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

Sumudu Samarakoon[*], Mehdi Bennis[*], Walid Saad[†], and Mérouane Debbah[‡]

[*] Centre for Wireless Communication, University of Oulu, Finland, email:
{sumudu.samarakoon,mehdi.bennis}@oulu.fi

[†]Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA,
email: walids@vt.edu

[‡] Mathematical and Algorithmic Sciences Lab, Huawei France R&D, Paris, France, (email:
merouane.debbah@huawei.com)

arXiv:1807.08127v2 [cs.IT] 1 Aug 2018

Abstract

In this paper, the problem of joint power and resource allocation for ultra reliable low latency communication (URLLC) in vehicular networks is studied. The key goal is to minimize the network-wide power consumption of vehicular users (VUEs) subject to high reliability in terms of probabilistic queuing delays. In particular, using extreme value theory (EVT), a new reliability measure is defined to characterize extreme events pertaining to vehicles' queue lengths exceeding a predefined threshold with non-negligible probability. In order to learn these extreme events in a dynamic vehicular network, a novel distributed approach based on federated learning (FL) is proposed to estimate the tail distribution of the queues. Taking into account the communication delays incurred by FL over wireless links, Lyapunov optimization is used to derive the joint transmit power and resource allocation 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 distributed method to estimate the tail distribution of queues with an accuracy that is very close to a centralized solution with up to 79% reductions in the amount of data that need to be exchanged. 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. For the VUEs with large queue lengths, the proposed method reduces their average queue lengths and fluctuations therein by about 30% compared to the aforementioned baseline.

Index Terms

V2V communication, Lyapunov optimization, extreme value theory, federated learning, URLLC, 5G.

I. INTRODUCTION

Providing efficient vehicle-to-vehicle (V2V) communications is a necessary stepping stone for enabling autonomous and intelligent transportation systems (ITS) [1]–[5]. 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 [1], [4]. 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 [1]. 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 [5]–[7]. 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 [8], [9].

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 [3]–[5], [8]–[17]. In [4], 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 [5] minimizes the total transmission power in a vehicular network while considering queuing latency and reliability. In [8], an energy-efficient resource allocation algorithm is proposed for cooperative V2V communication systems. The work in [9] proposed an energy saving sleep mode strategy for access points serving motorway vehicular traffic. The problem of vehicle network clustering is studied in [10] to reduce the power consumption of V2V communications. In [11], a joint resource allocation and power control algorithm is proposed to maximize the V2V sum rate. The authors in [12] optimize

the beam alignment and scheduling among vehicles to reduce the V2V communication delays. In [13], the tradeoff between service delay and transmission success in V2V communications is optimized. The URLLC aspects of this prior art [4], [5], [8]–[14] 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 [3], [18]–[21].

In practice, to enable a truly URLLC experience, it is imperative 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 [22]. In [19], 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 [20] studies the statistical distributions of inter-beacon delays in safety applications for vehicular adhoc networks (VANETs) using EVT. The authors in [21] 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 [3] 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 [23], [24]. 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 [23], 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 [25]. In [24], FL is used to develop distributed learning models for multiple related tasks simultaneously, referred to as multi-task learning. The recent work in [26], proposes a new FL protocol that solves a client selection problem with resource constraints in mobile edge computing. Our prior work in [17] 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 [17], 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) [22]. 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. 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 $\mathcal{U}$ of U communicating VUE pairs, using an RSU that allocates a set $\mathcal{N}$ of resource blocks (RBs) over a partition of the network $\mathcal{Z}$ defined as zones. Here, a *zone* 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 zones to reduce the interference among nearby VUE pairs. Hence, a VUE pair u is only allowed to use the subset $\mathcal{N}_{z(t,u)} \subseteq \mathcal{N}$ of RBs allocated to its corresponding zone $z(t,u)$ at time t. We denote the VUE
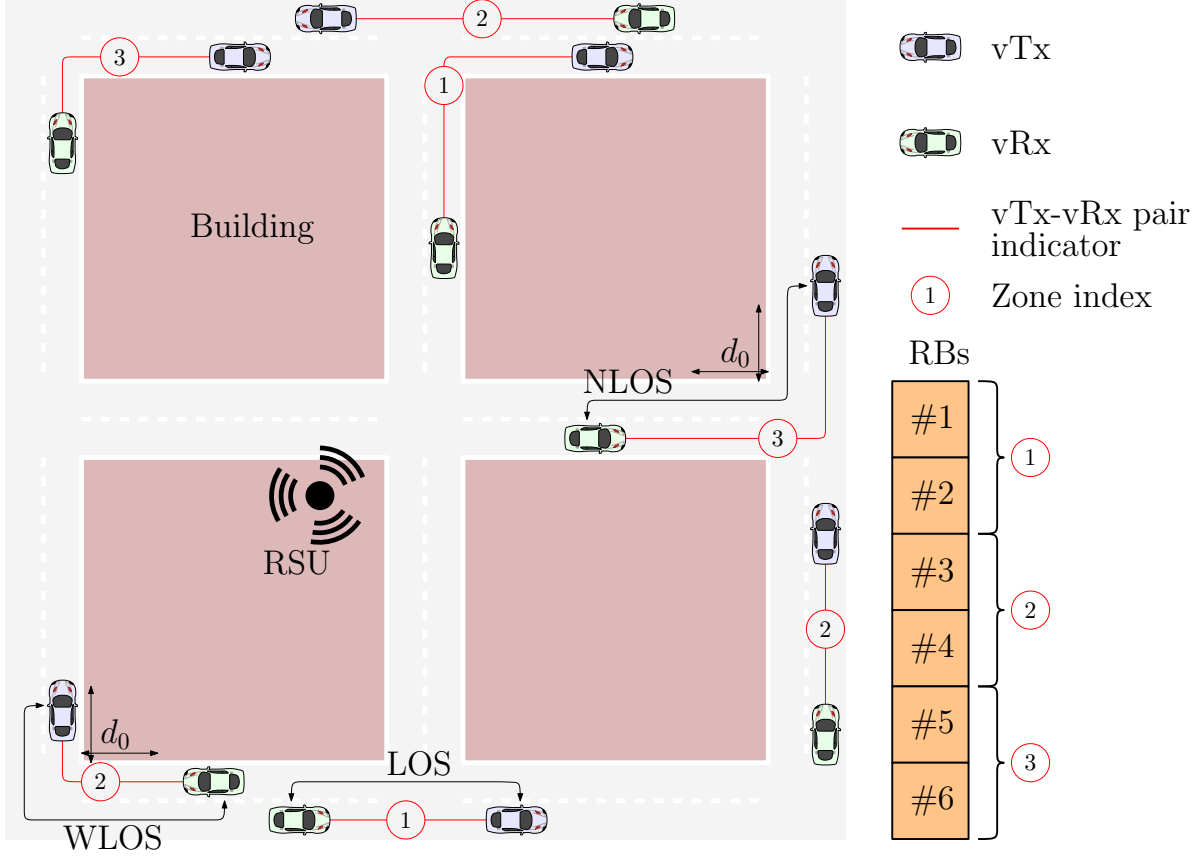
Fig. 1. Simplified illustration of the system model containing vTx-vRx pairs within their zone indexes and RB allocation over zones.

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 $\boldsymbol{p}_u(t) = [p_u^n(t)]_{n \in \mathcal{N}_{z(t,u)}}$ and $\boldsymbol{h}_{uu'}(t) = [h_{uu'}^n(t)]_{n \in \mathcal{N}_{z(t,u)}}$ 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 and a path loss $\Phi_{uu'}$ that uses the following model for urban areas using 5.9 GHz carrier frequency [27]:

$$\Phi_{uu'} = \begin{cases} \ell\|(x_u, y_u) - (x_{u'}, y_{u'})\|_2^{-c} & \text{for LOS,} \\ \ell\|(x_u, y_u) - (x_{u'}, y_{u'})\|_1^{-c} & \text{for WLOS,} \\ \ell'(|x_u - x_{u'}| \cdot |y_u - y_{u'}|)^{-c} & \text{for NLOS,} \end{cases} \quad (1)$$

where $\|\boldsymbol{x}\|_l$ is the l-th norm of vector $\boldsymbol{x}$, c is the path loss exponent, and the path loss coefficients ℓ and ℓ' satisfy $\ell' < \ell(\frac{d_0}{2})^c$. The transmission rate between the vTx-vRx pair u is given by,

$$r_u(t) = \sum_{n \in \mathcal{N}_{z(t,u)}} r_u^n(t) = \sum_n W \log_2 \left(1 + \frac{h_{uu}^n(t)p_u^n(t)}{I_u^n(t) + WN_0}\right), \quad (2)$$

where $I_u^n(t) = \sum_{u' \in \mathcal{U}\setminus\{u\}} h_{u'u}^n(t)p_{u'}^n(t)$ 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 $\overline{a}_u$ 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:

$$q_u(t + 1) = [q_u(t) + a_u(t) - r_u(t)]^+, \quad (3)$$

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. 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 ϵ. The reliability conditions can now be formally defined as:

$$\mathbb{E}[q_u] = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} q_u(t) < \infty \qquad \forall u \in \mathcal{U}, \quad (4)$$

$$\Pr(q_u(t) \geq q_0) \leq \epsilon \qquad \forall u \in \mathcal{U}, \forall t. \quad (5)$$

Note that the above reliability constraints cannot cope with extreme cases where $q_u(t) > q_0$. Such extreme cases essentially correspond to the worst case network queuing latency (as well as end-to-end latency [5]–[7]) which are a key determinant of the URLLC performance and, hence, must be properly addressed. In this regard, let $M(t)$ be a sample of queue length exceeding

the threshold q_0 observed over the network at time t and $M \in \{M(t)\}_{\forall t}$. By imposing the constraints,

$$\lim_{T \to \infty} \sum_{t=1}^{T} \left(q_u(t) - q_0 \right) \mathbb{1}_{q_u(t)} / \sum_{t=1}^{T} \mathbb{1}_{q_u(t)} \leq \mathbb{E}\left[M\right], \tag{6}$$

$$\lim_{T \to \infty} \sum_{t=1}^{T} \left(q_u(t) - q_0 \right)^2 \mathbb{1}_{q_u(t)} / \sum_{t=1}^{T} \mathbb{1}_{q_u(t)} \leq \mathbb{E}\left[M^2\right], \tag{7}$$

each VUE u can better control the fluctuations of its queue and maintain its extreme values below the desired threshold. Here, $\mathbb{1}_x$ is an indicator function with $\mathbb{1}_x = 1$ when $x > q_0$, and $\mathbb{1}_x = 0$, otherwise. We can now formally pose our network-wide power minimization problem:

$$\min_{[\boldsymbol{p}_u(t)]_{\forall u \in \mathcal{U}}^{\forall t}} \quad \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{u \in \mathcal{U}} \mathbf{1}^{\dagger} \boldsymbol{p}_u(t) \tag{8a}$$

$$\text{s.t.} \quad (3)\text{-}(7), \tag{8b}$$

$$\boldsymbol{p}_u(t) \succcurlyeq \boldsymbol{0}, \quad \mathbf{1}^{\dagger} \boldsymbol{p}_u(t) \leq p_0 \quad \forall u \in \mathcal{U}. \tag{8c}$$

Here, (8b) ensures queue dynamics and reliability while controlling the worst-case latency over all VUEs and p_0 is the transmit power budget of a VUE. Solving (8) to obtain the optimal transmission control policy over time is challenging due to two reasons: *i)* A decision at time t relies on future network states, and *ii)* The characteristics of the distribution of M for constraint (6) are unavailable. Moreover, solving (8) using a centralized approach requires exchanging channel state information (CSI) and QSI over the whole network resulting in unacceptable signaling overheads. Therefore, a distributed solution that requires minimal coordination within the vehicular network is needed.

III. PROPOSED DISTRIBUTED FRAMEWORK USING EVT AND LYAPUNOV OPTIMIZATION

Developing a distributed solution for solving (8) requires decoupling the optimization problem over VUE pairs. Therefore, next, we propose new solutions to decouple the objective function (8a) and the constraints (6) and (7) based on the statistics of queue lengths exceeding q_0 over the vehicular network.

A. Modeling Extreme Queue Lengths Using Extreme Value Theory

The samples of queue lengths exceeding the threshold $\{M(t)\}_{\forall t}$ are seen as extreme statistics of the system, and can be characterized using EVT. Assume that the individual queues at a given time $[q_u(t)]_{u \in \mathcal{U}}$ are samples of independent and identical distributions (i.i.d.) and the queue

threshold q_0 is large. Then, the distribution of M can be modeled as a GPD using [22, Theorem 3.2.5]. This fundamental EVT result mainly shows that, as $q_0 \to \sup\{q | \Pr(M(t) > q) > 0\}$, the conditional probability distribution of $M(t) = q(t) - q_0$ is given by,

$$G_M^{\boldsymbol{d}}(m) = \begin{cases} \frac{1}{\sigma}(1 + \xi m/\sigma)^{-1-1/\xi} & \text{for } \xi \neq 0, \\ \frac{1}{\sigma}e^{-m/\sigma} & \text{for } \xi = 0, \end{cases} \tag{9}$$

with $\boldsymbol{d} = [\sigma, \xi]$, and ξ and $\sigma(> 0)$ are called the shape and scale parameters, respectively. Here, $m \geq 0$ if $\xi \geq 0$ while $0 \leq m \leq -\sigma/\xi$ when $\xi < 0$. Moreover, $\mathbb{E}[M]$ and $\mathbb{E}[M^2]$ are bounded and equivalent to $\sigma/(1-\xi)$ and $2\sigma^2/(1-\xi)(1-2\xi)$, respectively, only if $\xi < 1/2$. In this regard, constraints (6) and (7) for all $u \in \mathcal{U}$ can be rewritten as follows:

$$\lim_{T \to \infty} \sum_{t=1}^{T} \left(q_u(t) - q_0 \right) \mathbb{1}_{q_u(t)} / \sum_{t=1}^{T} \mathbb{1}_{q_u(t)} \leq \sigma/(1-\xi), \tag{10}$$

$$\lim_{T \to \infty} \sum_{t=1}^{T} \left(q_u(t) - q_0 \right)^2 \mathbb{1}_{q_u(t)} / \sum_{t=1}^{T} \mathbb{1}_{q_u(t)} \leq 2\sigma^2/(1-\xi)(1-2\xi). \tag{11}$$

Assisted by the RSU, each VUE pair can estimate ξ and σ 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:

$$\underset{[\boldsymbol{p}_u(t)]_{\forall u \in \mathcal{U}}^{\forall t}}{\text{minimize}} \quad \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \left(\sum_{u \in \mathcal{U}} \mathbf{1}^\dagger \boldsymbol{p}_u(t) \right) \tag{12a}$$

$$\text{subject to} \quad (3)\text{-}(5), (8c), (10), (11). \tag{12b}$$

To devise a tractable solution for the modified stochastic optimization problem in (12), we resort to Lyapunov optimization [28]. 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 $\mathbb{E}[\mathbb{1}_{q_u}] \leq \epsilon$ for each VUE $u \in \mathcal{U}$. Our next goal is to introduce a virtual queue Ψ_u for the aforementioned constraint instead of (4) and (5). To maintain the virtual queue Ψ_u's order of magnitude close to the order of magnitude of the actual queue size q_u, both sides of $\mathbb{E}[\mathbb{1}_{q_u}] \leq \epsilon$ are scaled by the

queues, and thus, we will have: $\mathbb{E}\left[\mathbb{1}_{q_u} q_u\right] \leq \epsilon \mathbb{E}\left[q_u\right]$. Now, the time average constraints in (12b) for all $u \in \mathcal{U}$ are modeled by virtual queues as follows:

$$\Psi_u(t+1) = [\Psi_u(t) + (\mathbb{1}_{q_u(t)} - \epsilon)q_u(t+1)]^+, \tag{13a}$$

$$A_u(t+1) = [A_u(t) + \left(q_u(t+1) - q_0 - \tfrac{\sigma}{1-\xi}\right)\mathbb{1}_{q_u(t)}]^+. \tag{13b}$$

$$B_u(t+1) = [B_u(t) + \left(\left(q_u(t+1) - q_0\right)^2 - \tfrac{2\sigma^2}{(1-\xi)(1-2\xi)}\right)\mathbb{1}_{q_u(t)}]^+ \tag{13c}$$

Let $\Xi_u(t) = [q_u(t), \Psi_u(t), A_u(t)), B_u(t))]$ be the combined queue with $\Xi(t) = [\Xi_u(t)]_{u \in \mathcal{U}}$ and its quadratic Lyapunov function $L(\Xi(t)) = \frac{1}{2}\Xi^\dagger(t)\Xi(t)$. 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,

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

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 Δ_0 and a set of terms $\{\Delta_u\}_u$ 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 $\mathbb{E}\left[L(\Xi(t+1)) - L(\Xi(t))|\Xi(t)\right]$. 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 $V\mathbb{E}\left[\sum_u \mathbf{1}^\dagger \boldsymbol{p}_u | \Xi(t)\right]$ to the expected drift and minimizing the upper bound of the drift plus penalty (DPP), $V\mathbb{E}\left[\sum_u \mathbf{1}^\dagger \boldsymbol{p}_u | \Xi(t)\right] + \mathbb{E}\left[\Delta L_t | \Xi(t)\right]$. As a result, our goal will now be to minimize the following upper bound:

$$\sum_{u \in \mathcal{U}} V\mathbf{1}^\dagger \boldsymbol{p}_u + \left(a_u(t) - r_u(t)\right)\Big\{(1 + \epsilon^2)q_u(t) - \epsilon\Psi_u(t) + [2(1 - \epsilon)q_u(t) + \Psi_u(t)+$$
$$A_u(t) - q_0 - \tfrac{\sigma}{1-\xi} + 2(q_u(t) - q_0)\left(B_u(t) + (q_u(t) - q_0)^2 - \tfrac{2\sigma^2}{(1-\xi)(1-2\xi)}\right)]\mathbb{1}_{q_u(t)}\Big\}, \quad (15)$$

at each time t. Assuming that VUEs maintain channel-quality indicators (CQIs), each VUE can estimate the interference $I_u^n(t) \simeq \tilde{I}_u^n(t)$ based on past observations (time averaged interference) [29]. Hence, the minimization of the above upper bound can be decoupled among VUEs as follows:

$$\underset{\boldsymbol{p}_u(t)}{\text{minimize}} \sum_{n \in \mathcal{N}_{z(t,u)}} \left[V p_u^n(t) - \alpha_u(t) \ln \left(1 + \gamma_u^n(t) p_u^n(t) \right) \right] \tag{16a}$$

$$\text{subject to} \quad \sum_{n \in \mathcal{N}_{z(t,u)}} p_u^n(t) \le p_0, \tag{16b}$$

$$p_u^n(t) \ge 0 \qquad \forall n \in \mathcal{N}_{z(t,u)}, \tag{16c}$$

where $\alpha_u(t) = \frac{W}{\ln 2} \{ (1+\epsilon^2) q_u(t) - \epsilon \Psi_u(t) + [2(1-\epsilon) q_u(t) + \Psi_u(t) + A_u(t) - q_0 - \frac{\sigma}{1-\xi} + 2(q_u(t) - q_0)(B_u(t) + (q_u(t) - q_0)^2 - \frac{2\sigma^2}{(1-\xi)(1-2\xi)})] \mathbb{1}_{q_u(t)} \}$ and $\gamma_u^n(t) = \frac{h_{uu}^n(t)}{\tilde{I}_u^n(t) + W N_0}$. The optimal solution of the convex optimization problem of (16) is obtained by a *water-filling algorithm* [30] where $[p_u^n(t)]^\star = [\frac{\alpha_u(t)}{V + \lambda_u^\star(t)} - \frac{1}{\gamma_u^n(t)}]^+$, with $\lambda_u(t) \ge 0$ 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 $[p_u^n(t)]^\star$, 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 $G_M^{\boldsymbol{d}}(m)$. Hence, estimating the parameters σ and ξ 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. We now define $\mathcal{T}_k = \{(k-1)w, (k-1)w+1, \ldots, kw-1\}$ as the set of time instants during block $k \in \mathbb{N}$. Then, the set of queue samples at time t is $\mathcal{Q}_u(t) = \{Q_u = q_u(t_k^\star) - q_0 | q_u(t_k^\star) > q_0, t_k^\star = \arg\max_{\tau \in \mathcal{T}_k} q_u(\tau), k \in \{1, \ldots, \lfloor t/w \rfloor\}\}$ with a

sample size $K_u(t)$. Note that $0 \notin \mathcal{Q}_u(t)$ 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 $\boldsymbol{d} = [\sigma, \xi]$ which need to be accurately estimated. For this purpose, we use MLE [31] whose objective is to find the best set of parameters $\boldsymbol{d}$ that fits the GPD $G_X^{\boldsymbol{d}}(\cdot)$ to the samples via maximizing the log likelihood function (or minimizing its negative) as follows:

$$\min_{\boldsymbol{d} \in \mathcal{D}(\mathcal{Q})} \quad f^{\boldsymbol{d}}(\mathcal{Q}) = -\frac{1}{|\mathcal{Q}|} \sum_{Q \in \mathcal{Q}} \log G_X^{\boldsymbol{d}}(Q), \tag{17}$$

where $\mathcal{D}(\mathcal{Q}) = \{[\sigma, \xi] \in \Re^2 | \sigma > 0, \xi < 1, 1 + \xi Q/\sigma) \geq 0$ for all $Q \in \mathcal{Q}\}$ is the feasible set, and $\mathcal{Q} = \{\mathcal{Q}_u\}_{u \in \mathcal{U}}$ 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 $\boldsymbol{d}$ and a summation over all the samples in $\mathcal{Q}$, and thus, its gradient over a sample Q can be derived as follows.

Proposition 2: The derivative coefficient of the negative log-likelihood function of GPD at the queue length sample Q w.r.t. $\boldsymbol{d}$ is,

$$\nabla_{\boldsymbol{d}} f^{\boldsymbol{d}}(Q) = \begin{bmatrix} \frac{\partial f^{\boldsymbol{d}}(Q)}{\partial \sigma} \\ \frac{\partial f^{\boldsymbol{d}}(Q)}{\partial \xi} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sigma} \left(\frac{1+1/\xi}{1+\xi Q/\sigma} - \frac{1}{\xi} \right) \\ \frac{(1+1/\xi)(2+\xi Q/\sigma)}{1+\xi Q/\sigma} - \frac{\ln(1+\xi Q/\sigma)}{\xi^2} \end{bmatrix}. \tag{18}$$

Proof: see Appendix B. ∎

Using the stochastic variance reduced gradient descent (SVRGD) technique [32] alongside $\nabla_{\boldsymbol{d}} f^{\boldsymbol{d}}(Q)$ at the RSU, the optimal $\boldsymbol{d}^\star$ 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 τ is defined as follows:

$$\begin{cases} \boldsymbol{y} & = \boldsymbol{d}(\tau) - \delta \big[\nabla_{\boldsymbol{d}} f^{\boldsymbol{d}(\tau)}(Q) - \nabla_{\boldsymbol{d}} f^{\tilde{\boldsymbol{d}}(\tau)}(Q) + \Upsilon(\tau) \big], \\ \boldsymbol{d}(\tau) & = \arg \min_{\boldsymbol{d} \in \mathcal{D}(\mathcal{Q})} \|\boldsymbol{y} - \boldsymbol{d}\|, \end{cases} \tag{19}$$

where $\tilde{\boldsymbol{d}}(\tau) = \frac{1}{\tau-1} \sum_{\tau'=1}^{\tau-1} \boldsymbol{d}(\tau')$ is an average estimate of $\boldsymbol{d}$ over previous iterations and $\Upsilon(\tau) = \frac{1}{|\mathcal{Q}|} \sum_{Q \in \mathcal{Q}} \nabla_{\boldsymbol{d}} f^{\tilde{\boldsymbol{d}}(\tau)}(Q)$ is an estimate of the gradient, respectively. After computing the GPD

Algorithm 1 Centralized GPD parameter estimation in **CEN**

1: **input:** Gradients $\Upsilon(0)$, estimations $\boldsymbol{d}(0)$, and step size δ at RSU.

2: **for** $T_f = 1, 2, \ldots$ **do**

3:     All VUEs upload their new queue samples $\{\hat{\mathcal{Q}}_u(T_f)\}_{u \in \mathcal{U}}$ to the RSU where $\hat{\mathcal{Q}}_u(T_f) = \mathcal{Q}_u(T_f) \setminus \mathcal{Q}_u(T_f - 1)$.

4:     **set:** $\Upsilon(T_f) = \boldsymbol{0}$ and $\boldsymbol{d}(T_f) = \boldsymbol{d}(T_f - 1)$ at RSU.

5:     Let $\{i^k\}_{k=1}^{\sum_u K_u(T_f)}$ be a random permutation of $\{\mathcal{Q}_u(T_f)\}_{u \in \mathcal{U}}$.

6:     **for** $k = 1, \ldots, \sum_u K_u(T_f)$ **do**

7:         Evaluate $\Upsilon(T_f)$ and $\boldsymbol{d}(T_f)$ using (19) at RSU.

8:     **end for**

9:     Share (download) the GPD parameters $\boldsymbol{d}(T_f)$ with all the VUEs.

10: **end for**

---

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 RBs available for V2V communication. This sample uploading over wireless links introduces an additional overhead in which a significant performance degradation can be expected in URLLC. As the VUE density increases, the sample size grows and so does VUEs' access to the RSU causing congestion resulting in increased network latency. Henceforth, the need for a distributed learning technique for MLE that does not require sharing all the local samples at VUEs with the RSU or one another is crucial.

C. FL-Based GPD Parameter Estimation

Towards developing a distributed learning mechanism for GPD parameter estimation, first we rewrite the likelihood function as follows:

$$f^{\boldsymbol{d}}(\mathcal{Q}) = \frac{1}{|\mathcal{Q}|} \sum_{Q \in \mathcal{Q}} \log G_X^{\boldsymbol{d}}(Q) = \sum_{u \in \mathcal{U}} \kappa_u f^{\boldsymbol{d}}(\mathcal{Q}_u), \tag{20}$$

where $\kappa_u = \frac{|\mathcal{Q}_u|}{|\mathcal{Q}|} = \frac{K_u}{\sum_{u'} K_{u'}}$. In (20), we express the likelihood function of the network as a weighted sum of likelihood functions per VUE. Hereinafter, for simplicity, we use $f^{\boldsymbol{d}}$ and $f_u^{\boldsymbol{d}}$
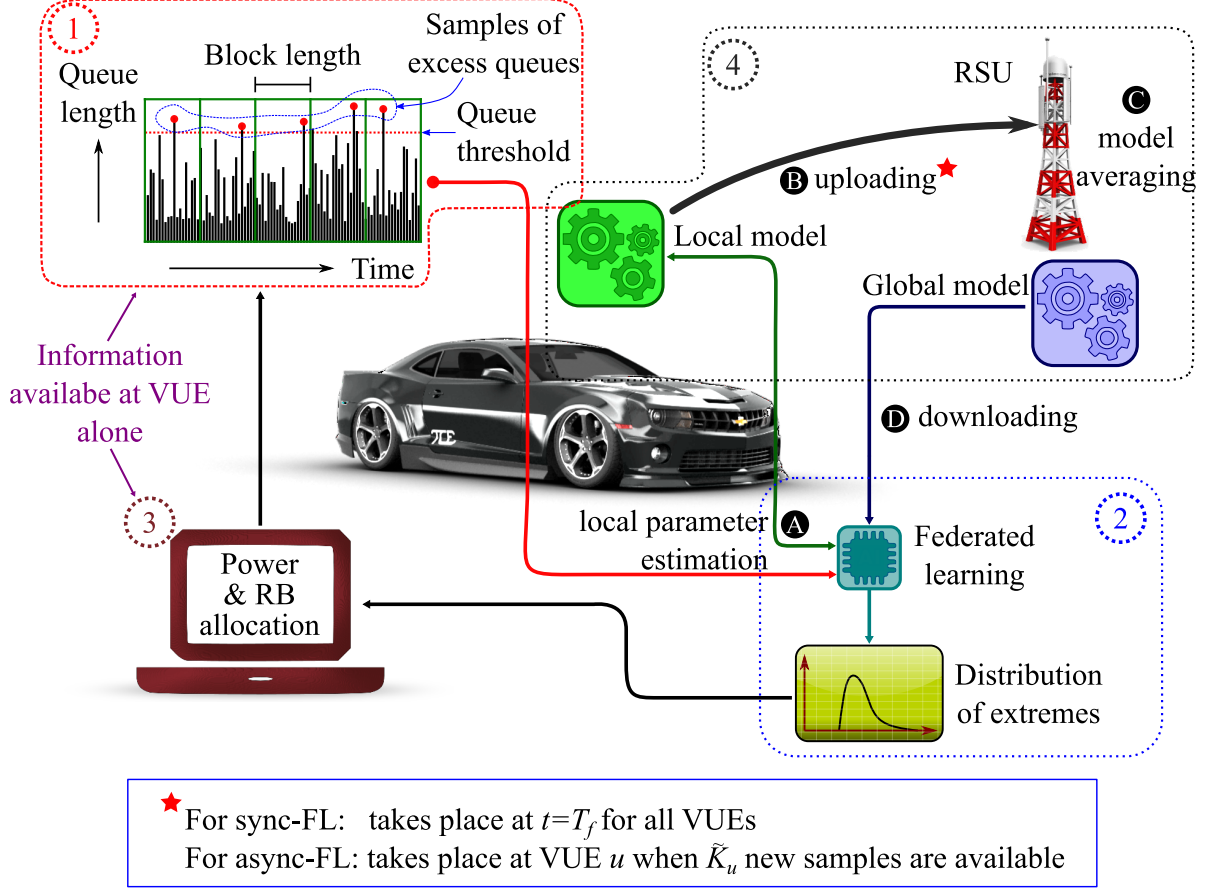
Fig. 2.   Interrelationships of the processes between VUEs and RSU: 1) excess queue sampling, 2) GPD parameter estimation, 3) transmit power and RB allocation, and 4) local and global models exchange with the RSU.

instead of $f^{\boldsymbol{d}}(\mathcal{Q})$ and $f^{\boldsymbol{d}}(\mathcal{Q}_u)$, respectively. The idea behind FL is to use $f_u^{\boldsymbol{d}}$ to evaluate $\boldsymbol{d}_u$ and $\nabla_{\boldsymbol{d}} f_u^{\boldsymbol{d}_u}$ locally, where $\boldsymbol{d}_u$ is the local estimate of $\boldsymbol{d}$ at VUE u, and update the local estimations via sharing the individual learning *models* $(\nabla_{\boldsymbol{d}} f_u^{\boldsymbol{d}_u}, \boldsymbol{d}_u, K_u, \hat{Q}_u)$ where $\hat{Q}_u = \max \mathcal{Q}_u$. Note that sharing $\hat{Q}_u$ over the network through RSU is sufficient to determine the domain $\mathcal{D}(\mathcal{Q})$, which is needed for the SVRGD procedure.

To evaluate the gradients and GPD parameters locally, VUE u uses the SVRGD with a step size $\delta_u = \delta/K_u$ [23]. In this case, given the local and global copies of the GPD parameters and gradients at a time τ, $\boldsymbol{d}_u(\tau)$, $\boldsymbol{d}(\tau)$, $\Upsilon_u(\tau)$, and $\Upsilon(\tau)$, respectively, the local GPD parameters and

gradients are updated for each QSI sample $i_u \in \mathcal{Q}_u$, process **A** in Fig. 2, as follows:

$$\begin{cases} \boldsymbol{y}_u & = \boldsymbol{d}_u(\tau) - \delta_u \big[\nabla_{\boldsymbol{d}} f_u^{\boldsymbol{d}_u(\tau)}(i_u) - \nabla_{\boldsymbol{d}} f_u^{\boldsymbol{d}(\tau)}(i_u) + \Upsilon(\tau) \big], \\ \boldsymbol{d}_u(\tau) & = \arg\min_{\boldsymbol{d}_u \in \mathcal{D}(\mathcal{Q}_u)} \|\boldsymbol{y}_u - \boldsymbol{d}_u\|, \\ \Upsilon_u(\tau) & = \nabla_{\boldsymbol{d}} f_u^{\boldsymbol{d}}(\tau) + \Upsilon_u(\tau). \end{cases} \tag{21}$$

After computing the gradients and GPD parameters locally, each VUE u uploads its model at time τ, $(\Upsilon_u, \boldsymbol{d}_u, K_u, \hat{Q}_u)_\tau$, 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:

$$\begin{cases} \boldsymbol{d}(\tau) = \boldsymbol{d}(\tau - 1) + \sum_u \kappa_u(\tau - 1)\big(\boldsymbol{d}_u(\tau - 1) - \boldsymbol{d}(\tau - 1)\big), \\ \Upsilon(\tau) = \frac{1}{\sum_{u'} K_{u'}(\tau-1)} \sum_u \Upsilon_u(\tau - 1). \end{cases} \tag{22}$$

Then, the global model $(\Upsilon, \boldsymbol{d}, \sum_u K_u, \hat{Q})_\tau$ 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 $(\Upsilon_u, \boldsymbol{d}_u, K_u, \hat{Q}_u)_{T_f}$ to the RSU. Then the RSU averages out all the local models after which all VUEs download the global model $(\Upsilon, \boldsymbol{d}, \sum_u K_u, \hat{Q})_{T_f}$. 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 $\tilde{K}$ of new QSI samples are collected. In essence, at time $\tilde{t}_u$ with $K_u(\tilde{t}_u)/\tilde{K} \in \mathbb{N}$, VUE u evaluates and uploads its local model $(\Upsilon_u, \boldsymbol{d}_u, K_u, \hat{Q}_u)_{\tilde{t}_u}$ 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 $(\Upsilon, \boldsymbol{d}, \sum_u K_u, \hat{Q})_{\tilde{t}_u}$ 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.

Algorithm 2 MLE for GPD using **sync-FL**

1: **input:** Gradients $\{\Upsilon_u(0)\}_{u \in \mathcal{U}}$, local estimations $\{\boldsymbol{d}_u(0)\}_{u \in \mathcal{U}}$, and step size δ.

2: **for** $T_f = 1, 2, \dots$ **do**

3:     **for** each VUE $u \in \mathcal{U}$ **do** {in parallel}

4:         **set:** $\Upsilon_u(T_f) = \boldsymbol{0}$, $\boldsymbol{d}_u(T_f) = \boldsymbol{d}(T_f)$ and $\delta_u = \delta/K_u(T_f)$.

5:         Let $\{i_u^k\}_{k=1}^{K_u(T_f)}$ be a random permutation of $\mathcal{Q}_u$.

6:         **for** $k = 1, \dots, K_u(T_f)$ **do**

7:             Evaluate $\Upsilon_u(T_f)$ and $\boldsymbol{d}_u(T_f)$ using (21).

8:         **end for**

9:         Upload the model $(\Upsilon_u, \boldsymbol{d}_u, K_u, \hat{Q}_u)_{T_f}$ to RSU.

10:     **end for**

11:     Update $\boldsymbol{d}(T_f)$ and $\Upsilon(T_f)$ at the RSU using (22).

12:     Share (download) the model $(\Upsilon, \boldsymbol{d}, \sum_u K_u, \hat{Q})_{T_f}$ with all the VUEs.

13: **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_Υ, $J_{\boldsymbol{d}}$, and J_Q be the sizes (in bits) of gradient values, GPD parameters, and queue samples of any VUE, respectively. Suppose VUE u has $\hat{K}_u$ new samples, and its uplink and downlink rate between RSU are r_{u0} and r_{0u}, respectively. In the **CEN** approach, VUE u dedicates $J_Q \hat{K}_u / r_{u0}$ time to upload all its new queue samples to the RSU while $J_{\boldsymbol{d}} / r_{0u}$ 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

---

Algorithm 3 MLE for GPD using async-FL

1: **input:** Gradients $\{\Upsilon_u(0)\}_{u \in \mathcal{U}}$, local estimations $\{\boldsymbol{d}_u(0)\}_{u \in \mathcal{U}}$, and step size δ.

2: **for** $t = 1, 2, \ldots$ **do**

3:    **for** each VUE $u \in \mathcal{U}$ **do** {in parallel}

4:       **if** $K_u(t)/\tilde{K} \in \mathbb{N}$ **then**

5:          **set:** $\tilde{t}_u = t$, $\Upsilon_u(\tilde{t}_u) = \boldsymbol{0}$, $\boldsymbol{d}_u(\tilde{t}_u) = \boldsymbol{d}(\tilde{t}_u)$ and $\delta_u = \delta/K_u(\tilde{t}_u)$.

6:          Let $\{i_u^k\}_{k=1}^{K_u(\tilde{t}_u)}$ be a random permutation of $\mathcal{Q}_u$.

7:          **for** $k = 1, \ldots, K_u(\tilde{t}_u)$ **do**

8:             Evaluate $\Upsilon_u(\tilde{t}_u)$ and $\boldsymbol{d}_u(\tilde{t}_u)$ using (21).

9:          **end for**

10:         Upload the model $(\Upsilon_u, \boldsymbol{d}_u, K_u, \hat{Q}_u)_{\tilde{t}_u}$ to RSU.

11:       **end if**

12:    **end for**

13:    RSU observes the set $\hat{\mathcal{U}}(t)$ of VUEs that upload models at time t.

14:    **if** $\hat{\mathcal{U}}(t) \neq \emptyset$ **then**

15:       Update $\boldsymbol{d}(t)$ and $\Upsilon(t)$ using (22).

16:       Share (download) the model $(\Upsilon, \boldsymbol{d}, \sum_u K_u, \hat{Q})_t$ with all VUEs $u \in \hat{\mathcal{U}}(t)$.
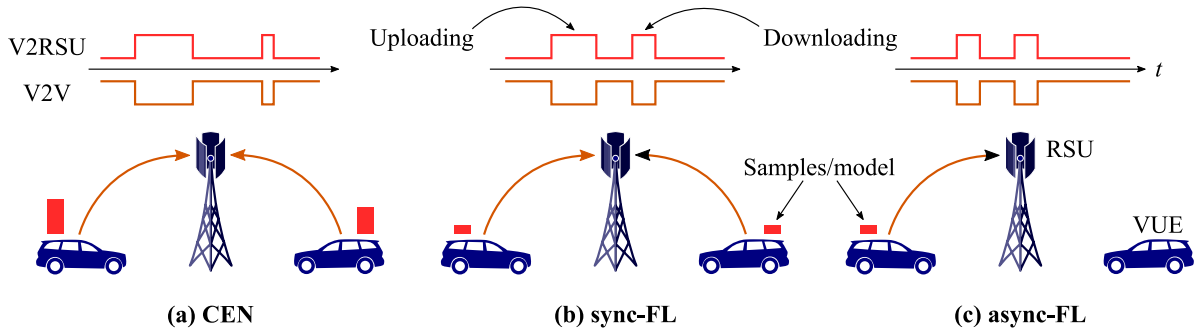
17:    **end if**

18: **end for**

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.

TABLE I

SIMULATION PARAMETERS [3], [5], [13], [27].

Para.	Value	Para.	Value	Para.	Value
ℓ	-68.5 dBm	W	180 kHz	w	10
ℓ'	-54.5 dBm	N_0	-174 dBm/Hz	δ	(50,0.005)
c	1.61	q_0	46.29 kb	$\nabla_{\boldsymbol{d}} f_u^{\boldsymbol{d}}(0)$	(1,1000)
d_0	15 m	ϵ	0.001	$\boldsymbol{d}_u(0)$	(1,0)

a result, an additional delay of $(J_Q \hat{K}_u / r_{u0} + J_{\boldsymbol{d}} / r_{0u})$ 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 $(J_\Upsilon + J_{\boldsymbol{d}} + J_Q)/r_{u0}$ and $(J_\Upsilon + J_{\boldsymbol{d}} + J_Q)/r_{0u}$ 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 $250\,\mathrm{m} \times 250\,\mathrm{m}$ 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 kmph and a fixed gap of 50 m. VUEs share 60 RBs and have a maximum transmit power of $p_0 = 10\,\mathrm{W}$. The RB allocation per zone is adopted from [5] and [3]. The rest of the parameter values are presented in Table I.

A. Centralized vs distributed GPD parameter estimation

Fig. 4 compares the accuracy of GDP 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 **CEN** estimations. To evaluate
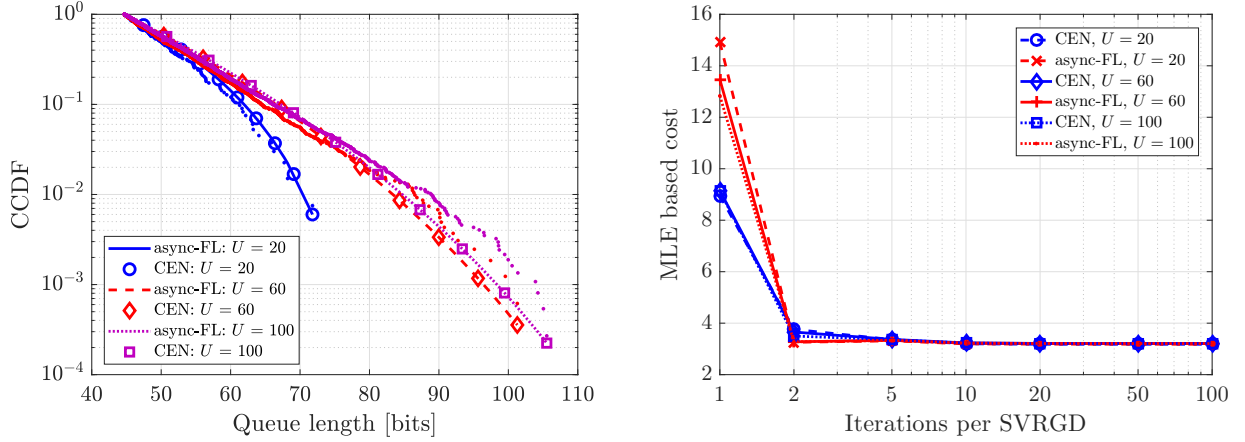
(a) Complementary cumulative distribution functions (CCDF) of queue lengths exceeding q_0 using **CEN** and **async-FL** methods for different number of VUEs.

(b) The MLE based cost $f^{\boldsymbol{d}}(\mathcal{Q})$ in (17) as a function of number of iterations used for SVRGD in **CEN** and **async-FL** methods. The selected scenarios are with number of queue length samples about $2U$.

Fig. 4. Comparison between **CEN** and **async-FL** in terms of the accuracy of GPD parameter estimation.

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**. For larger number of iterations (> 2) per SVRGD is used, the costs of **async-FL** is only about 0.5% higher than the costs of **CEN** when $U = 20$ while for $U = 100$, **async-FL** yields about 0.5% lower cost compared to **CEN**. It highlights that the performance of FL improves over a centralized SVRGD-based estimator with the increasing sample size.

In Fig. 5, we compare the amount of data exchange and the achieved reliability in terms of
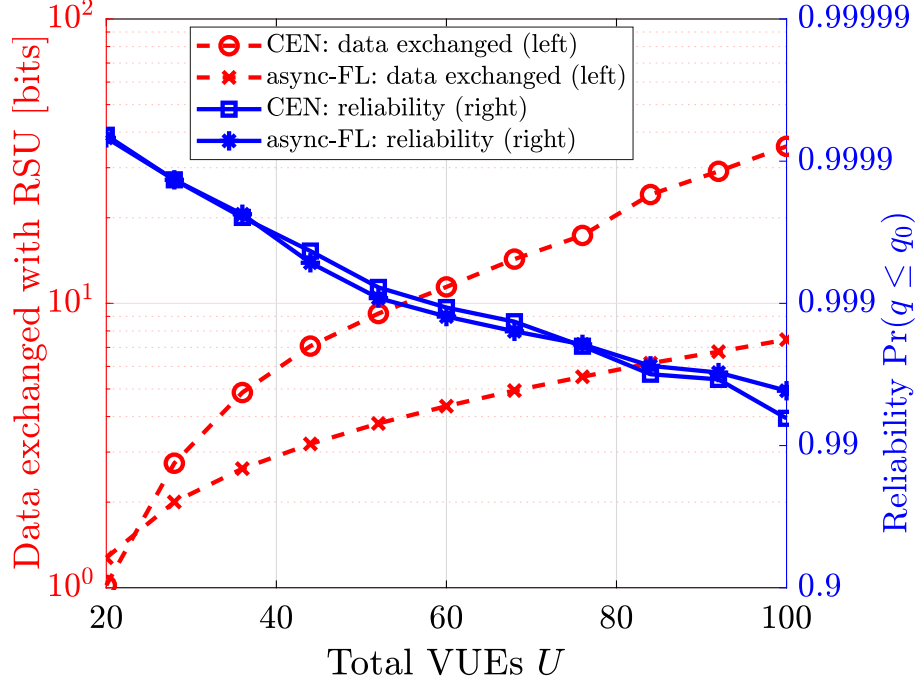
Fig. 5. Comparison of the amount of data exchanged between RSU and VUEs (left) and the achieved reliability (right) for different VUE settings with two approaches used to estimate GPD parameters: the proposed FL and the centralized SVRG method.

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 $(\Upsilon_u, \boldsymbol{d}_u, K_u, \hat{Q}_u)$ 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-FL**, VUEs must upload and download both parameters and gradients yielding higher data exchange compared to **CEN**. However, as the number of VUEs increases (beyond 28), the sample size grows, and thus, **CEN** incurs higher amount of data exchanged between the RSU and VUEs compared to **async-FL**. The reductions of the exchanged data in **async-FL** compared to **CEN** is about 27% for $U = 28$ and improves up to 79% when $U = 100$. Finally, Fig. 5 clearly demonstrates that the **async-FL** approach is particularly effective for large-scale and dense vehicular networks.
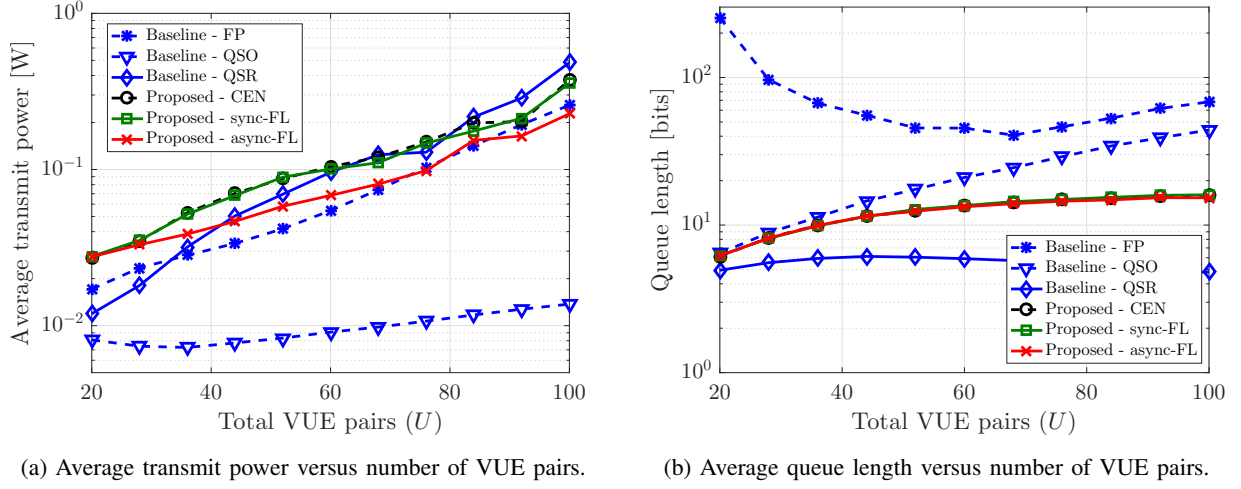
(a) Average transmit power versus number of VUE pairs.

(b) Average queue length versus number of VUE pairs.

Fig. 6. Comparison of average transmit power and average queue lengths for different number of VUEs.

B. Performance Evaluation

Next, the proposed approaches, **CEN**, **sync-FL**, and **async-FL**, that utilize EVT to characterize the tail distribution of queue lengths are compared with three other baseline models namely: *i)* **FP**: a V2V network where vTxs use fixed transmit power, *ii)* **QSO**: a V2V network with the objective of power minimization while ensuring only the queue stability (3)-(4), and *iii)* **QSR**: a V2V network that minimizes transmit power while focusing on the probabilistic constraint on average queue length and the queue stability (3)-(5).

Fig. 6a compares the average transmit power of all approaches for different VUE densities. For a fair comparison, the transmit power of vTxs 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
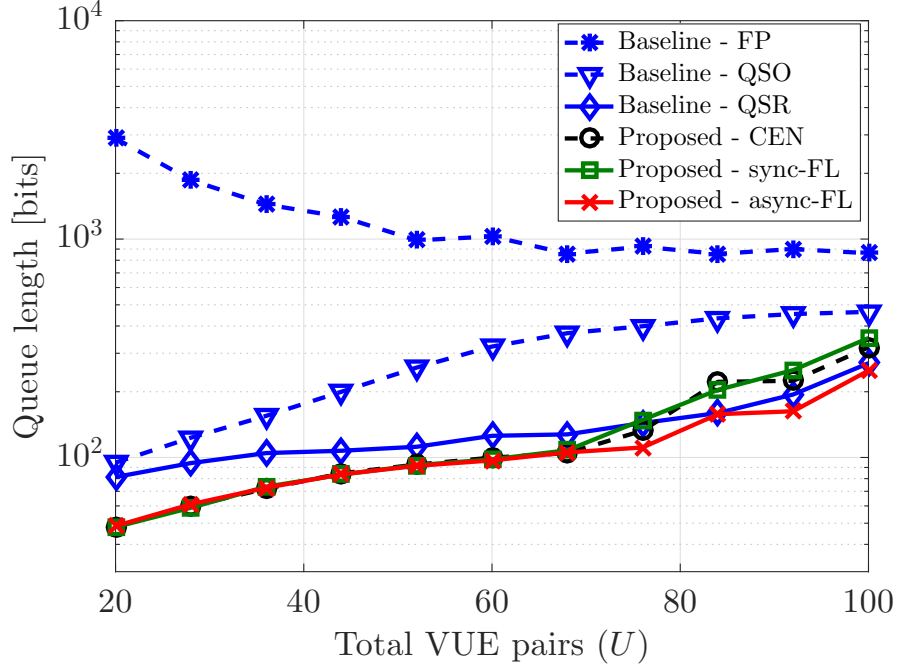
Fig. 7.   Maximum queue length corresponding to the worst-case latency as a function of the number of VUE pairs.

requirement of lower transmit power to upload/download learning models due to asynchronous communication between the RSU and VUEs in **async-FL** results 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. 7 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. 7 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. 8 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. 8. 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.
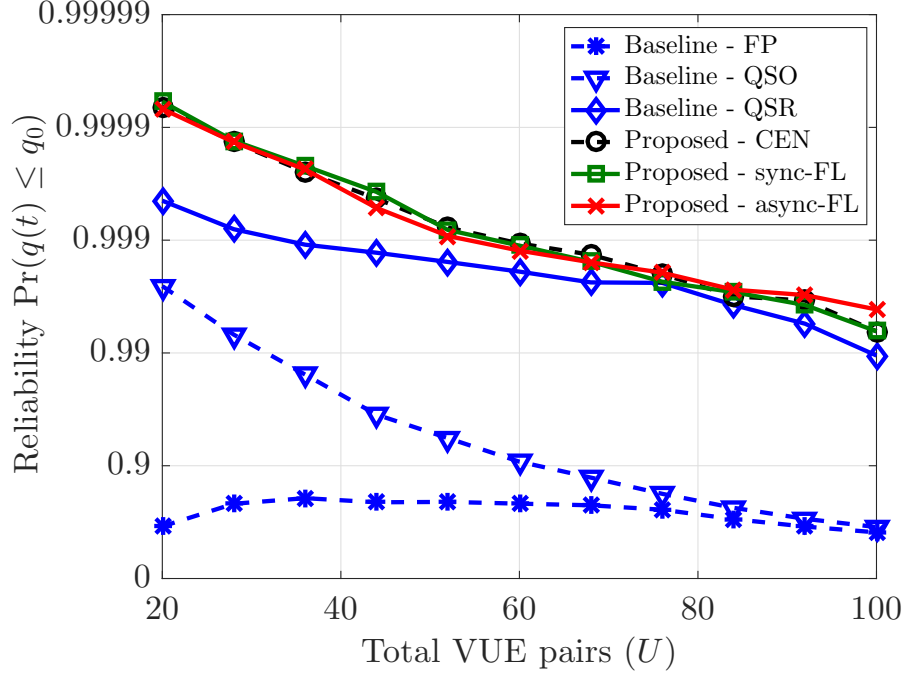
Fig. 8. Reliability in terms of the probability that the queue lengths are maintained below q_0 versus number of VUE pairs.

Fig. 9 illustrates the mean and standard deviation of the 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. 9, 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.
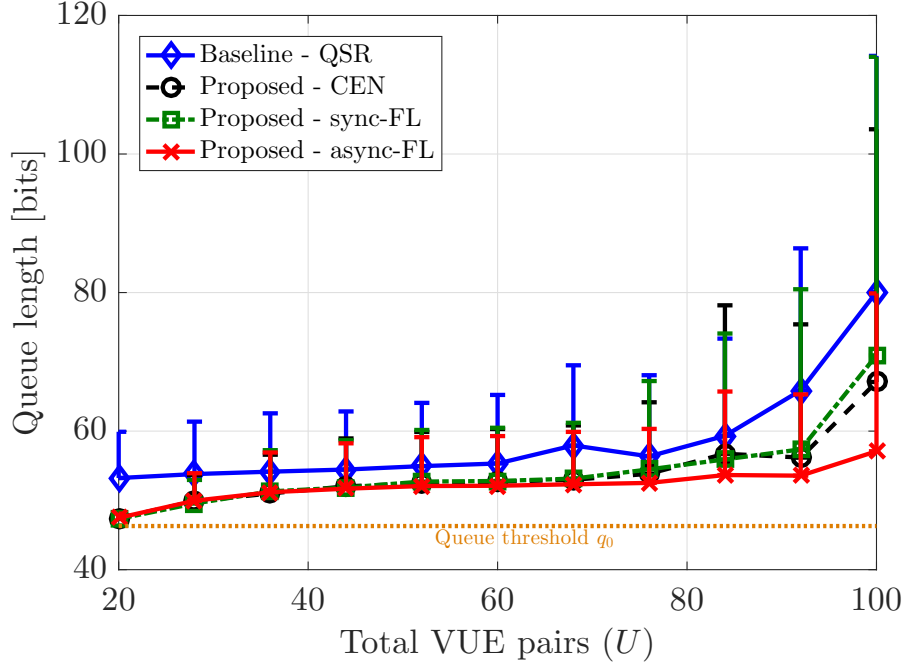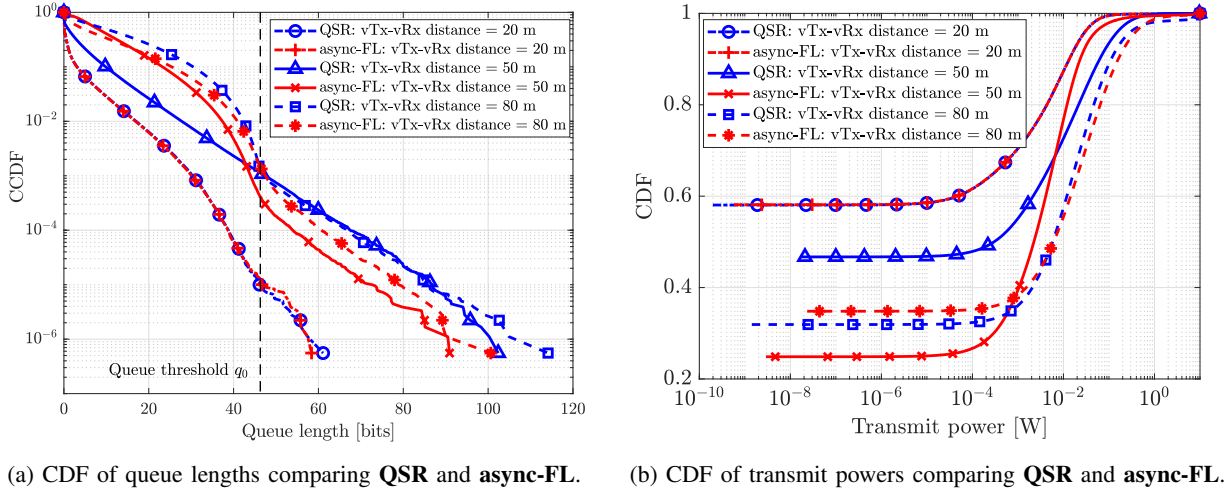
Fig. 9. Mean and standard deviation of the tail distribution, i.e., distribution of queue length exceeding q_0, versus number of VUE pairs.

(a) CDF of queue lengths comparing **QSR** and **async-FL**.

(b) CDF of transmit powers comparing **QSR** and **async-FL**.

Fig. 10. The impact of vTx-vRx distances on queue lengths and transmit powers.

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. 10. According to Fig. 10a, 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 $20\,\mathrm{m}$ and $80\,\mathrm{m}$, respectively. For $50\,\mathrm{m}$, **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 $20\,\mathrm{m}$, $50\,\mathrm{m}$, and $80\,\mathrm{m}$, respectively. Fig. 10b 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. 10b 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 $20\,\mathrm{m}$, $50\,\mathrm{m}$, and $80\,\mathrm{m}$, respectively, compared to **QSR** method.

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.

APPENDIX A

PROOF OF PROPOSITION 1

First, consider the one-slot drift of the Lyapunov function.

$$\Delta L_t = L(\mathbf{\Xi}(t+1)) - L(\mathbf{\Xi}(t))$$

$$= \sum_{u \in \mathcal{U}} \left(\frac{q_u^2(t+1) - q_u^2(t)}{2} + \frac{\Psi_u^2(t+1) - \Psi_u^2(t)}{2} + \frac{A_u^2(t+1) - A_u^2(t)}{2} + \frac{B_u^2(t+1) - B_u^2(t)}{2}. \right) \quad (23)$$

Using the relation $([q + a - r]^+)^2 \le q^2 + (a - r)^2 + 2q(a - r)$ for the $(t+1)$-th terms in (23), upper bounds for each of the above terms can be derived as follows:

$$\frac{q_u^2(t+1) - q_u^2(t)}{2} \le \left[\frac{(a_u(t) - r_u(t))^2}{2} \right]_{\#\mathrm{a}_1} + q_u(t)(a_u(t) - r_u(t)), \quad (24\mathrm{a})$$

$$\frac{\Psi_u^2(t+1) - \Psi_u^2(t)}{2} \le \frac{(\mathbb{1}_{q_u(t)} - \epsilon)^2 q_u^2(t+1)}{2} + \Psi_u(t)(\mathbb{1}_{q_u(t)} - \epsilon)q_u(t+1),$$

$$= \left[\frac{(\mathbb{1}_{q_u(t)} - \epsilon)^2 \left(q_u^2(t) + (a_u(t) - r_u(t))^2 \right)}{2} \right]_{\#\mathrm{a}_2} + \left[\Psi_u(t)(\mathbb{1}_{q_u(t)} - \epsilon)q_u(t) \right]_{\#\mathrm{b}_1}$$

$$+ (\mathbb{1}_{q_u(t)} - \epsilon)(a_u(t) - r_u(t))[\Psi_u(t) + (\mathbb{1}_{q_u(t)} - \epsilon)q_u(t)], \quad (24\mathrm{b})$$

$$\frac{A_u^2(t+1) - A_u^2(t)}{2} \le \left\{ \frac{q_u^2(t+1) + M_1^2}{2} - M_1 A_u(t) + (A_u(t) - M_1)q_u(t+1) \right\} \mathbb{1}_{q_u(t)},$$

$$= \left\{ \left[\frac{q_u^2(t) + (a_u(t) - r_u(t))^2 + M_1^2}{2} \right]_{\#\mathrm{a}_3} + \left[(A_u(t) - M_1)q_u(t) - M_1 A_u(t) \right]_{\#\mathrm{b}_2} \right.$$

$$\left. + (q_u(t) + A_u(t) - M_1)(a_u(t) - r_u(t)) \right\} \mathbb{1}_{q_u(t)}, \quad (24\mathrm{c})$$

$$\frac{B_u^2(t+1) - B_u^2(t)}{2} \le \left\{ \frac{\left((q_u(t+1) - q_0)^2 - M_2 \right)^2}{2} + B_u(t)\left((q_u(t+1) - q_0)^2 + M_2 \right) \right\} \mathbb{1}_{q_u(t)},$$

$$= \left\{ \left[\frac{\left((q_u(t) - q_0)^2 + (a_u(t) - r_u(t))^2 - M_2 \right)^2}{2} + \left((q_u(t) - q_0)^2 + (a_u(t) - r_u(t))^2 \right) B_u(t) \right. \right.$$

$$\left. + 2\left((q_u(t) - q_0)^2 + (q_u(t) - q_0)(a_u(t) - r_u(t)) \right)(a_u(t) - r_u(t))^2 \right]_{\#\mathrm{a}_4}$$

$$\left. + 2(q_u(t) - q_0)(a_u(t) - r_u(t))\left((q_u(t) - q_0)^2 - M_2 + B_u(t) \right) \right\} \mathbb{1}_{q_u(t)}. \quad (24\mathrm{d})$$

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 $\mathbb{1}_{q_u(t)}$, 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 $\mathbb{1}_{q_u(t)} = 1$ ensures nonempty queues, i.e. $\mathbb{1}_{q_u(t)} = 1 \implies q_u(t+1) = [q_u(t) + a_u(t) - r_u(t)]^+ = q_u(t) + a_u(t) - r_u(t) > 0$.

Note that the terms $\#a_1$-$\#a_4$ are quadratic in which the assumption of queue stability forces them to be bounded. Hence sum of $\#a_1$-$\#a_4$ terms are replaced by a bounded value Δ_0 with,

$$\Delta_0 \geq \sum_{u \in \mathcal{U}} \left[\frac{(a_u(t) - r_u(t))^2}{2} + \frac{(\mathbb{1}_{q_u(t)} - \epsilon)^2 \left(q_u^2(t) + (a_u(t) - r_u(t))^2 \right)}{2} + \left\{ \frac{q_u^2(t) + (a_u(t) - r_u(t))^2 + M_1^2}{2} \right. \right.$$

$$+ \frac{\left((q_u(t) - q_0)^2 + (a_u(t) - r_u(t))^2 - M_2 \right)^2}{2} + \left((q_u(t) - q_0)^2 + (a_u(t) - r_u(t))^2 \right) B_u(t)$$

$$\left. \left. + 2\left((q_u(t) - q_0)^2 + (q_u(t) - q_0)(a_u(t) - r_u(t)) \right)(a_u(t) - r_u(t))^2 \right\} \mathbb{1}_{q_u(t)} \right]. \quad (25)$$

Terms $\#b_1$ and $\#b_2$ are independent from the control variables. Therefore, we denote them by Δ_u for VUE $u \in \mathcal{U}$ where

$$\Delta_u = \Psi_u(t)(\mathbb{1}_{q_u(t)} - \epsilon)q_u(t) + (A_u(t) - M_1)q_u(t) - M_1 A_u(t). \quad (26)$$

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

APPENDIX B

PROOF OF PROPOSITION 2

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

$$f^{\boldsymbol{d}}(\mathcal{Q}) = \frac{1}{|\mathcal{Q}|} \sum_{Q \in \mathcal{Q}} \left(\ln \sigma - (\xi + 1) \ln g^{\boldsymbol{d}}(Q) \right) = \frac{1}{|\mathcal{Q}|} \sum_{Q \in \mathcal{Q}} f^{\boldsymbol{d}}(Q). \quad (27)$$

Hence, $\nabla_{\boldsymbol{d}} f^{\boldsymbol{d}}(\mathcal{Q}) = \frac{1}{|\mathcal{Q}|} \sum_{Q \in \mathcal{Q}} \nabla_{\boldsymbol{d}} f^{\boldsymbol{d}}(Q)$ is held.

First, the gradient of $g^{\boldsymbol{d}}(Q)$ is found by,

$$\nabla_{\boldsymbol{d}} g^{\boldsymbol{d}}(Q) = \begin{bmatrix} \frac{\partial g^{\boldsymbol{d}}(Q)}{\partial \sigma} \\ \frac{\partial g^{\boldsymbol{d}}(Q)}{\partial \xi} \end{bmatrix} = \begin{bmatrix} \frac{Q}{\sigma} g^{\boldsymbol{d}}(Q)^{\xi+1} \\ \frac{g^{\boldsymbol{d}}(Q)}{\xi^2} \left(g^{\boldsymbol{d}}(Q)^{\xi} - \xi \ln g^{\boldsymbol{d}}(Q) - 1 \right) \end{bmatrix}. \quad (28)$$

Thus, the gradient of $f^{\boldsymbol{d}}(Q)$ can be calculated as follows:

$$\nabla_{\boldsymbol{d}} f^{\boldsymbol{d}}(Q) = \begin{bmatrix} \frac{\partial f^{\boldsymbol{d}}(Q)}{\partial \sigma} \\ \frac{\partial f^{\boldsymbol{d}}(Q)}{\partial \xi} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sigma} - \frac{1+\xi}{g^{\boldsymbol{d}}(Q)} \frac{\partial g^{\boldsymbol{d}}(Q)}{\partial \sigma} \\ -\frac{1+\xi}{g^{\boldsymbol{d}}(Q)} \frac{\partial g^{\boldsymbol{d}}(Q)}{\partial \xi} - \ln g^{\boldsymbol{d}}(Q) \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sigma} \left(\frac{1+1/\xi}{1+\xi Q/\sigma} - \frac{1}{\xi} \right) \\ \frac{(1+1/\xi)(2+\xi Q/\sigma)}{1+\xi Q/\sigma} - \frac{\ln(1+\xi Q/\sigma)}{\xi^2} \end{bmatrix}.$$

REFERENCES

[1] A. E. Fernandez and M. Fallgren, "5GCAR scenarios, use cases, requirements and KPIs," Fifth Generation Communication Automotive Research and innovation, Tech. Rep. D2.1, Aug. 2017.

[2] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally, "5G for vehicular communications," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 111–117, Jan 2018.

[3] C.-F. Liu and M. Bennis, "Ultra-reliable and low-latency vehicular transmission: an extreme value theory approach," *IEEE Communications Letters*, pp. 1–1, 2018.

[4] T. Zeng, O. Semiari, W. Saad, and M. Bennis, "Joint communication and control for wireless autonomous vehicular platoon systems," *CoRR*, vol. abs/1804.05290, 2018. [Online]. Available: https://arxiv.org/abs/1804.05290

[5] M. I. Ashraf, C.-F. Liu, M. Bennis, and W. Saad, "Towards low-latency and ultra-reliable vehicle-to-vehicle communication," in *Proc. of European Conference on Networks and Communications (EuCNC)*, Oulu, Finland, June 2017, pp. 1–5.

[6] "5G highlights," 5G Technology Workshop-Potential Technology for 3GPP, Tech. Rep. Rel-15, 2016.

[7] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: performance and tradeoffs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, June 2016.

[8] Z. Zhou, F. Xiong, C. Xu, Y. He, and S. Mumtaz, "Energy-efficient vehicular heterogeneous networks for green cities," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1522–1531, April 2018.

[9] W. Kumar, S. Bhattacharya, B. R. Qazi, and J. M. H. Elmirghani, "A vacation-based performance analysis of an energy-efficient motorway vehicular communication system," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 4, pp. 1827–1842, May 2014.

[10] P. Dong, X. Du, J. Sun, and H. Zhang, "Energy-efficient cluster management in heterogeneous vehicular networks," in *Proc. of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, San Francisco, CA, USA, April 2016, pp. 644–649.

[11] W. Sun, E. G. Strom, F. Brannstrom, Y. Sui, and K. C. Sou, "D2D-based V2V communications with latency and reliability constraints," in *Proc. of IEEE Globecom Workshops (GC Wkshps)*, Dec 2014, pp. 1414–1419.

[12] C. Perfecto, J. D. Ser, and M. Bennis, "On the interplay between scheduling interval and beamwidth selection for low-latency and reliable V2V mmwave communications," in *Proc. of Conference on Innovations in Clouds, Internet and Networks (ICIN)*, March 2017, pp. 1–8.

[13] M. I. Ashraf, M. Bennis, C. Perfecto, and W. Saad, "Dynamic proximity-aware resource allocation in vehicle-to-vehicle (V2V) communications," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[14] B. Zhou and W. Saad, "Optimal sampling and updating for minimizing age of information in the internet of things," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, editor, Ed., Abu Dhabi, UAE, Dec. 2018.

[15] T. Liu, Y. Zhu, R. Jiang, and Q. Zhao, "Distributed social welfare maximization in urban vehicular participatory sensing systems," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, Oct. 2017.

[16] J. Mei, K. Zheng, L. Zhao, Y. Teng, and X. Wang, "A latency and reliability guaranteed resource allocation scheme for LTE V2V communication systems," *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, pp. 1–1, Mar. 2018.

[17] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Federated learning for ultra-reliable low-latency V2V communications," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, UAE, Dec. 2018.

[18] M. Bennis, M. Debbah, and H. V. Poor, "Ultra-reliable and low-latency wireless communication: tail, risk and scale," *CoRR*, vol. abs/1801.01270, 2018. [Online]. Available: http://arxiv.org/abs/1801.01270

[19] C. Liu, Y. Shu, J. Liu, and O. W. W. Yang, "Application of extreme value theory to the analysis of wireless network traffic," in *Proc. of IEEE International Conference on Communications*, June 2007, pp. 486–491.

[20] A. Mouradian, "Extreme value theory for the study of probabilistic worst case delays in wireless networks," *Ad Hoc Networks*, vol. 48, pp. 1 – 15, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870516301287

[21] S. Wei, D. L. Goeckel, and P. E. Kelly, "A modern extreme value theory approach to calculating the distribution of the peak-to-average power ratio in OFDM systems," in *Proc. of IEEE International Conference on Communications (ICC)*, vol. 3, 2002, pp. 1686–1690.

[22] B. Finkenstadt and H. Rootzen, Eds., *Extreme values in finance, telecommunications, and the environment*, 1st ed., ser. Monographs on Statistics & Applied Probability (Book 99). Chapman and Hall/CRC, 2003.

[23] J. Konecný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: distributed machine learning for on-device intelligence," *CoRR*, vol. abs/1610.02527, 2016. [Online]. Available: http://arxiv.org/abs/1610.02527

[24] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4424–4434. [Online]. Available: http://papers.nips.cc/paper/7029-federated-multi-task-learning.pdf

[25] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: strategies for improving communication efficiency," in *Proc. of NIPS Workshop on Private Multi-Party Machine Learning*, 2016. [Online]. Available: https://arxiv.org/abs/1610.05492

[26] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," 2018 IEEE Global Communications Conference (GLOBECOM), submitted. [Online]. Available: https://arxiv.org/abs/1804.08333

[27] T. Mangel, O. Klemp, and H. Hartenstein, "A validated 5.9 GHz non-line-of-sight path-loss and fading model for inter-vehicle communication," in *Proc. of 11th International Conference on ITS Telecommunications*, Aug 2011, pp. 75–80.

[28] M. J. Neely, *Stochastic network optimization with application to communication and queueing systems*, ser. Synthesis Lectures on Communication. Morgan & Claypool Publishers, 2010, vol. 3, no. 1.

[29] P. Luoto, M. Bennis, P. Pirinen, S. Samarakoon, K. Horneman, and M. Latva-aho, "Vehicle clustering for improving enhanced LTE-V2X network performance," in *Proc. of European Conference on Networks and Communications (EuCNC)*, Oulu, Finland, June 2017, pp. 1–5.

[30] P. Massoud Salehi and J. Proakis, *Digital communications*. McGraw-Hill Education, 2007.

[31] P. G. Hoel, *Introduction to mathematical statistics*, 5th ed. Wiley, 1984.

[32] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. of the 26th International Conference on Neural Information Processing Systems*, ser. NIPS'13, vol. 1. USA: Curran Associates Inc., 2013, pp. 315–323. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999611.2999647