Distributed Federated Learning for Ultra-Reliable Low-Latency Vehicular Communications

Sumudu Samarakoon, Member, IEEE, Mehdi Bennis, Senior Member, IEEE, Walid Saad, Fellow, IEEE, and Mérouane Debbah, Fellow, IEEE

Abstract—In this paper, the problem of joint power and resource allocation (JPPA) for ultra-reliable low-latency communication (URLLC) in vehicular networks is studied. Therein, the network-wide power consumption of vehicular users (VUEs) is minimized subject to high reliability in terms of probabilistic queuing delays. Using extreme value theory, a new reliability measure is defined to characterize extreme events pertaining to vehicles’ queue lengths exceeding a predefined threshold. To learn these extreme events, assuming they are independently and identically distributed over VUEs, a novel distributed approach based on federated learning (FL) is proposed to estimate the tail distribution of the queue lengths. Considering the communication delays incurred by FL over wireless links, Lyapunov optimization is used to derive the JPPA policies enabling URLLC for each VUE in a distributed manner. The proposed solution is then validated via extensive simulations using a Manhattan mobility model. Simulation results show that FL enables the proposed method to estimate the tail distribution of queues with an accuracy that is close to a centralized solution with up to 79% reductions in the amount of exchanged data. Furthermore, the proposed method yields up to 60% reductions of VUEs with large queue lengths, while reducing the average power consumption by two folds, compared to an average queue-based baseline.

I. INTRODUCTION

Providing efficient vehicle-to-vehicle (V2V) communications is a necessary stepping stone for enabling autonomous and intelligent transportation systems (ITS) [2]–[6]. V2V communications can extend drivers’ field of view, thus enhancing traffic safety and driving experience, while enabling new transportation features such as platooning, real-time navigation, collision avoidance, and autonomous driving [2], [5]. However, the performance of emerging transportation applications heavily rely on the availability of V2V communication links with extremely low errors and delays. In this regard, achieving ultra-reliable low-latency communication (URLLC) for V2V networks is necessary for realizing the vision of intelligent transportation [2], [7]. The modeling of URLLC has focused on different system design aspects such as guaranteeing the signal-to-interference ratio, data rate, over-the-air/queuing latency, connectivity, age-of-information (AoI), and decoding probability [8], [9]. Since over-the-air latency and queuing latency are coupled, ensuring low queuing latency is required to achieve the much coveted target end-to-end latency of 1 ms. This, in turn, necessitates efficient radio resource management (RRM) techniques [6], [10]–[12]. Furthermore, the increased energy consumption and its negative impact on the environment due to the large number of vehicles in modern transportation system, and improving energy-efficiency/energy savings need to be addressed within RRM in V2V communications [13], [14].

Several existing RRM techniques have been proposed for enabling ultra-reliable low-latency vehicular communications while factoring in several challenges such as rate maximization, delay minimization, improving energy-efficiency, energy saving, and vehicle clustering/platooning [1], [4]–[6], [13]–[21]. In [5], the performance of vehicular platooning is optimized while jointly considering the delay of the wireless network and the stability of the vehicle’s control system. By grouping vehicles into clusters, the work in [6] minimizes the total transmission power in a vehicular network while considering queuing latency and reliability. In [13], an energy-efficient resource allocation algorithm is proposed for cooperative V2V communication systems. The work in [14] proposed an energy saving sleep mode strategy for access points serving motorway vehicular traffic. The problem of vehicle network clustering is studied in [15] to reduce the power consumption of V2V communications. In [16], a joint resource allocation and power control algorithm is proposed to maximize the V2V sum rate. The authors in [17] optimize the beam alignment and scheduling among vehicles to reduce the V2V communication delays. In [18], the tradeoff between service delay and transmission success in V2V communications is optimized. The URLLC aspects of this prior art [5], [6], [13]–[19] are captured by either improving the average latencies or imposing a probabilistic constraint to maintain small queue lengths. Although such a probabilistic constraint on the queue length improves network reliability, it fails to control rare events in which large queue lengths occur with low probability, i.e., the tail distribution of queue lengths. As a result, if the network relies on these existing schemes, some of the vehicular users (VUEs) may experience unacceptable latencies yielding degraded performance [4], [22]–[25].

In practice, to enable a truly URLLC experience, it is im-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2019.2956472, IEEE Transactions on Communications
perative to model and capture extreme, low probability events. To this end, extreme value theory (EVT), a powerful tool from statistics that characterizes the occurrences of extreme, low probability events instrumental in enabling URLLC [26]. In [23], EVT is used to model the distributions of data rates exceeding a threshold for few traffic traces and then, the accuracy of the analytical model is evaluated using simulations. The work in [24] studies the statistical distributions of inter-beacon delays in safety applications for vehicular adhoc networks (VANETs) using EVT. The authors in [25] use EVT to model the peak distribution of the orthogonal frequency division multiplexing envelope while characterizing the variations in peak-to-average-power ratios. The work in [4] employs EVT to characterize the statistics of maximal queue length so as to control the worst-case latency of V2V communication links therein. Characterizing the distribution of extreme events using EVT, i.e., determining the location, shape, and scale parameters of the tail distribution, in the above works necessitates the acquisition of sufficient samples capturing extreme events. Depending on the network size and the quality of the communication within the network, the process of gathering samples over the network may introduce unacceptable overheads that are not investigated in the aforementioned works. In a real-time system such as a V2V communication network, VUEs may have access to limited number of queue length samples (particularly those that are locally in excess over a high threshold) and hence they are unable to estimate the tail distribution of the network-wide queue lengths. Therefore, roadside units (RSUs) can assist in gathering samples over the network at a cost of additional data exchange overheads. Furthermore, due to the resource limitations available for V2V communication, VUEs may be unwilling to allocate their resources to share their individual queue state information (QSI) with an RSU and other VUEs. This shortcoming warrants a collaborative learning model that does not rely on sharing individual QSI.

Recently, federated learning (FL) was proposed as a decentralized learning technique where training data is distributed (possibly unevenly) across learners, instead of being centralized [27], [28]. FL allows each learner to derive a set of local learning parameters from the available training data, referred to as local model. Instead of sharing the training data, learners share their local models with a central entity, which in turn does model averaging then sharing a global model with the learners. In [27], the applicability of several existing algorithms for FL are studied and a novel algorithm is proposed to handle the sparse data available at individual learners. The means of minimizing the communication cost by sharing a reduced number of parameters of FL models are discussed in [29]. In [28], FL is used to develop distributed learning models for multiple related tasks simultaneously, referred to as multitask learning. The recent work in [30], proposes a new FL protocol that solves a client selection problem with resource constraints in mobile edge computing. Our prior work in [1] proposes a distributed FL-based algorithm for VUEs based on a maximum likelihood estimation (MLE). However, this prior work does not consider sharing wireless resources for FL and V2V communications, whereby the impact of FL over shared wireless resources on V2V URLLC is not investigated. To the best of our knowledge, with the exception of [1], no work has studied the use of federated learning in the context of URLLC.

The main contribution of this paper is to propose a distributed, FL-based, joint transmit power and resource allocation framework for enabling ultra-reliable and low-latency vehicular communication. We formulate a network-wide power minimization problem while ensuring low latency and high reliability in terms of probabilistic queue lengths. To model reliability, we first obtain the statistics of the queue lengths exceeding a high threshold by using the EVT notion of a generalized Pareto distribution (GPD) [26]. Using the statistics of the GPD, we impose a local constraint on extreme events pertaining to queue lengths exceeding a predefined threshold for each VUE. Here, the characteristic parameters of the GPD are known as scale and shape, which are obtained by using the MLE. In contrast to the classical MLE design which requires a central controller (e.g., RSU) to collect samples of queue lengths exceeding a threshold from all VUEs in the network, using FL every vehicle builds and shares its own local model (two gradient values) with the RSU. The RSU aggregates the local models, does model averaging across vehicles, and feeds back the global model to VUEs. Leveraging different time scales, using our proposed approach, each VUE learns its GPD parameters locally in a short time scale while the model averaging (global learning) takes place in a longer time scale. Here, an assumption of independently and identically distributed queue length samples exceeding the threshold over all VUEs is imposed to ensure accurate GPD parameter estimation using FL. In our model, we take into account the communication overheads of URLLC due to the model exchange over shared wireless resources. Then, we propose a distributed algorithm that allows all VUEs to simultaneously learn the GPD parameters using FL. To further reduce the overhead due to the need of synchronization and simultaneous model sharing, next we develop an asynchronous FL algorithm for MLE that allows VUEs to model and independently learn the tail distribution of queue lengths in a distributed manner. Finally, Lyapunov optimization is used to decouple and solve the network-wide optimization problem per VUE. Simulation results show that the proposed solutions estimate the GPD parameters very accurately compared to a centralized learning module and yields significant gains in terms of reducing the number of VUEs with large queue lengths while minimizing power consumption. For dense systems with 100 VUE pairs, the proposed solution yields about 60.9% reduction of VUEs with large queue lengths by reducing the power consumption by two folds, compared to a baseline model that controls the reliability using a probabilistic constraint on average queue lengths. Furthermore, 28.6% and 33.2% reductions in averages and fluctuations of extreme queue lengths, respectively, can be seen in the proposed solution compared to the aforementioned baseline.

The rest of the paper is organized as follows. Section II describes the system model and the network-wide power minimization problem. The distributed solution based on EVT and Lyapunov optimization is presented in Section III. In Section IV, estimation of the extreme value distribution using...
FL and the cost of enabling FL for both synchronous and asynchronous approaches are discussed. Section V evaluates the proposed solution by extensive set of simulations. Finally, conclusions are drawn in Section VI.

II. SYSTEM MODEL AND PROBLEM DEFINITION

Consider a vehicular network consisting of a set $U$ of communicating VUE pairs, using an RSU that allocates a set $N$ of resource blocks (RBs) over a partition of the network $Z$ defined as groups. Here, a group consists of VUE pairs that can reuse the same RBs with low-to-no interference on one another. The RSU allocates RBs orthogonally across the groups to reduce the interference among nearby VUE pairs. Hence, a VUE pair $u$ is only allowed to use the subset $N_{z(t,u)} \subseteq N$ of RBs allocated to its corresponding group $z(t,u)$ at time $t$. We denote the VUE transmitter (vTx) and receiver (vRx) that belong to VUE pair $u$ by vTx $u$ and vRx $u$, hereinafter. An illustration of our system model is presented in Fig. 1.

Let $p_u(t) = \{p_u^n(t)\}_{n \in N_{z(t,u)}}$ and $h_{uu'}(t) = [h_{uu'}^n(t)]_{n \in N_{z(t,u)}}$ be, respectively, the transmit power vector of vTx $u$, and the channel gain vector between vTx $u$ and vRx $u'$ over the subset of allocated RBs at time $t$. Depending on whether the vTx and vRx are located in the same lane or separately in perpendicular lanes, the channel model is categorized into three types: i) Line-of-sight (LOS): both vTx $u$ and vRx $u'$ are located in the same lane, ii) Weak-line-of-sight (WLOS): vTx $u$ and vRx $u'$ are in perpendicular lanes and at least one of them is located at a distance of no more than $d_0$ from the corresponding intersection, and iii) Non-line-of-sight (NLOS), otherwise. Let $(x_u, y_u)$ and $(x_{u'}, y_{u'})$ be the Cartesian coordinates of vTx $u$ and vRx $u'$, respectively. The channel $h_{uu'} = \phi_{uu'} \Phi_{uu'}$ includes a fast fading component $\phi_{uu'}$ following a Rayleigh distribution with a unit scale parameter for LOS, a Nakagami-m distribution with $m = 1.41$ and a unit scale for both WLOS and NLOS, in addition to a path loss model $\Phi_{uu'}$ for urban areas at $5.9$ GHz carrier frequency [31]:

$$\Phi_{uu'} = \begin{cases} \ell \| (x_u, y_u) - (x_{u'}, y_{u'}) \|_2^{-c} & \text{for LOS}, \\ \ell' \| (x_u, y_u) - (x_{u'}, y_{u'}) \|_2^{-c} & \text{for WLOS}, \\ \ell' \| (x_u - x_{u'}) \|_2^{-c} \| (y_u - y_{u'}) \|_2^{-c} & \text{for NLOS}, \end{cases}$$

where $\| x \|_t$ is the $t$-th norm of vector $x$, $c$ is the path loss exponent, and the path loss coefficients $\ell$ and $\ell'$ satisfy $\ell' < \ell (\frac{d_0}{d})^c$. The transmission rate between the vTx-vRx pair $u$ is given by,

$$r_u(t) = \sum_{n \in N_{z(t,u)}} r_u^n(t) = \sum_n W \log_2 \left(1 + \frac{h_{uu'}^n(t)p_u^n(t)}{I_u^n(t)+W_N} \right),$$

where $I_u^n(t) = \sum_{u' \in \mathcal{U}(t)} h_{uu'}^n(t)p_{uu'}^n(t)$ is the interference from other vTxs, $W$ is the bandwidth of each RB, and $N_0$ is the noise power spectral density. At each time $t$, $a_u(t)$ data bits are randomly generated with a mean of $\pi_u$ at vTx $u$ that must be delivered to its corresponding vRx. Thus, at the vTx, a data queue is maintained and has the following dynamics:

$$q_u(t+1) = [q_u(t) + a_u(t) - r_u(t)]^+,$$

where $[x]^+ = \max(x, 0)$.

The number of vehicles is expected to grow continuously, in which improving energy efficiency and saving energy in vehicular networks is a key requirement. Our goal is therefore to minimize the network-wide power consumption while ensuring URLLC. Considering use cases such as information exchange on blind-spots or sequences of future actions (turning, parking, slowing/speeding) based on the traffic ahead, it is important to optimize decision making taking into account the queue length and its tail distributions [4], [6], [32]. In this view, here, reliability is achieved by guaranteeing queue stability for each vTx while keeping outages below a predefined threshold, i.e., the probability that the queue length exceeding a threshold $q_0$ is below a certain probability $\epsilon$. The reliability conditions can now be formally defined as:

$$E[q_u] = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} q_u(t) < \infty \quad \forall u \in U,$$

$$\Pr(q_u(t) \geq q_0) < \epsilon \quad \forall u \in U, \forall t.$$ 

Note that the above reliability constraints cannot cope with the extreme cases in which the queue lengths $q_u(t) > q_0$ that occur with a probability below $\epsilon$. Such extreme cases essentially correspond to the worst case network queuing latency (as well as end-to-end latency [6], [10], [11]) which are a key determinant of the URLLC performance and, hence, must be properly addressed. In this regard, the set of queue length samples exceeding the threshold $q_0$ over the network is defined as $M = \{(q - q_0)q > q_0: q \in \{q_u(t)\}_{u \in \mathcal{U}(t)}\}$. Then, a sample of an extreme event, i.e., queue length exceeding $q_0$, is $M \in M$. By imposing the following constraints for all VUEs,

$$\lim_{T \to \infty} \sum_{t=1}^{T} (q_u(t) - q_0) I_t^a / \sum_{t=1}^{T} I_t^a \leq E[M],$$

$$\lim_{T \to \infty} \sum_{t=1}^{T} (q_u(t) - q_0)^2 I_t^a / \sum_{t=1}^{T} I_t^a \leq E[M^2],$$

each VUE $u$ can better control the fluctuations of its queue and maintain its extreme values below the desired threshold. Here, $I_t$ is an indicator function with $I_t = 1$ when $q(t) > q_0$, and $I_t = 0$, otherwise. Note that the benefits of imposing
\[ \mathbb{E}[M] \text{ and } \mathbb{E}[M^2] \text{ as targets of (6) and (7) include: i) The targets are naturally generated within the system which can prevent infeasibility and instability under predefined targets, ii) The chosen targets allow vehicular users with larger queue lengths to exploit more transmit power compared to ones with short queue lengths allowing to control interference, and iii) In consequence, the tail distributions of all vehicles converge short queue lengths allowing to control interference, and prevent infeasibility and instability under predefined targets.} \]

\[ \text{E} \text{M} \text{mainly shows that, as } M > q \text{ bounded and equivalent to } } \sigma/(1-\xi) \text{ and } 2\sigma^2/(1-\xi)(1-2\xi), \text{ respectively, only if } \xi < 1/2. \text{ In this regard, constraints (6) and (7) for all } u \in U \text{ can be rewritten as follows:} \]

\[ \lim_{T \to \infty} T^{1/2} \sum_{t=1}^{T} (q_u(t) - q_0) I_t \leq \frac{\sigma}{1-\xi}, \quad (10) \]

\[ \lim_{T \to \infty} T^{1/2} \sum_{t=1}^{T} (q_u(t) - q_0)^2 I_t / \sum_{t=1}^{T} I_t \leq \frac{2\sigma^2}{(1-\xi)(1-2\xi)}. \quad (11) \]

Assisted by the RSU, each VUE pair can estimate \( \xi \) locally without sharing its QSI, hence effectively decoupling the constraints (6) and (7), and imposing them locally as in (10) and (11), respectively.

### B. Lyapunov Optimization for Power Allocation

By using EVT to model \( M = q - q_0 > 0 \) and its first two moments, we recast the original problem into an equivalent form:

\[ \text{minimize } \left[ p_u(t) \right]_{t \in U} \quad \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \left( \sum_{u \in U} 1^1 p_u(t) \right) \quad (12a) \]

subject to \( (3)-(5), (8c), (10), (11). \quad (12b) \]

To devise a tractable solution for the modified stochastic optimization problem in (12), we resort to Lyapunov optimization [33]. To this end, first, we should model the time average constraints as virtual queues. As such, the reliability constraint in (5) can be recast as \( \mathbb{E}[q_u] \leq \epsilon q_0 \) for each VUE \( u \in U \) using the upper bound condition \( \text{Pr}(q_u \geq q_0) \leq \mathbb{E}[q_u]/q_0 \) based on the Markov’s inequality [34]. Our next goal is to introduce a virtual queue \( \Psi_u \) for the aforementioned constraint instead of (4) and (5). Now, the time average constraints in (12b) for all \( u \in U \) are modeled by virtual queues as follows:

\[ \Psi_u(t+1) = \Psi_u(t) + (q_u(t+1) - \epsilon q_0)^+ \quad (13a) \]

\[ A_u(t+1) = A_u(t) + (q_u(t+1) - q_0 - \frac{\sigma}{1-\xi}) I_t^+ \quad (13b) \]

\[ B_u(t+1) = B_u(t) + I_t (q_u(t+1) - q_0)^2 - \frac{2\sigma^2 I_t}{(1-\xi)(1-2\xi)} \quad (13c) \]

Let \( \Xi_u(t) = [q_u(t), \Psi_u(t), A_u(t), B_u(t)] \) be the combined queue with \( \Xi(t) = [\Xi_u(t)]_{u \in U} \) and its quadratic Lyapunov function \( L(\Xi(t)) = \frac{1}{2} \Xi(t)^\top \Xi(t) \). The one-slot drift of the Lyapunov function is defined as \( \Delta L_t = L(\Xi(t+1)) - L(\Xi(t)) \).

**Proposition 1**: The upper bound of the Lyapunov drift is given by,

\[ \Delta L_t \leq \sum_{u \in U} \left[ \Delta_u + (q_u(t) - r_u(t)) \left\{ (1+\Psi_u(t) - \epsilon q_0) q_u(t) - \epsilon \Psi_u(t) + (2(q_u(t) - q_0)) B_u(t) + (q_u(t) - q_0)^2 - M_2 \right\} + q_u(t) + A_u(t) - M_1 I_t \right] + \Delta_0, \quad (14) \]

where \( M_1 = q_0 + \frac{\sigma}{1-\xi} \) and \( M_2 = \frac{2\sigma^2}{(1-\xi)(1-2\xi)} \) are the first two moments of the tail distribution of queue lengths. A constant bound \( \Delta_0 \) and a set of terms \( \Delta_u \) independent from the control variables at time \( t \) are given in (25) and (26), respectively.

**Proof**: See Appendix A.

By controlling the upper bound given in Proposition 1, the network can ensure the stability of both actual and virtual queues.
The conditional expected Lyapunov drift at time \( t \) is defined as \( \mathbb{E}[L(\Xi(t + 1)) - L(\Xi(t))|\Xi(t)] \). We define \( V \geq 0 \) as a parameter that controls the tradeoff between the queue length and the accuracy of the optimal solution of (12). We then find the network policies by introducing a penalty term \( \mathbb{E}[\sum_u q_u(t)\Xi(t)] \) to the expected drift and minimizing the upper bound of the drift plus penalty (DPP), \( \mathbb{E}[\sum_u q_u(t)\Xi(t)] + \mathbb{E}[\Delta L_u\Xi(t)] \). As a result, our goal will now be to minimize the following upper bound:

\[
\sum_{u \in \mathcal{U}} V^1p_u + (\alpha_u(t) - r_u(t))\left((1 + \Psi u(t) - \epsilon q_u(t)
\right)
\]

\[
+ [2(q_u(t) - q_0)(B_u(t) + q_u(t) - q_0)^2 - \frac{2\sigma^2}{(1 - \xi)(1 - 2\xi)}]
\]

\[
+ q_u(t) + A_u(t) - q_0 - \frac{\sigma}{\xi - \tau}I_t - \epsilon \Psi u(t))\right]
\]

(15)

at each time \( t \). Assuming that VUEs maintain channel-quality indicators (CQIs), each VUE can estimate the interference \( I_u\Xi(t) \approx I_u\Xi(t) \) based on past observations (time averaged interference) [35]. Hence, the minimization of the upper bound can be decoupled among VUEs as follows:

\[
\min_{p_u(t)} \left\{ \sum_{n \in N_e(t,u)} [Vp_n^u(t) - \alpha_u(t)\ln(1 + \eta_n^u(t)p_n^u(t))] \right\}
\]

subject to

\[
\sum_{n \in N_e(t,u)} p_n^u(t) \leq p_0,
\]

(16a)

\[
p_n^u(t) \geq 0 \quad \forall n \in N_e(t,u),
\]

(16b)

where \( \alpha_u(t) = \frac{w}{I_\Xi(t)} \left[ (1 + \Psi u(t) - \epsilon q_0)q_u(t) - \epsilon \Psi u(t) + \right]
\]

\[
B_u(t) + q_u(t) - q_0 - \frac{\sigma}{\xi - \tau}I_t - \epsilon \Psi u(t) + \right)
\]

\[
q_u(t) + A_u(t) - q_0 - \frac{\sigma}{\xi - \tau}I_t - \epsilon \Psi u(t))\right]
\]

\[
\sum_{n \in N_e(t,u)} \eta_n^u(t) = \frac{h_n^u(t)}{I_u\Xi(t) + W N_u}. \]

The optimal solution of the convex optimization problem of (16) is obtained by a water-filling algorithm [36] where \( p_n^u(t)^* = \frac{\alpha_n^u(t)}{\eta_n^u(t)} - \frac{1}{\eta_n^u(t)} \), with \( \lambda_n(t) \geq 0 \) being the Lagrangian dual coefficient corresponding to constraint (16b). Since the first two moments, \( M_1 \) and \( M_2 \), of the distribution of queue lengths exceeding \( q_0 \) impact the optimal solution \( p_n^u(t)^* \), in what follows we propose a mechanism to estimate the GPD parameters accurately.

IV. LEARNING THE PARAMETERS OF THE MAXIMUM QUEUE DISTRIBUTION

The optimal power allocation problem in (16) relies on the characteristics of the excess queue distribution \( G^d_M(m) \). Hence, estimating the parameters \( \sigma \) and \( \xi \) with high accuracy using QSI samples gathered over the network is imperative. In this regard, modeling the distribution of queue lengths exceeding the threshold requires a central controller (e.g., the RSU) to compute and communicate with all VUEs at each time \( t \).

A. Queue Sampling via Block Maxima (BM)

Let \( w \) be the block length (or time window) during which each VUE draws at most one (the maximum) queue length sample if the queue length exceeds the threshold \( q_0 \). The size of \( w \) should be sufficiently large to minimize correlation between QSI samples while being sufficiently small to avoid undersampling. Henceforth, the assumption of independent queue length samples over VUEs for EVT-based modeling is satisfied. We now define \( T_k = \{(k-1)w, (k-1)w + 1, \ldots, kw - 1\} \) as the set of time instants during block \( k \) in \( \mathbb{N} \). Then, the set of queue samples at time \( t \) is \( Q_u(t) = \{q_u(t_k^* - 1) > \arg \max_{x \in T_k} q_u(x), k \in \{1, \ldots, \lfloor t/w \rfloor\}\} \) with a sample size \( K_u(t) \). Note that 0 \( \notin Q_u(t) \) for all \( u \in \mathcal{U} \) and the total number of samples may vary across VUEs since each VUE can independently perform its own QSI sampling process. Fig. 2 illustrates each VUE’s QSI sampling process.

B. RSU-Centric GPD Parameter Estimation

As shown in Section III-A, the distribution of the queue lengths exceeding the threshold is characterized by two parameters \( d = [\sigma, \xi] \) which need to be accurately estimated. For this purpose, we use MLE [37] whose objective is to find the best set of parameters \( d \) that fits the GPD \( G^d_X(-) \) to the samples via maximizing the log likelihood function (or minimizing its negative) as follows:

\[
\min_{d \in D(q)} \sum_{Q \in Q} \log G^d_X(Q),
\]

(17)

where \( D(Q) = \{\sigma, \xi \in \mathbb{R}^2 | \sigma > 0, \xi < 1, 1 + \xi \sigma > 0\} \) and \( Q \in \{Q_u\}_{u \in U} \) is the set of network queue length samples. Here, we omit the time index \( t \) for simplicity. Note that the likelihood function is a smooth function of \( d \) and a summation over all the samples in \( Q \), and thus, its gradient over a sample \( Q \) can be derived as follows.

\[
\nabla_d f^d(Q) = \left[ \frac{\partial f^d(Q)}{\partial \sigma} \right] = \left[ \frac{1}{1 + \xi Q/\sigma} \left[ 1 + \frac{1 + \xi}{\xi Q/\sigma} \right] \right].
\]

(18)

Proof: see Appendix B.

Using the stochastic variance reduced gradient descent (SVRGD) technique [38] alongside \( \nabla_d f^d(Q) \) at the RSU, the optimal \( d^* \) can be derived in an iterative manner (iterating over the sample set) with fast convergence. For a given predefined step size \( \delta > 0 \), the evaluation procedure of the GPD parameters using SVRGD over a sample \( Q \) at iteration \( \tau \) is defined as follows:

\[
\begin{align*}
\{ d(\tau) = d(\tau - 1) - \delta [\nabla d f^d(\tau - 1) - \nabla d f^d(\tau)] + \Upsilon(\tau) \},
\end{align*}
\]

(19)

where \( \Upsilon(\tau) = \frac{1}{\tau - 1} \sum_{\tau' = 1}^{\tau - 1} d(\tau') \) is an average estimate of \( d \) over previous iterations and \( \Upsilon(\tau) = \frac{1}{\tau} \sum_{Q \in Q} \nabla d f^d(Q) \) is an estimate of the gradient, respectively. After computing the GPD parameters by iterating over the sample set, RSU shares the optimal GPD parameters with all the VUEs. This RSU-centric GPD parameter estimation is referred to as “CEN”, hereinafter, and it is summarized in Algorithm 1.

In CEN, all VUEs in the network need to frequently upload their local queue length samples to the RSU by reusing the
function as follows:

\[ f^d(Q) = \frac{1}{|Q|} \sum_{Q \in \mathbb{Q}} \log G_X^d(Q) = \sum_{u \in \mathbb{U}} \kappa_u f^d(Q_u), \]

where \( \kappa_u = \frac{|Q_u|}{|Q|} = \frac{K_u}{\sum_{u'} K_{u'}}. \) In (20), we express the likelihood function of the network as a weighted sum of likelihood functions per VUE. Hereinafter, for simplicity, we use \( f^d \) and \( f^d_u \) instead of \( f^d(Q) \) and \( f^d(Q_u) \), respectively. The idea behind FL is to use \( f^d_u \) to evaluate \( d_u \) and \( \nabla d_u f^d_u \) locally, where \( d_u \) is the FL estimate of \( d \) at VUE \( u \), and update the local estimations via sharing the individual learning models \((\nabla d_u f^d_u, d_u, K_u, Q_u)\) where \( Q_u = \max Q_u \). Note that sharing \( Q_u \) over the network through RSU is sufficient to determine the domain \( D(Q) \), which is needed for the SVRGD procedure.

To evaluate the gradients and GPD parameters locally, VUE \( u \) uses the SVRGD with a step size \( \delta_u = \delta/K_u \) [27]. In this case, given the local and global copies of the GPD parameters and gradients at a time \( \tau \), \( d_u(\tau), d(\tau), \tau_u(\tau) \), and \( \Upsilon(\tau) \), respectively, the local GPD parameters and gradients are updated for each QSI sample \( i_u \in Q_u \), process A in Fig. 2, as follows:

\[
\begin{align*}
    y_u &= d_u(\tau) - \delta_u \left[ \nabla d_u f^d_u(i_u) - \nabla d f^d(i_u) + \Upsilon(\tau) \right], \\
    d_u(\tau) &= \arg \min_{d_u \in D(Q_u)} \| y_u - d_u \|, \\
    \Upsilon_u(\tau) &= \nabla d f^d_u(\tau) + \Upsilon(\tau).
\end{align*}
\]

(21)

After computing the gradients and GPD parameters locally, each VUE \( u \) uploads its model at time \( \tau \), \((\Upsilon_u, d_u, K_u, Q_u)\), to the RSU as illustrated in Fig. 2 by process B.

The RSU will then perform model averaging over the network while calculating the global GPD parameters and gradients as per process C in Fig. 2:

\[
\begin{align*}
    d(\tau) &= d(\tau - 1) + \sum_{u} \kappa_u (\tau - 1) - d(\tau - 1)), \\
    \Upsilon(\tau) &= \frac{1}{\sum_{u} \kappa_u (\tau - 1)} \sum_{u} \Upsilon_u(\tau - 1).
\end{align*}
\]

(22)

Then, the global model \((\Upsilon, d, \sum_{u} K_u, \hat{Q})\) is shared with the network (process D in Fig. 2).

The evaluation and sharing of parameters at the VUEs and the RSU can be done in either a synchronous or asynchronous manner. In the synchronous approach, at the end of a predefined time interval \( T_f \gg w \), all VUEs evaluate their local gradients and simultaneously upload their local models \((\Upsilon_u, d_u, K_u, Q_u)\) to the RSU. Then the RSU averages out all the local models after which all VUEs download the global model \((\Upsilon, d, \sum_{u} K_u, \hat{Q})\). Here, synchronization may improve the accuracy of the estimation of global gradients. However, the simultaneous transmissions to the RSU by all VUEs degrades the VUE-RSU data rates and thus, introduces significant delays to ongoing V2V communication. This synchronous FL approach presented above is dubbed “sync-FL” hereinafter, and summarized in Algorithm 2.

In contrast, in the asynchronous approach, each VUE must wait until a predefined number \( \hat{K} \) of new QSI samples are collected. In essence, at time \( t_u \) with \( K_u(t_u) / \hat{K} \in \mathbb{N} \), VUE \( u \) evaluates and uploads its local model \((\Upsilon_u, d_u, K_u, Q_u)\) to the RSU. At the RSU, the newly received local model is averaged out with the existing local models of other VUEs, and the updated global model \((\Upsilon, d, \sum_{u} K_u, \hat{Q})\) is fed back to VUE \( u \). Note that the delay for the upload and download processes will be very small due to the fact that the likelihood of multiple VUEs simultaneously sharing their model is very low. We designate the asynchronous approach as “async-FL”, as seen in Algorithm 3.
Fig. 3. Illustration of QSI/model uploading and downloading for CEN, sync-FL, and async-FL. Switching between VUE-to-RSU (V2RSU) and V2V communications are shown at the top and bottom of the time axis, respectively.

Algorithm 2 MLE for GPD using sync-FL

1: **input:** Gradients \( \{T_u(t)\}_{u \in \mathcal{U}} \), local estimations \( \{d_u(t)\}_{u \in \mathcal{U}} \), and step size \( \delta \).
2: **for** \( t = 1, 2, \ldots \) **do** {in parallel}
3: **for** each VUE \( u \in \mathcal{U} \) **do** {in parallel}
4: **if** \( K_u(t) = K \in \mathbb{N} \) **then**
5: **set:** \( T_u(t) = 0 \), \( d_u(t) = d(t) \) and \( \delta_u = \delta/K_u(t) \).
6: **else**
7: Let \( i_k \) be a random permutation of \( Q_u \).
8: **for** \( k = 1, \ldots, K_u(t) \) **do**
9: **end for**
10: Evaluate \( T_u(t) \) and \( d_u(t) \) using (21).
11: **end if**
12: **end for**
13: **end for**
14: **if** \( \hat{U}(t) \neq \emptyset \) **then**
15: Update \( d(t) \) and \( \hat{T} \) using (22).
16: **end if**
17: **end for**

Algorithm 3 MLE for GPD using async-FL

1: **input:** Gradients \( \{T_u(t)\}_{u \in \mathcal{U}} \), local estimations \( \{d_u(t)\}_{u \in \mathcal{U}} \), and step size \( \delta \).
2: **for** \( t = 1, 2, \ldots \) **do** {in parallel}
3: **for** each VUE \( u \in \mathcal{U} \) **do** {in parallel}
4: **if** \( K_u(t) = K \in \mathbb{N} \) **then**
5: **set:** \( t_u = t \), \( T_u(t_u) = 0 \), \( d_u(t_u) = \hat{d}(t) \) and \( \delta_u = \delta/K_u(t_u) \).
6: **else**
7: Let \( i_k \) be a random permutation of \( Q_u \).
8: **for** \( k = 1, \ldots, K_u(t_u) \) **do**
9: **end for**
10: Evaluate \( T_u(t_u) \) and \( d_u(t_u) \) using (21).
11: **end if**
12: **end for**
13: **end for**
14: RSU observes the set \( \hat{U}(t) \) of VUEs that upload models at time \( t \).
15: **if** \( \hat{U}(t) \neq \emptyset \) **then**
16: Update \( d(t) \) and \( \hat{T} \) using (22).
17: **end if**
18: **end for**

D. Cost of Communication with the RSU

In all three methods used to estimate the GPD parameters, CEN, sync-FL, and async-FL, the QSI samples or local/global models are exchanged between VUEs and RSU by reusing the RBs available for V2V communication. This communication between VUEs and the RSU for GPD parameter estimation introduces additional latencies to the ongoing V2V communication. Such latencies from GPD parameter learning can be seen as additional costs for URLLC applications. In this regard, modeling the cost of uploading/downloading the learning models or queue samples in terms of an additional delay on V2V communication is illustrated in Fig. 3 and discussed next.

Let \( J_T, J_d, \) and \( J_Q \) be the sizes (in bits) of gradient values, GPD parameters, and queue samples of any VUE, respectively. Suppose VUE \( u \) has \( K_u \) new samples, and its uplink and downlink rate between RSU are \( r_{u0} \) and \( r_{0u} \), respectively. In the CEN approach, VUE \( u \) dedicates \( J_Q(K_u/r_{u0}) \) time to upload all its new queue samples to the RSU while \( J_d/r_{0u} \) time to download the GPD parameters from the RSU. Since all VUEs access the RSU simultaneously, the RSU will schedule VUEs over the RBs that are already allocated for their V2V communication links. As a result, an additional delay of \( J_Q(K_u/r_{u0} + J_d/r_{0u}) \) is introduced for \( u \)'s V2V communication.

Similar to CEN, in sync-FL, the RSU schedules VUEs due to their simultaneous access to the RSU. However, in sync-FL, only the learning models are shared. Therefore, the corresponding uplink and downlink durations \( (J_T + J_d + J_Q)/r_{u0} \) and \( (J_T + J_d)/r_{0u} \) are introduced as additional delays for VUE \( u \)'s V2V communication. Similar delays can be observed for async-FL approach. However, in async-FL, VUEs independently access the RSU in which lower interference on VUE-RSU communication links compared to sync-FL can be expected. Therefore, higher rates for \( r_{u0} \) and \( r_{0u} \), and lower delays on V2V communication can be expected in async-FL compared to the other two methods.

V. SIMULATION RESULTS AND ANALYSIS

For our simulations, we consider a network based on a 250 m x 250 m Manhattan mobility model with nine intersections. In this setting, a road consists of two lanes with 4 m width in each direction. We uniformly deploy VUE pairs within each lane with the vRx always following the vTx with a speed of 60 km/h and a fixed gap of 50 m. VUEs share 60 RBs and have a maximum transmit power of \( p_0 = 10 \) W. The RB allocation per group is adopted from [6] and [4]. The traffic generation at each VUE transmitter follows a Poisson arrival
TABLE I
SIMULATION PARAMETERS [4], [6], [18], [31].

<table>
<thead>
<tr>
<th>Para.</th>
<th>Value</th>
<th>Para.</th>
<th>Value</th>
<th>Para.</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\ell$</td>
<td>-65.5 dBm</td>
<td>$N_0$</td>
<td>-174 dBm/Hz</td>
<td>$\omega$</td>
<td>10</td>
</tr>
<tr>
<td>$\ell'$</td>
<td>-54.5 dBm</td>
<td>$\delta$</td>
<td>(50,0.005)</td>
<td>$c$</td>
<td>1.61</td>
</tr>
<tr>
<td>$W$</td>
<td>180 kHz</td>
<td>$\nabla_d f_u^{Q}(0)$</td>
<td>(1,000)</td>
<td>$d_0$</td>
<td>15 m</td>
</tr>
<tr>
<td>$q_0$</td>
<td>16.26 kb</td>
<td>$d_u(0)$</td>
<td>(1.0)</td>
<td>$\epsilon$</td>
<td>0.001</td>
</tr>
</tbody>
</table>

Fig. 4 compares the accuracy of GPD parameter estimation numerically, we use MLE-based cost function in (17) and the corresponding results are illustrated in Fig. 4b. Furthermore, in Fig. 4b, the impact of number of iterations used in SVRGD on the accuracy of GPD parameter estimation in CEN and async-FL are observed. Here, the selected scenarios have about $2U$ number of queue length samples exceeding $q_0$ in which $U = 20$ has the lowest number of samples while $U = 100$ has the highest. When one iteration is used for SVRGD (at the RSU in CEN and at VUEs in async-FL), Fig. 4b shows that higher samples yielding lower cost, i.e., better accuracy of GPD parameter estimation. Therein, the cost of async-FL is about 66.6%, 47.1%, and 40.4% higher than of CEN for $U = 20$, 40, and 60, respectively. Increasing the number of iterations used in SVRGD reduces the cost rapidly at first, then the reductions are insignificant. When two iterations are used for SVRGD, FL yields lower cost and thus, a higher accuracy in parameter estimation compared to CEN.

CEN estimations. To evaluate the accuracy of GPD parameter estimation numerically, we use MLE-based cost function in (17) and the corresponding results are illustrated in Fig. 4b. Furthermore, in Fig. 4b, the impact of number of iterations used in SVRGD on the accuracy of GPD parameter estimation in CEN and async-FL are observed. Here, the selected scenarios have about $2U$ number of queue length samples exceeding $q_0$ in which $U = 20$ has the lowest number of samples while $U = 100$ has the highest. When one iteration is used for SVRGD (at the RSU in CEN and at VUEs in async-FL), Fig. 4b shows that higher samples yielding lower cost, i.e., better accuracy of GPD parameter estimation. Therein, the cost of async-FL is about 66.6%, 47.1%, and 40.4% higher than of CEN for $U = 20$, 40, and 60, respectively. Increasing the number of iterations used in SVRGD reduces the cost rapidly at first, then the reductions are insignificant. When two iterations are used for SVRGD, FL yields lower cost and thus, a higher accuracy in parameter estimation compared to CEN.

In Fig. 5, we compare the amount of data exchange and the achieved reliability in terms of maintaining the queue length below $q_0$ for different VUE densities. As the reliability decreases with increasing the number of VUEs, async-FL achieves a reliability that is slightly lower to the one resulting from the CEN approach for $U < 72$, while outperforms CEN when $U > 72$. Note that the CEN method requires all VUEs to upload all their queue length samples to the RSU and to receive the estimated GPD parameters. In contrast, in async-FL, VUEs upload their locally estimated learning model $(\mathbf{T}_u, d_u, K_u, Q_u)$ and receive the global estimation of the model. For fewer number of VUEs, $U = 20$, the sample size of the network is small, and, thus, CEN can operate efficiently using very few data samples. In contrast, in async-

**A. Centralized vs distributed GPD parameter estimation**

Fig. 4 compares the accuracy of GPD parameter estimation using CEN and async-FL. In Fig. 4a, the estimated GPDs for CEN and async-FL approaches for $U = 20$, 60, and 100 are shown. Here, the original samples are plotted along the estimated distributions. From Fig. 4a it can be noted that the estimations of async-FL are almost equivalent to the
the transmit power of vTx in FP is chosen as the average of transmit powers from all other five methods. The baseline QSO, which is oblivious to reliability, consumes a minimum transmit power out of all other methods. QSR baseline takes into account reliability while neglecting VUEs with extreme queue lengths, exhibits lower power consumption compared to all three proposed approaches for total VUEs $U < 40$. For the cases with $U > 40$, QSR consumes higher power compared to async-FL on average, and beyond $U = 80$, it is the most power consuming method. In QSR, there is no control on the number of VUEs with extreme queue lengths that increases with $U$, and thus, their power consumption degrades the performance of QSR. Both CEN and sync-FL methods exhibit almost equal average power consumption while async-FL uses less transmission power compared to CEN and sync-FL. The requirement of lower transmit power to upload/download learning models due to asynchronous communication between the RSU and VUEs in async-FL results in the power reductions therein. The power reductions in async-FL compared to CEN and sync-FL are negligible for $U = 20$, improves up to $31.6\%$ when $U = 42$, and remains around $35\%$ for $U > 42$.

The average queue length as a function of total VUE pairs for all baseline and proposed methods are shown in Fig. 6b. Here, the average queue length reflects the average queuing latency. In FP, due to the low and fixed transmit power, the VUE queues grow large even for few VUE pairs. Since the fixed power is increased with $U$ as shown in Fig. 6a, the average queue length decreases with $U$ first, then rises again due to the increased interference of the network. Although QSO has the lowest power consumption, it yields higher queuing latency compared to all other methods except FP. All three proposed techniques exhibit similar queue lengths on average while QSR results in the lowest average queuing latency. Compared to QSR, all three proposed methods that control VUEs with extreme queue lengths suffer up to three times in average queuing latency when $U = 100$.

Fig. 7a plots the maximum queue length that is proportional to the worst-case latency observed for all methods as a function of the total number of VUE pairs. Similar to average queue lengths, FP and QSO exhibit the highest worst-case latencies. QSR which has the lowest average queue lengths displays higher worst-case queue lengths compared to CEN and sync-FL for $U \leq 68$ while it fails to outperform async-FL for all $U \leq 100$. Although QSR limits the fraction of VUE queue lengths exceeding $q_0$ and provides the best average latency, Fig. 7a shows that QSR neglects VUEs with extreme large queue lengths (worst-case VUEs). All three proposed methods CEN, sync-FL, and async-FL that have control over the tail distribution of the queue lengths yield almost equal worst-case queuing latencies up to $U = 68$. The reductions in worst-case latencies for all proposed methods are about $40.3\%$ for $U = 20$ and about $17.3\%$ for $U = 68$ compared to QSR. Further increasing $U$ increases the number of queue length samples exceeding $q_0$ in which frequent communications between VUEs and RSU take place. As a result, the learning procedure imposes undesirable delays on V2V communication in which high worst-case latencies can be observed in the proposed methods. However, due to the asynchronous nature...
of async-FL, VUEs communicate with the RSU independently in which the delay imposed by model sharing is reduced in async-FL compared to CEN and sync-FL. Hence, async-FL yields 7.5%, 21.7%, and 29% reductions in worst-case latencies compared to QSR, CEN, and sync-FL, respectively, for \( U = 100 \).

The reliability in terms of the probability that the queue lengths are maintained below \( q_0 \) for all methods is presented in Fig. 7b as a function of the total number of VUE pairs. It can be noted that FP and QSO, which have no interest in improving V2V communication reliability are the first and second most unreliable methods, respectively. Since QSR has a reliability constraint, it yields greatly improved reliability over FP and QSO. CEN, sync-FL, and async-FL control the tail distribution of queue lengths along with the reliability constraint and thus, exhibit further improvements in reliability, i.e., outage reductions, compared to QSR. Similar to the explanation of the behavior of maximum queue lengths, asynchronous model sharing in async-FL reduces the delays introduced by the RSU-VUE communications compared to CEN and sync-FL methods. As a result, for \( U \geq 84 \), lower queue lengths and thus, reduced outages in async-FL method over CEN and sync-FL can be observed in Fig. 7b.

The reductions in outages (or reliability gains) of async-FL compared to QSR are 84.6% and 18.8% for \( U = 20 \) and 76, respectively. At \( U = 100 \), async-FL yields about 60.9%, 36%, and 35.9% reductions in outages compared to QSR, CEN, and sync-FL, respectively.

Fig. 8 illustrates the mean and standard deviation of the distribution of queue length exceeding \( q_0 \), versus number of VUE pairs.

Fig. 9. The impact of vTx-vRx distances on queue lengths and transmit powers.
queue length tail distributions of QSR, CEN, sync-FL, and async-FL methods for different number of VUE pairs. The standard deviation at a given $U$ is drawn on top of the corresponding mean value to clearly highlight the fluctuations of queue lengths above $q_0$. Note that FP and QSO are neglected since they have large means and standard deviations that do not scale well with the other four methods. CEN exhibits the lowest means and standard deviations of extreme queue lengths up to $U = 44$. For $U > 44$ async-FL displays the lowest mean and fluctuations of queue lengths exceeding $q_0$ proving to be the best candidate for URLLC with large number of VUE pairs. The reductions in average extreme queue lengths in async-FL are about 28.6%, 41.9%, and 19.5% compared to QSR, CEN, and sync-FL methods. From Fig. 8, we can see that QSR has the highest averages of extreme queue lengths compared to CEN, sync-FL, and async-FL methods. However, the fluctuations of queue lengths above $q_0$ are high in QSR only for $U \leq 68$. Beyond $U = 68$, highest fluctuations in extreme queue lengths are seen in both CEN and sync-FL. At $U = 100$, the fluctuation reductions in async-FL are about 33.2%, 38%, and 47.1% compared to QSR, CEN, and sync-FL methods.

The queue length CCDF and transmit power cumulative density function (CDF) of QSR and async-FL for different vTx-vRx distances are shown in Fig. 9. According to Fig. 9a, as vTx-vRx distance increases, the queue lengths increase in both QSR and async-FL methods due to the reduced over-the-air data rates. QSR which essentially neglects the queue lengths exceeding $q_0$ exhibits longer tails compared to the tail distribution-aware async-FL method. In async-FL, reductions of average queue lengths are 1.2% and 29.4% compared to QSR for vTx-vRx distances of 20 m and 80 m, respectively. For 50 m, QSR yields a reduction of 67% in the average queue length over async-FL. In terms of reductions of the worst-case queue lengths, VUEs with queue lengths exceeding $q_0$, in async-FL compared to QSR are 1.1%, 5.2%, and 5.8% for vTx-vRx distances of 20 m, 50 m, and 80 m, respectively. Fig. 9b shows that both QSR and async-FL methods consume lower power for the networks with close vTxS and their corresponding vRxs. For larger vTx-vRx distances, vTxS need higher transmit powers to serve their vRxs, yielding increased transmit powers in both QSR and async-FL methods. async-FL utilizes the characteristics of the queue length tail distribution to reduce the number of VUEs with large queue lengths and thus, minimizes the communications that need high data rates to meet the target reliability. In contrast, QSR has no control on the queue lengths exceeding $q_0$, and thus, requires high data rates to serve VUEs with extreme large queue lengths yielding higher transmit power consumption compared to async-FL. Fig. 9b shows that async-FL method reduces the transmit power consumption on average by 3.2%, 18.2%, and 43.1% for vTx-vRx distances of 20 m, 50 m, and 80 m, respectively, compared to QSR method.

C. Impact of finite block length

It is worth mentioning that the high mobility in V2V communication networks limits the channel coherence time and thus, the codeword length (or block length) in each transmission. The works in [39]–[41] analyze the URLLC in the finite block length regime and propose resource management solutions therein. In this regard, we adopt the approximated achievable rate derived for the finite block length regime from [41], [42] yielding $r_u \approx W \left( \log_2(1 + \gamma_u) - \frac{2^{2\gamma_u} - 1}{2} \sqrt{D(\gamma_u + 1)} \right)$ where $\gamma_u$, $D$, and $\varepsilon$ are the signal-to-interference and noise ratio, the block length, and the error probability, respectively. Here, the block length $D$ is proportional to the coherence time, and is inversely proportional to the VUE speed [43]. Thus, we consider a total of $U = 40$ VUEs with three speeds, $40, 60, 80$ km/h, that correspond to $D = \{800, 534, 400\}$ bits. The choice of $\varepsilon = .5$ represents the original formulation, though the differences in VUE speed are reflected in the channel coherence time, which impacts latency compared with $\varepsilon = \{.1, .01\}$. Fig. 10 shows the average queue lengths, maximum queue length, average transmit power, and outages in terms of queue lengths exceeding the threshold $q_0$ (unreliability) for the async-FL method. For $\varepsilon = .5$, which corresponds to the formulation in Section III, Figs. 10b and 10d illustrate that both the worst-case queuing latency and the queue outages increase with the VUE speed. This is a result of the lower achievable rates due to the smaller coherence times and shorter block lengths at higher speeds. However, higher power consumption yields lower queuing latency on average as exhibited in Fig. 10a. Furthermore, Fig. 10 shows that the
performance of the system degrades with decreasing $\varepsilon$.

VI. CONCLUSIONS

In this paper, we have formulated the problem of joint power control and resource allocation for V2V communication network as a network-wide power minimization problem subject to ultra reliability and low latency constraints. The constraints in terms of URLLC are characterized using extreme value theory and modeled as the tail distribution of the network-wide queue lengths over a predefined threshold. Leveraging concepts of federated learning, a distributed learning mechanism is proposed where VUEs estimate the tail distribution locally with the assistance of a RSU. Here, FL enables VUEs to learn the tail distribution of the network-wide queues locally without sharing the actual queue length samples reducing unnecessary overheads. Combining both EVT and FL approaches, we have proposed a Lyapunov-based distributed transmit power and resource allocation procedure for VUEs. Using simulations, we have shown that the proposed method learns the statistics of the network-wide queues with high accuracy. Furthermore, the proposed method shows considerable gains in reducing extreme events where the queue lengths grow beyond a predefined threshold compared to systems that account for reliability by imposing probabilistic constraints on the average queue lengths. It is worth noting that the accuracy of FL-based tail distribution modeling relies on the assumption of IID queue length samples. In this regard, a future extension of this work is to study FL in the presence of non-IID training data.

APPENDIX A

PROOF OF PROPOSITION 1

First, consider the one-slot drift of the Lyapunov function $\Delta L_{t} = L(\mathbf{E}(t+1)) - L(\mathbf{E}(t))$.

$$\Delta L_{t} = \sum_{u \in \mathcal{U}} \left( \frac{q_{u}^{2}(t+1)-q_{u}^{2}(t)}{2} + \frac{\Psi_{1}^{2}(t+1)-\Psi_{1}^{2}(t)}{2} \right) + \frac{\Delta_{1}^{2}(t+1)-\Delta_{1}^{2}(t)}{2} + \frac{\Delta_{2}^{2}(t+1)-\Delta_{2}^{2}(t)}{2}.$$  \hfill (23)

Using the relation $(q+\alpha-r)^{2} \leq q^{2}+(\alpha-r)^{2}+2q(\alpha-r)$ for the $(t+1)$-th terms in (23), upper bounds for each of the above terms can be derived as in (24). Here, $M_{1} = q_{0} + \sigma/(1 - \xi)$ and $M_{2} = 2\sigma^{2}/(1 - \xi)(1 - 2\xi)$. Furthermore, along the queue exceeding indicator $\mathbf{1}_{t}$, the definition $q_{u}(t+1) = q_{u}(t) + a_{u}(t) - r_{u}(t)$ is used instead of (3) due to the fact that $\mathbf{1}_{t} = 1$ ensures nonempty queues, i.e. $\mathbf{1}_{t} = 1 \implies q_{u}(t+1) = [q_{u}(t) + a_{u}(t) - r_{u}(t)]^{+} = q_{u}(t) + a_{u}(t) - r_{u}(t) > 0$.

Note that the terms $#a_{1} \#a_{3}$ are quadratic in which the assumption of queue stability forces them to be bounded. Hence sum of $#a_{1} \#a_{3}$ terms are replaced by a bounded value $\Delta_{0}$ given in (25). Terms $#b_{1}$ and $#b_{2}$ are independent from the control variables. Therefore, we denote them by $\Delta_{u}$ for VUE $u \in \mathcal{U}$ where

$\Delta_{u} = \Psi_{u}(q_{u}(t) - \epsilon q_{0}) - \epsilon q_{0} q_{u}(t)$

$+ (A_{u}(t) - M_{1}) q_{u}(t) - M_{1} A_{u}(t).$ \hfill (26)

Combining the results of (24)-(26) and applying them into (23) conclude the proof.

APPENDIX B

PROOF OF PROPOSITION 2

Let $g_{t}^{d}(Q) = (1 + \xi Q/\sigma)^{-1}/\xi$. Since $g_{t}^{d}(Q) \to e^{-Q/\sigma}$ as $\xi \to 0$, the distribution can be rewritten as $G_{\lambda}^{Q}(Q) = 1/\sigma g_{t}^{d}(Q)^{\xi+1}$. Using the above notation, it can be noted that

$$f_{t}^{d}(Q) = \sum_{Q \in \mathcal{Q}} \ln g_{t}^{d}(Q) = \frac{1}{|\mathcal{Q}|} \sum_{Q \in \mathcal{Q}} f_{t}^{d}(Q).$$  \hfill (27)

Hence, $\nabla f_{t}^{d}(Q) = \frac{|\mathcal{Q}|}{|\mathcal{Q}|} \sum_{Q \in \mathcal{Q}} \nabla f_{t}^{d}(Q)$ is held. First, the gradient of $g_{t}^{d}(Q)$ is found by

$$\nabla g_{t}^{d}(Q) = \left[ \frac{Q}{\sigma} g_{t}^{d}(Q)^{\xi+1} - g_{t}^{d}(Q) \ln g_{t}^{d}(Q) \right].$$  \hfill (28)

Thus, the gradient of $f_{t}^{d}(Q)$ can be calculated as follows:

$$\nabla f_{t}^{d}(Q) = \left[ \frac{1}{\sigma} - \frac{1}{\sigma} \frac{g_{t}^{d}(Q)}{\partial g_{t}^{d}(Q)} - \ln g_{t}^{d}(Q) \right]$$

$$\left[ \frac{1}{\sigma} - \frac{1+\xi/\sigma}{(1+\xi/\sigma)/2} \ln(1+\xi/\sigma) \right].$$

REFERENCES


\[
\begin{align}
\frac{q_u^2((t+1)/2-q_u^2(t))}{2} & \leq \left[ \frac{g_u(t)q_u(t)-r_u(t)}{2} \right]_{a_2} + \left[ q_u(t)(a_u(t)-r_u(t)) \right]_{a_2} \\
\frac{\Psi^2_u((t+1)/2-q^2(u(t))}{2} & \leq \left[ \frac{g_u(t)q_u(t)-r_u(t)}{2} \right]_{b_2} + \left[ q_u(t)(a_u(t)-r_u(t)) \right]_{b_2} \\
A^2_u((t+1)/2-A^2_u(t)) & \leq \left[ \frac{g_u(t)q_u(t)-r_u(t)}{2} \right]_{a_4} + \left[ q_u(t)(a_u(t)-r_u(t)) \right]_{a_4} \\
B^2_u((t+1)/2-B^2_u(t)) & \leq \left[ \frac{g_u(t)q_u(t)-r_u(t)}{2} \right]_{b_4} + \left[ q_u(t)(a_u(t)-r_u(t)) \right]_{b_4}
\end{align}
\]

\[
\Delta_0 \geq \sum_{u \in U} \left[ \frac{g_u(t)q_u(t)-r_u(t)}{2} \right]_{a_4} + \left[ \frac{g_u(t)q_u(t)-r_u(t)}{2} \right]_{b_4} + \left[ \frac{g_u(t)q_u(t)-r_u(t)}{2} \right]_{a_4} + \left[ \frac{g_u(t)q_u(t)-r_u(t)}{2} \right]_{b_4}
\]


**Sumudu Samarakoon** (S’08, M’18) received the B.Sc. degree (Hons.) in electronic and telecommunication engineering from the University of Moratuwa, Moratuwa, Sri Lanka, in 2009, the M.Eng. degree from the Asian Institute of Technology, Khlong Luang, Thailand, in 2011, and the Ph.D. degree in communication engineering from the University of Oulu, Oulu, Finland in 2017. He is currently with the Intelligent Connectivity and Networks/Systems Group (ICON) in the Centre for Wireless Communications (CWC), University of Oulu, as a Postdoctoral Researcher. His current research interests include heterogeneous networks, small cells, radio resource management, reinforcement learning, and game theory. Dr. Samarakoon received the Best Paper Award at the European Wireless Conference and the Excellence Awards for innovators and the outstanding doctoral student at the Radio Technology Unit, CWC, University of Oulu, in 2016.

**Mehdi Bennis** (S’07, AM’08, SM’15) is currently an Associate Professor with the Centre for Wireless Communications, University of Oulu, Oulu, Finland, where he is also an Academy of Finland Research Fellow and the Head of the Intelligent Connectivity and Networks/Systems Group (ICON). He has authored one book and published more than 200 research articles in international conferences, journals, and book chapters. His current research interests include radio resource management, heterogeneous networks, game theory, and machine learning in 5G networks and beyond. Dr. Bennis was a recipient of several prestigious awards, including the 2015 Fred W. Ellersick Prize from the IEEE Communications Society, the 2016 Best Tutorial Prize from the IEEE Communications Society, the 2017 EURASIP Best Paper Award for the Journal on Wireless Communications and Networks, the All-University of Oulu Award for research, and the 2019 IEEE ComSoc Radio Communications Committee Early Achievement Award. He is an Editor of the IEEE Transactions on Communications.

**Walid Saad** (S’07, M’10, SM’15, F’19) received his Ph.D degree from the University of Oslo in 2010. He is currently a Professor at the Department of Electrical and Computer Engineering at Virginia Tech, where he leads the Network science, Wireless, and Security (NEWS) laboratory. His research interests include wireless networks, machine learning, game theory, security, unmanned aerial vehicles, cyber-physical systems, and network science. Dr. Saad is a Fellow of the IEEE and an IEEE Distinguished Lecturer. He is also the recipient of the NSF CAREER award in 2013, the AFSOR summer fellowship in 2014, and the Young Investigator Award from the Office of Naval Research (ONR) in 2015. He was the author/co-author of eight conference best paper awards at WiOpt in 2009, ICIMP in 2010, IEEE WCNC in 2012, IEEE PIMRC in 2015, IEEE SmartGridComm in 2015, EuCNC in 2017, IEEE GLOBECOM in 2018, and IFIP NTMS in 2019. He is the recipient of the 2015 Fred W. Ellersick Prize from the IEEE Communications Society, of the 2017 IEEE ComSoc Best Young Professional in Academia award, and of the 2018 IEEE ComSoc Radio Communications Committee Early Achievement Award. From 2015-2017, Dr. Saad was named the Stephen O. Lane Junior Faculty Fellow at Virginia Tech and, in 2017, he was named College of Engineering Faculty Fellow. He received the Dean’s award for Research Excellence from Virginia Tech in 2019. He currently serves as an editor for the IEEE Transactions on Wireless Communications, IEEE Transactions on Mobile Computing, IEEE Transactions on Cognitive Communications and Networking, and IEEE Transactions on Information Forensics and Security. He is an Editor-at-Large for the IEEE Transactions on Communications.

**Mérouane Debbah** (S’01, M’04, SM’08, F’15) received the M.Sc. and Ph.D. degrees from the Ecole Normale Supérieure Paris-Saclay, France. He was with Motorola Labs, Saclay, France, from 1999 to 2002, and also with the Vienna Research Center for Telecommunications, Vienna, Austria, until 2003. From 2003 to 2007, he was an Assistant Professor with the Mobile Communications Department, Institut Eurecom, Sophia Antipolis, France. From 2007 to 2014, he was the Director of the Alcatel-Lucent Chair on Flexible Radio. Since 2007, he has been a Full Professor with CentraleSupélec, Gif-sur-Yvette, France. Since 2014, he has been a Vice-President of the Huawei France Research Center and the Director of the Mathematical and Algorithmic Sciences Lab. He has managed 8 EU projects and more than 24 national and international projects. His research interests lie in fundamental mathematics, algorithms, statistics, information, and communication sciences research. He is an IEEE Fellow, a WWRF Fellow, and a Membre émérite IEEE. He was a recipient of the ERC Grant MORE (Advanced Mathematical Tools for Complex Network Engineering) from 2012 to 2017. He was a recipient of the Mario Boella Award in 2005, the IEEE Glavieux Prize Award in 2011, and the Qualcomm Innovation Prize Award in 2012. He received 20 best paper awards, among which the 2007 IEEE GLOBECOM Best Paper Award, the Wi-Opt 2009 Best Paper Award, the 2010 Newcom++ Best Paper Award, the WUN ComC Best Paper 2012 and 2013 Award, the 2014 WCNC Best Paper Award, the 2015 ICC Best Paper Award, the 2015 IEEE Communications Society Leonard G. Abraham Prize, the 2015 IEEE Communications Society Fred W. Ellersick Prize, the 2016 IEEE Communications Society Best Tutorial Paper Award, the 2016 European Wireless Best Paper Award, the 2017 EURasip Best Paper Award, the 2018 IEEE Marconi Prize Paper Award, the 2019 IEEE Communications Society Young Author Best Paper Award and the Valuetoools 2007, Valuetoools 2008, CrownCom 2009, Valuetoools 2012, SAM 2014, and 2017 IEEE Sweden VT-COM-IT Joint Chapter best student paper awards. He is an Associate Editor-in-Chief of the journal Random Matrix: Theory and Applications. He was an Associate Area Editor and Senior Area Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2011 to 2013 and from 2013 to 2014, respectively.