, W. Chen, S. Zhang, and S. Xu, "Energy-efficient NOMA multicasting system for beyond 5G cellular V2X communications with imperfect CSI," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10721–10735, Aug. 2022.
- [37] L. Shan et al., "Energy-efficient resource allocation in NOMA-integrated V2X networks," *Comput. Commun.*, vol. 197, pp. 23–33, Jan. 2023.
- [38] F. Pezone, S. Barbarossa, and P. Di Lorenzo, "Goal-oriented communication for edge learning based on the information bottleneck," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2022, pp. 8832–8836.
- [39] M. R. Ansari, J.-P. Monteuis, J. Petit, and C. Chen, "V2X misbehavior and collective perception service: Considerations for standardization," in *Proc. IEEE Conf. Stand. Commun. Netw. (CSCN)*, 2021, pp. 1–6.
- [40] "Study on evaluation methodology of new vehicle-to-everything (V2X) use cases for LTE and NR," 3GPP, Sophia Antipolis, France, Rep. TR 37.885, V.15.3.0, Jun. 2019.
- [41] K.-H. Lin, H.-H. Liu, K.-H. Hu, A. Huang, and H.-Y. Wei, "A survey on DRX mechanism: Device power saving from LTE and 5G new radio to 6G communication systems," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 156–183, 1st Quart., 2023.
- [42] Z. Tan, J. Zhao, Y. Li, Y. Xu, and S. Lu, "Device-based LTE latency reduction at the application layer," in *Proc. NSDI*, 2021, pp. 471–486.
- [43] J. Kingman, "The single server queue in heavy traffic," *Math. Proc. Cambridge Philos. Soc.*, vol. 57, no. 4, pp. 902–904, 1961.
- [44] G. O. Pérez, D. L. López, and J. A. Hernández, "5G new radio fronthaul network design for eCPRI-IEEE 802.1 CM and extreme latency percentiles," *IEEE Access*, vol. 7, pp. 82218–82230, 2019.
- [45] Y. Su, M. LiWang, L. Huang, X. Du, and N. Guizani, "Green communications for future vehicular networks: Data compression approaches, opportunities, and challenges," *IEEE Netw.*, vol. 34, no. 6, pp. 184–190, Nov./Dec. 2020.
- [46] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of Massive Data Sets*. Cambridge, U.K.: Cambridge Univ. Press, 2020.
- [47] D. Han, W. Chen, B. Bai, and Y. Fang, "Offloading optimization and bottleneck analysis for mobile cloud computing," *IEEE Trans. Commun.*, vol. 67, no. 9, pp. 6153–6167, Sep. 2019.
- [48] J.-B. Wang, J. Zhang, C. Ding, H. Zhang, M. Lin, and J. Wang, "Joint optimization of transmission bandwidth allocation and data compression for mobile-edge computing systems," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2245–2249, Oct. 2020.
- [49] S. Mittal and S. Vaishay, "A survey of techniques for optimizing deep learning on GPUs," *J. Syst. Archit.*, vol. 99, Oct. 2019, Art. no. 101635.
- [50] C. Luo and R. Suda, "A performance and energy consumption analytical model for GPU," in *Proc. IEEE 9th Int. Conf. Dependable, Auton. Secure Comput.*, 2011, pp. 658–665.
- [51] Q. Wang, N. Li, L. Shen, and Z. Wang, "A statistic approach for power analysis of integrated GPU," *Soft Comput.*, vol. 23, pp. 827–836, Feb. 2019.
- [52] S. Tarkoma, C. E. Rothenberg, and E. Lagerspetz, "Theory and practice of Bloom filters for distributed systems," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 1, pp. 131–155, 1st Quart., 2011.
- [53] B. Fan, D. G. Andersen, M. Kaminsky, and M. D. Mitzenmacher, "Cuckoo filter: Practically better than Bloom," in *Proc. 10th ACM Int. Conf. Emerg. Netw. Exp. Technol.*, 2014, pp. 75–88.
- [54] T. M. Graf and D. Lemire, "Xor filters: Faster and smaller than Bloom and Cuckoo filters," *J. Exp. Algorithmics (JEA)*, vol. 25, pp. 1–16, Mar. 2020.
- [55] P. Reviriego, J. Martínez, D. Larrabeiti, and S. Pontarelli, "Cuckoo filters and Bloom filters: Comparison and application to packet classification," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2690–2701, Dec. 2020.
- [56] C. Zhang, H. Zou, S. Lasaulce, W. Saad, M. Kountouris, and M. Bennis, "Goal-oriented communications for the IoT and application to data compression," *IEEE Internet Things Mag.*, vol. 5, no. 4, pp. 58–63, Dec. 2022.
- [57] Z. Cao, H. Zhang, L. Liang, and G. Y. Li, "Machine learning for wireless communication: An overview," *APSIPA Trans. Signal Inf. Process.*, vol. 11, no. 1, p. e33, 2022.
- [58] Y. E. Sagduyu, S. Ulukus, and A. Yener, "Task-oriented communications for NextG: End-to-end deep learning and AI security aspects," 2022, *arXiv:2212.09668*.
- [59] D. Huang, F. Gao, X. Tao, Q. Du, and J. Lu, "Toward semantic communications: Deep learning-based image semantic coding," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 55–71, Jan. 2023.
- [60] Q. Wan, Z. Huang, J. Lu, Y. Gang, and L. Zhang, "SeaFormer: Squeeze-enhanced axial transformer for mobile semantic segmentation," in *Proc. 11th Int. Conf. Learn. Represent. (ICLR)*, 2023, pp. 1–19.
- [61] B. Dong, P. Wang, and F. Wang, "Head-free lightweight semantic segmentation with linear transformer," in *Proc. 37th AAAI Conf. Artif. Intell. (AAAI)*, 2023, pp. 1–9.
- [62] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," *HotCloud*, vol. 10, no. 4, p. 19, 2010.

- [63] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 13, no. 2–3, pp. 203–221, 1996.
- [64] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.
- [65] K. De Vogeleer, G. Memmi, P. Jouvelot, and F. Coelho, "The energy/frequency convexity rule: Modeling and experimental validation on mobile devices," in *Proc. 10th Int. Conf. Parallel Process. Appl. Math.*, 2014, pp. 793–803.
- [66] K. Sadatdiyev, L. Cui, L. Zhang, J. Z. Huang, S. Salloum, and M. S. Mahmud, "A review of optimization methods for computation offloading in edge computing networks," *Digit. Commun. Netw.*, vol. 9, no. 2, pp. 450–461, 2022.
- [67] *Intelligent Transport Systems (ITS)*, ETSI Standard 302 665, V.1. 1.1, 2010.
- [68] *Intelligent Transport Systems—Station and Communication Architecture*, ISO Standard 21217:2020, 2020.
- [69] G. Zhu, J. Xu, K. Huang, and S. Cui, "Over-the-air computing for wireless data aggregation in massive IoT," *IEEE Wireless Commun.*, vol. 28, no. 4, pp. 57–65, Aug. 2021.
- [70] C. M. de Farias et al., "A multisensor data fusion algorithm using the hidden correlations in multiapplication wireless sensor data streams," in *Proc. IEEE 14th Int. Conf. Netw. Sens. Control (ICNSC)*, 2017, pp. 96–102.
- [71] R. F. Albuquerque, P. D. de Oliveira, and A. P. D. S. Braga, "Adaptive fuzzy learning vector quantization (AFLVQ) for time series classification," in *Proc. 37th Conf. North Amer. Fuzzy Inf. Process. Soc.*, 2018, pp. 385–397.
- [72] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
- [73] "Study on user equipment (UE) power saving in NR (release 16)," 3GPP, Sophia Antipolis, France, Rep. TR 38.840, V16.0.0, Jun 2019.
- [74] A. Sen, A. Mondal, B. Palit, J. Jayatheerthan, K. Paul, and S. Chakraborty, "An NS3-based energy module of 5G NR user equipments for millimeter wave networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM) Workshops*, 2021, pp. 1–2.
- [75] "Radio resource control (RRC) protocol specification (Release 17)," 3GPP, Sophia Antipolis, France, Rep. 38.331, V17.3.0, Dec. 2022.
- [76] "Crypto++ benchmarks—Speed comparison of popular crypto algorithms." Accessed: Feb. 9, 2023. [Online]. Available: <https://cryptopp.com/benchmarks.html>
- [77] J. Lu et al., "One-hashing Bloom filter," in *Proc. IEEE 23rd Int. Symp. Qual. Service (IWQoS)*, 2015, pp. 289–298.
- [78] Q. Pan et al., "Image segmentation semantic communication over Internet of Vehicles," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2023, pp. 1–6.
- [79] S. Wang et al., "Wireless deep video semantic transmission," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 214–229, Jan. 2023.
- [80] F. Jia et al., "CoDL: Efficient CPU-GPU co-execution for deep learning inference on mobile devices," in *Proc. 20th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2022, pp. 209–221.



Jose Manuel Gimenez-Guzman received the M.S. and Ph.D. (cum laude and extraordinary prize) degrees in telecommunication engineering from the Universitat Politècnica de València, Valencia, Spain, in 2002 and 2008, respectively.

From 2008 to 2021, he was an Associate Professor with the University of Alcalá, Madrid, Spain. He is currently with the Universitat Politècnica de València, where he is engaged in research and teaching in the areas of analysis and performance evaluation of wireless networks and optimization of complex networked systems. He was a Visiting Scholar with the University of Würzburg, Würzburg, Germany, and Aalborg University, Aalborg, Denmark, in 2012 and 2023, respectively. He has published more than 60 technical papers in these areas in international journals and conference proceedings.

Dr. Gimenez-Guzman is an Associate Editor of the *Wireless Networks*.



Israel Leyva-Mayorga (Member, IEEE) received the B.Sc. degree in telematics engineering and the M.Sc. degree (Hons.) in mobile computing systems from the Instituto Politécnico Nacional (IPN), Mexico City, Mexico, in 2012 and 2014, respectively, and the Ph.D. degree (cum laude and extraordinary prize) in telecommunications from the Universitat Politècnica de València (UPV), Valencia, Spain, in 2018.

He was a Visiting Researcher with the Department of Communications, UPV in 2014, and the Deutsche Telekom Chair of the Communication Networks, Technische Universität Dresden, Dresden, Germany, in 2018. He is currently an Assistant Professor with the Connectivity Section, Department of Electronic Systems, Aalborg University, Aalborg, Denmark, where he served as a Postdoctoral Researcher from January 2019 to July 2021. His research interests include beyond-5G, and 6G networks, satellite communications, and random and multiple access protocols.

Dr. Leyva-Mayorga is an Associate Editor of IEEE WIRELESS COMMUNICATIONS LETTERS and a Board Member of one6G.



Petar Popovski (Fellow, IEEE) received the Dipl.-Ing. and M.Sc. degrees in communication engineering from the University of Sts. Cyril and Methodius in Skopje, Skopje, North Macedonia, in 1997 and 2000, respectively, and the Ph.D. degree from Aalborg University, Aalborg, Denmark, in 2005.

He is a Professor with Aalborg University, where he heads the section on Connectivity and a Visiting Excellence Chair with the University of Bremen, Bremen, Germany. He authored the book *Wireless*

Connectivity: An Intuitive and Fundamental Guide. His research interests are in the area of wireless communication and communication theory.

Prof. Popovski received an ERC Consolidator Grant in 2015, the Danish Elite Researcher Award in 2016, the IEEE Fred W. Ellersick Prize in 2016, the IEEE Stephen O. Rice Prize in 2018, the Technical Achievement Award from the IEEE Technical Committee on Smart Grid Communications in 2019, the Danish Telecommunication Prize in 2020, and the Villum Investigator Grant in 2021. He was a Member at Large at the Board of Governors in IEEE Communication Society from 2019 to 2021. He is currently the Editor-in-Chief of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and the Chair of the IEEE Communication Theory Technical Committee. He was the General Chair for IEEE SmartGridComm 2018 and IEEE Communication Theory Workshop 2019.