Resource Constrained Vehicular Edge Federated Learning With Highly Mobile Connected Vehicles

Md Ferdous Pervej, *Graduate Student Member, IEEE*, Richeng Jin, *Member, IEEE*, and Huaiyu Dai, *Fellow, IEEE*

***Abstract*— This paper proposes a vehicular edge federated learning (VEFL) solution, where an edge server leverages highly mobile connected vehicles' (CVs') onboard central processing units (CPUs) and local datasets to train a global model. Convergence analysis reveals that the VEFL training loss depends on the successful receptions of the CVs' trained models over the intermittent vehicle-to-infrastructure (V2I) wireless links. Owing to high mobility, in the full device participation case (FDPC), the edge server aggregates client model parameters based on a weighted combination according to the CVs' dataset sizes and sojourn periods, while it selects a subset of CVs in the partial device participation case (PDPC). We then devise joint VEFL and radio access technology (RAT) parameters optimization problems under delay, energy and cost constraints to maximize the probability of successful reception of the locally trained models. Considering that the optimization problem is NP-hard, we decompose it into a VEFL parameter optimization sub-problem, given the estimated worst-case sojourn period, delay and energy expense, and an online RAT parameter optimization sub-problem. Finally, extensive simulations are conducted to validate the effectiveness of the proposed solutions with a practical 5G new radio (5G-NR) RAT under a realistic microscopic mobility model.**

***Index Terms*— Connected vehicle (CV), energy efficiency (EE), federated learning (FL), vehicular edge network (VEN).**

I. Introduction

WHILE modern connected vehicles (CVs) are an essential part of an intelligent transportation system (ITS), higher automation on the road demands more exploration. One way to achieve higher automation is to put more sensors on the onboard units of these CVs to facilitate real-time sensing and onboard computing [1]. Machine learning (ML) has shown its potential in various ITS applications, such as object detection, traffic sign classification, congestion prediction, velocity/acceleration prediction, etc., to name a few [2]. However, the sensing capabilities and onboard computation powers of CVs are still limited. Moreover, offloading raw data to an edge server raises immense privacy risks and requires humongous bandwidth. Therefore, a privacy-preserving distributed ML solution is urgently needed for modern vehicular edge networks (VENs) to ensure higher automation levels on the road where the moving CVs must make operational decisions swiftly.

With its privacy-preserving and distributed learning abilities, federated learning (FL) [3] is, thus, an ideal solution for VENs. Note that FL follows the parameter server paradigm, where the server distributes a global ML model to the clients, who then perform local model training in parallel on their devices and send their locally trained model parameters to the server [4]. Thus, the CVs do not need to share their raw data, i.e., data remains private. Besides, system and data heterogeneity of the CVs can be handled by carefully designing model aggregation rules and local training loss functions.

Unlike traditional stationary clients, however, devising a vehicular edge FL (VEFL) framework is challenging for multiple reasons. Firstly, limited radio coverage makes the sojourn periods of the highly mobile CVs very short. Therefore, the CVs can perform local model training only for a few iterations before moving out of the coverage area. Secondly, modern CVs' onboard central processing units (CPUs) are responsible for many operational computations. Besides, the CVs are owned by different clients who may not readily join the FL process. Therefore, a service level agreement (SLA) between a CV that wishes to utilize its limited resource for FL model training and the edge server should exist. Note that an SLA is a commitment between the server and the CV that both parties agree to uphold. Thirdly, a proper radio access technology (RAT) solution is required since the server can aggregate trained models only if these models are successfully received at the aggregation time. However, the high mobility of the CVs makes communication over the intermittent wireless vehicle-to-infrastructure (V2I) links even more challenging. As such, we shall carefully orchestrate the interplay between the server and the RAT solution to perform VEFL. Moreover, the underlying RAT

Manuscript received 26 October 2022; revised 6 April 2023; accepted 10 April 2023. Date of publication 8 May 2023; date of current version 19 June 2023. This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LQ23F010021, in part by the Ng Teng Fong Charitable Foundation in the form of ZJU-SUTD IDEA Grant under Grant 188170-11102, in part by the National Key Research and Development Program of China under Grant 2018YFB1801104, and in part by the U.S. National Science Foundation under Grant CNS-1824518 and Grant ECCS-2203214. *(Corresponding author: Richeng Jin.)*

Md Ferdous Pervej and Huaiyu Dai are with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695 USA (e-mail: mpervej@ncsu.edu; hdai@ncsu.edu).

Richeng Jin is with the Zhejiang–Singapore Innovation and AI Joint Research Laboratory, Department of Information and Communication Engineering, Zhejiang University, Hangzhou 310007, China, and also with the Zhejiang Provincial Key Laboratory of Information Processing, Communication, and Networking (IPCAN), Hangzhou 310007, China (e-mail: richengjin@zju.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2023.3273700.

Digital Object Identifier 10.1109/JSAC.2023.3273700

requires mandatory resource management. Finally, system and data heterogeneity among the CVs is a norm in VENs since automotive makers produce products with different features.

A. Related Work

We have seen many remarkable contributions to joint FL and wireless network parameter optimizations [4], [5], [6], [7], [8]. However, these studies did not consider the fundamental constraint in VEN, i.e., client's high mobility, which results in a very short sojourn period. Some recent works [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] also considered FL for different tasks in VENs. However, only a handful of studies [19], [20], [21], [22] addressed the constraints present in VENs. Zeng et al. proposed a dynamic federated proximal algorithm to design a controller for autonomous vehicles in [19]. The authors considered moving connected and autonomous vehicles as FL clients and devised their algorithm accounting for the communication and computation delay constraints. The communication delay was derived using a simplistic channel model with one channel realization. Each vehicle had a unique orthogonal resource block to offload its trained model to the server.

Xiao et al. jointly consider vehicular client selection, transmission power selection, CPU frequency selection and local model accuracy optimization under delay and energy constraints in [20]. More specifically, the authors assumed data quality is known, and optimized local model precision before the server performs global aggregation. Besides, the authors considered a transmission control protocol/internet protocol based channel model with a unique radio resource for each client for offloading its trained model. Taik et al. considered a clustered vehicular FL in [21]. The authors used the traditional federated averaging (FedAvg) algorithm, where the vehicular cluster head performed model aggregation from the cluster members and then forwarded the aggregated model to the server. Liu et al. used a proximal FL algorithm, which is very similar to the widely used FedProx algorithm [23], for vehicular edge computing in [22]. The impact of mobility and wireless links was not considered in [22].

Asynchronous communication and model aggregation mechanisms were also proposed in some recent works [24], [25], [26]. More specifically, [24] considered a semi-synchronous FL for the Internet of vehicles, where the authors dynamically adjusted the server's waiting time between two global rounds in proportion to the total participating clients. A hierarchical asynchronous FL was considered in [25]. A semi-asynchronous hierarchical FL for transportation system was proposed in [26]. Particularly, [26] assumed a synchronous model aggregation for the local-edge level and a semi-synchronous model aggregation for the edge-cloud level.

B. Motivations and Our Contributions

While [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] showed the efficacy of FL in different vehicular applications and [19], [20], [21], [22] addressed some typical resource-constraints in VENs, these studies had their own limits. Particularly, due to the intermittent wireless V2I links, VEFL is not as straightforward as broadcasting the global model and then aggregating locally trained model parameters under perfect wireless communication links between the server and clients. A practical RAT, such as the 5G new radio (5G-NR), is required for the parameter server to broadcast the global model in the downlink and then receive the model from the vehicular clients in the uplink. Moreover, the parameter server must devise the VEFL strategy to accommodate the underlying RAT's characteristics. As such, a joint study should address the constraints of the parameter server, mobile clients and the underlying RAT. It is worth pointing out that 3rd generation partnership project (3GPP) release 18 will include different artificial intelligence and ML solutions for its data-driven network applications [27]. Besides, different work groups within 3GPP are working actively to include ML in the next-generation standard. Moreover, ML-application-based RAT design is also a part of standardization for release 18 [28].

In this work, we, therefore, present a VEFL framework with a joint study of the impact of the mobility of the clients, i.e., the CVs, with a practical 5G-NR-based RAT solution and under strict delay, energy, computation resource, radio resource and cost constraints. More specifically, our contributions are summarized as follows:

- Leveraging 5G-NR RAT, we propose a VEFL framework where an edge server utilizes a fixed bandwidth part (BWP) [29] and an uplink heavy frame structure to receive the locally trained ML models over the intermittent V2I links from highly mobile CVs which participate in the model training and charge the server based on SLAs.
- We consider a full device participation case (FDPC) and a more practical partial device participation case (PDPC), where all CVs and only a subset of CVs participate in the model training, respectively. As FDPC is less flexible, to combat high mobility, i.e., short sojourn period, the server aggregates local model parameters based on a weighted combination reflecting the CVs' expected sojourn periods and dataset sizes.
- In both cases, corresponding joint VEFL and RAT parameter optimization problems are formulated to maximize the probability of successful trained models reception at the server under strict delay, energy and cost constraints. Since channel state information (CSI) can vary in each slot and is unknown beforehand, the original joint problem is decomposed into a VEFL parameter optimization sub-problem, given the upper bounds of the communication delay, energy expense and cost, and a RAT parameter optimization sub-problem that aims to maximize long-term energy-efficiency (EE).
- The non-convex VEFL parameter optimization sub-problems are solved using standard relaxations and the difference between convex (DC) approach. The fractional non-convex long-term EE optimization problem is first transformed into a tractable form using the Dinkelbach method, which is further converted into a per-slot online optimization problem leveraging Lyapunov drift-plus-penalty-based stochastic optimization.

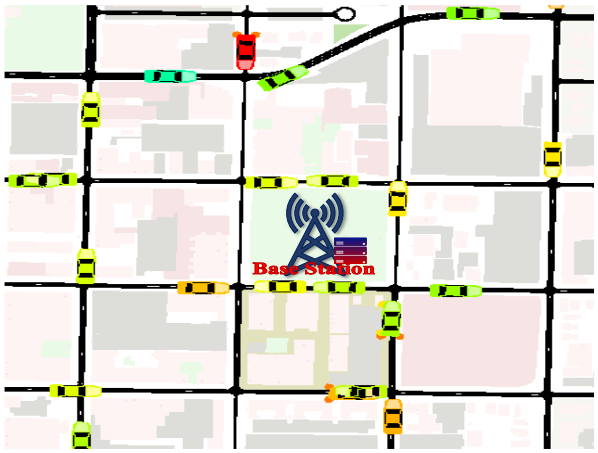

Fig. 1. Vehicular FL system model.

- Finally, using simulation of urban mobility (SUMO) [30], we simulate a microscopic mobility scenario in Downtown Raleigh, NC, USA, and use four popular ML datasets to show the effectiveness of our proposed solutions.

The rest of the paper is organized as follows: Section II introduces our proposed VEFL system model. Section III provides the convergence analysis and our joint problem formulation. Section IV presents the solution to the problem. We discuss our simulation results in Section V. Finally, Section VI concludes the paper.

II. VEFL System Model

We consider a VEN confined within a region of interest (RoI), as shown in Fig. 1. The edge server—embedded into the next generation Node B (gNB)—wishes to perform a distributed ML task leveraging the moving CVs' onboard CPUs and local datasets. Similar to [31], [32], SLAs between the CVs and the edge server, which require the edge server to pay the CVs for contributing to the VEFL task, are assumed.[1] Note that the terms server and gNB are used interchangeably when there is no ambiguity. We consider a general learning task, which can be object detection, traffic sign classification/detection, velocity/acceleration prediction, traffic congestion prediction, travel time prediction, fuel consumption prediction, etc., for our VEFL. Moreover, the VEN operates in a discrete time-slotted manner. The slots are denoted by $\mathscr{T} = \{t\}_{t=1}^{|\mathscr{T}|}$. Particularly, the gNB has a fixed BWP for the VEFL to provide radio connectivity to the moving client CVs. The server can only leverage the trained ML model on the CVs' onboard CPUs within the communication range of the gNB. Denote the communication radius of the gNB by r.

The CVs enter and leave the RoI following some distributions, i.e., the CV set may not be the same in all time slots due to high mobility. Denote the CV set during time slot t by $\mathscr{V}_t = \{v\}_{v=1}^{V_t}$. Moreover, the server knows the maximum possible velocity on the roads inside this RoI. Denote the maximum possible velocity of the RoI by $\mathrm{u}^{\max}$. Assuming the gNB is located at the center of the coordinate system of the RoI, its coverage circle is given by

$$x^2 + y^2 = \mathrm{r}^2, \tag{1}$$

where x and y are the horizontal and vertical coordinates, respectively.

A. CV Mobility Model

Denote the coordinate of $v \in \mathscr{V}_t$, during slot t, by $\left(\mathrm{x}_v^{\mathrm{loc}}(t), \mathrm{y}_v^{\mathrm{loc}}(t)\right)$ as shown in Fig. 2. Also, let us denote CV v's velocity and acceleration during time t by $u_v(t)$ and $\dot{u}_v(t)$, respectively. We consider the widely used car-following mobility (CFP) model, known as the intelligent driver model (IDM) mobility model, to model microscopic mobility for the CVs [35]. In IDM, the mobility is controlled by the following instantaneous acceleration equation [35]:

$$\dot{u}_v(t) = \bar{\mathrm{u}}_1 \left(1 - [u_v(t)/\mathrm{u}^{\max}]^4\right) - \bar{\mathrm{u}}_1 [s_v^*/\Delta d_v(t)\ ]^2, \tag{2}$$

where $\bar{\mathrm{u}}_1$ is the maximum acceleration or a constant that depends on the design. $\Delta d_v(t)$ is the front bumper to the back bumper distance of CV v and the vehicle directly in front of it, during time t. Furthermore, s_v^* is the desired dynamical distance and is calculated as follows:

$$s_v^* = s^{\mathrm{saf}} + u_v(t) \cdot t^{\mathrm{dst}} + [u_v(t) \cdot \Delta u_v(t)]/(2\sqrt{\bar{\mathrm{u}}_1 \bar{\mathrm{u}}_2}), \tag{3}$$

where s^{saf} is the safety distance between v and the CV directly in front of it, t^{dst} is the desired time headway that gives the minimum possible time to the CV directly in front of v, $\Delta u_v(t)$ is the velocity difference between v and the vehicle in front of it, and $\bar{\mathrm{u}}_2$ is a positive number that defines the comfortable braking deceleration. Note that, in (2), the first term, i.e., $\bar{\mathrm{u}}_1 \left(1 - [u_v(t)/\mathrm{u}^{\max}]^4\right)$, is the instantaneous acceleration of CV v, which essentially is the desired acceleration on a free road. The second term is the deceleration induced by the CV in front of it [35].

Moreover, we assume the server does not need to know the entire trajectory of the vehicle. Therefore, the server will only estimate the guaranteed sojourn period $\mathrm{t}_{v,t}^{\mathrm{soj}}$. To find $\mathrm{t}_v^{\mathrm{soj}}(t)$, first, we write the horizontal and vertical lines that intersect the gNB' radio coverage circle boundary as follows:

$$y = \mathrm{y}_v^{\mathrm{loc}}(t), \tag{4}$$

$$x = \mathrm{x}_v^{\mathrm{loc}}(t). \tag{5}$$

Note that (4) and (5) are represented by the purple and orange color chords, respectively, in Fig. 2. We can then find the x-coordinates of the gNB coverage boundary by solving (1) and (4) as $x_v^{\mathrm{bnd},1}(t) = \sqrt{\mathrm{r}^2 - \mathrm{y}_v^{\mathrm{loc}}(t)^2}$ and $x_v^{\mathrm{bnd},2}(t) = -\sqrt{\mathrm{r}^2 - \mathrm{y}_v^{\mathrm{loc}}(t)^2}$. As such, we can find the horizontal distances of these boundary points and the current location of the CV $(\mathrm{x}_v^{\mathrm{loc}}(t), \mathrm{y}_v^{\mathrm{loc}}(t))$ as $\mathrm{d}_{x_1}^y = \left|\mathrm{x}_v^{\mathrm{loc}}(t) - \mathrm{x}_v^{\mathrm{bnd},1}(t)\right|$ and $\mathrm{d}_{x_2}^y = \left|\mathrm{x}_v^{\mathrm{loc}}(t) - \mathrm{x}_v^{\mathrm{bnd},2}(t)\right|$. Similarly, we can find the y-coordinates of the gNB coverage boundary by solving (1) and (5) as $y_v^{\mathrm{bnd},1}(t) = \sqrt{\mathrm{r}^2 - \mathrm{x}_v^{\mathrm{loc}}(t)^2}$ and

[1] However, the implementation of SLAs in a practical VEN implicates the core network and the user plane and control plane protocol stacks [33], [34], which are beyond the scope of this paper.

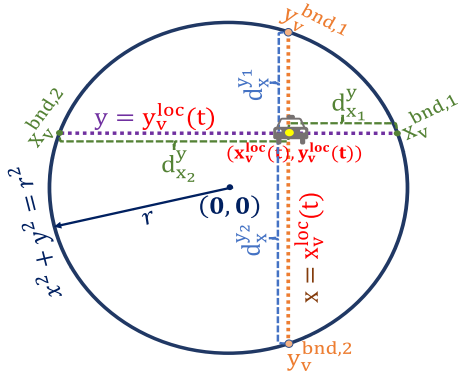

Fig. 2. Finding coverage boundary points.

$y_v^{\mathrm{bnd},2}(t) = -\sqrt{\mathrm{r}^2 - \mathrm{x}_v^{\mathrm{loc}}(t)^2}$. Then, the corresponding vertical distances of these boundary points and the current location of the CV $(\mathrm{x}_v^{\mathrm{loc}}(t), \mathrm{y}_v^{\mathrm{loc}}(t))$ are calculated as $\mathrm{d}_x^{y_1} = \left|\mathrm{y}_v^{\mathrm{loc}}(t) - \mathrm{y}_v^{\mathrm{bnd},1}(t)\right|$ and $\mathrm{d}_x^{y_2} = \left|\mathrm{y}_v^{\mathrm{loc}}(t) - \mathrm{y}_v^{\mathrm{bnd},2}(t)\right|$.

As such, from time t, the minimum expected sojourn period of v under the gNB's coverage is bounded below by

$$\mathrm{t}_v^{\mathrm{soj}}(t) \geq \min\left\{\mathrm{d}_{x_1}^{y}, \mathrm{d}_{x_2}^{y}, \mathrm{d}_x^{y^1}, \mathrm{d}_x^{y^2}\right\}/\mathrm{u}^{\max}. \tag{6}$$

Note that (6) is based on a linear trajectory and uniform velocity $\mathrm{u}^{\max}$, which is the worst-case estimation of the sojourn period of the CV.

B. V2I Radio Access Technology Model

We assume that the VEN operates in TDD mode and has a fixed W Hz BWP for providing RAT connectivity over the universal mobile telecommunications system (UMTS) air interface (Uu interface) for the VEFL. Note that TDD facilitates channel reciprocity and thus offers less control overhead. Besides, we assume that all transceivers can mitigate the Doppler effect satisfactorily.[2] For model distribution in the downlink, as the gNB sends the same data to all selected CVs, it can use the entire spectrum and high transmission power to broadcast the model. Therefore, similar to [4], [8], [19], we ignore the downlink communication. The W Hz bandwidth is divided into orthogonal physical resource blocks (pRBs). Denote the pRB set of the allocated BWP by the set $\mathscr{Z} = \{z\}_{z=1}^{Z}$. Note that due to orthogonal pRBs, there is no intra-cell interference. Besides, as we consider a single cell, the proposed VEN is interference-free.

The gNB considers the 5G-NR frame structure in which the radio frame is 10 ms long with 10 subframes. Within each 1 ms subframe, there are $2^{\bar{n}}$ slots, where $\bar{n}$ is the sub-carrier spacing numerology [36]. The pRB allocation *granularity* is the *slot*, i.e., the pRBs can be allocated to different users in each slot t. Each slot carries 14 OFDM symbols in the time domain and 12 sub-carriers in the frequency domain. Moreover, the OFDM symbols can be configured based on the duplexing mode. As this is uplink-heavy transmission, we consider a downlink-control uplink transmission slot format as shown in Fig. 3. As such, we consider the effective uplink data rate per slot, which will be fleshed out in what follows.

We consider a single-input-multiple-output (SIMO) case, where the gNB has N antennas, and each CV has a single antenna. Note that our framework can also be extended to other multiple-antenna communication models.[3] During slot t, denote the channel between CV v and the n^{th} antenna of the gNB over pRB z by $h_{n,v,z}(t)$. Then, we denote the entire channel response at VU v from the gNB over PRB z as

$$\mathbf{h}_{v,z}(t) = \sqrt{\psi_v(t)}\varrho_v(t)\breve{\mathbf{h}}_{v,z}(t) \in \mathbb{C}^{N\times 1}, \tag{7}$$

where $\sqrt{\psi_v(t)}$, $\varrho_v(t)$ and $\breve{\mathbf{h}}_{v,z}(t) = [h_{1,v,z}(t), \ldots, h_{N,v,z}(t)]^T \in \mathbb{C}^{N\times 1}$ are large scale fading, log-Normal shadowing and fast fading channel responses from the N antennas, respectively. The path losses are modeled based on the urban macro (UMa) model [37].

To that end, denote CV v's unit powered intended signal for the gNB by $s_v(t) \in \mathbb{C}$ and allocated uplink transmission power for pRB z by $P_{v,z}(t)$. Assuming receiver beamforming vector $\mathbf{g}_{v,z}(t) \in \mathbb{C}^{N\times 1}$, the effective uplink signal received at the gNB, over pRB z, is calculated as follows:

$$y_{v,z}(t) = \mathrm{I}_{v,z}(t) \cdot \sqrt{P_{v,z}(t)}\mathbf{g}_{v,z}(t)^H \mathbf{h}_{v,z}(t)s_v(t) + \eta, \tag{8}$$

where $\mathrm{I}_{v,z}(t) \in \{0,1\}$ is an indicator function that takes value 1 when pRB z is allocated to CV v, and $\eta \sim CN(0, \varsigma^2)$ is the circularly symmetric zero mean Gaussian distributed random variable with variance ς^2.

Then, we calculate the received signal-to-noise ratio (SNR) at the gNB for CV v's uplink transmission as follows:

$$\Gamma_{v,z}(t) = \left(\mathrm{I}_{v,z}(t) \cdot P_{v,z}(t)\left|\mathbf{g}_{v,z}(t)^H \mathbf{h}_{v,z}(t)\right|^2\right)/(\omega\varsigma^2), \tag{9}$$

where ω is the pRB size. The gnB can configure CV-specific CSI reference signal (RS) to estimate the channels. Since 5G-NR has the flexibility of configuring CSI-RS periodically, semi-persistently or aperiodically and may also perform uplink channel information multiplexing on the physical uplink shared channel [38], this work primarily focuses on the overall VEFL framework considering CSI is known at the gNB.[4] Therefore, the gNB can use maximal ratio combining receiver beamforming to get $\mathbf{g}_{v,z}(t) = \mathbf{h}_{v,z}(t)/\left\|\mathbf{h}_{v,z}(t)\right\|$, which gives the following uplink SNR over pRB z.

$$\Gamma_{v,z}(t) = \left(\mathrm{I}_{v,z}(t) \cdot P_{v,z}(t)\left\|\mathbf{h}_{v,z}(t)\right\|^2\right)/(\omega\varsigma^2). \tag{10}$$

To this end, we can calculate the achievable data rate at the gNB from CV v's uplink as follows:

$$r_v(t) = \omega(1-\upsilon) \cdot \mathrm{I}_v(t) \sum\nolimits_{z=1}^{Z} \mathbb{E}_{\mathbf{h}}\left[\log_2(1+\Gamma_{v,z}(t))\right], \tag{11}$$

where $\mathrm{I}_v(t) \in \{0,1\}$ is an indicator function that takes value 1 when CV $v \in \mathscr{V}_t$ is scheduled for transmissions in slot t and the expectation is over the channel $\mathbf{h}_{v,z}(t)$. Besides, υ

[2]Although Doppler shift is a well-known problem, if the underlying RAT mitigates it adequately, it is less critical for our proposed VEFL framework.

[3]However, the TDD-based massive MIMO requires rigorous channel estimation/equalization, beam management, etc., which deserve separate studies.

[4]Channel estimation delay is a part of the RAT and is less critical for our proposed VEFL framework as the server only uses the worst-case estimated channel, discussed in Section IV, to determine the approximate upper bound for the uplink model offloading delay.

1 Slot = 14 OFDM Symbols													
Symb 0	Symb 1	Symb 2	Symb 3	Symb 4	Symb 5	Symb 6	Symb 7	Symb 8	Symb 9	Symb 10	Symb 11	Symb 12	Symb 13
DL	Fexible	UL	UL	UL	UL	UL	UL	UL	UL	UL	UL	UL	UL

Fig. 3. OFDM Symbols within a slot [36].

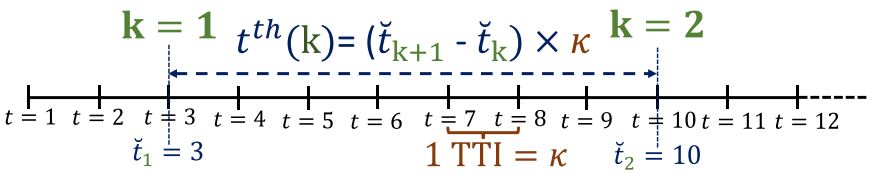

Fig. 4. Time slot orchestration in the proposed VEFL.

is the loss due to control signaling overhead. For our case, if the *flexible* symbol in Fig. 3 is allocated for uplink, we set $\upsilon = 1/14$. Moreover, if it is not assigned to uplink, we set $\upsilon = 2/14$.

C. Preliminaries of VEFL

Denote the server's global ML model by $\boldsymbol{\omega}$. Denote the dataset available at CV v by $\mathscr{D}_v = \{\mathbf{x}_i, y_i\}_{i=1}^{|\mathscr{D}_v|}$, where $\mathbf{x}_i$ and y_i are the i^{th} sample feature and the corresponding label, respectively. Therefore, during time t, the entire dataset can be denoted as $\mathscr{D}_t = \{\mathscr{D}_v\}_{v=1}^{V_t}$. The edge server aims to optimize

$$\min_{\boldsymbol{\omega}} F(\boldsymbol{\omega}) = \sum\nolimits_{v=1}^{V_t} p_v f_v(\boldsymbol{\omega}), \tag{12}$$

where $p_v \in [0,1]$ is the linear combination weight for CV v with $\sum_{v=1}^{V_t} p_v = 1$. While the typical FedAvg set $p_v = |\mathscr{D}_v|/|\mathscr{D}_t|$ [3], we will explore more on how to properly set these weights in what follows. Besides, $f_v(\boldsymbol{\omega})$ is the local empirical loss function of CV v.

The server distributes the global model during the VEFL global rounds $k = 1, \ldots, K$ as shown in Fig. 4. Denote the slots corresponding to the global rounds by the set $\mathscr{T}_g = \{\check{t}_k\}_{k=1}^{K}$, where $\check{t}_1$ represents the VEN slot t at which $k = 1$. For example, in Fig. 4, $\check{t}_1 = 3$ and $\check{t}_2 = 10$. Note that throughout our discussions, we will use the notation k and term *round* to represent *VEFL global round*, while the notation t and term *slot* will represent the *discrete time slot* of the VEN.

Denote the duration between global round $k+1$ and k by

$$\mathrm{t}^{\mathrm{th}}(k) = \kappa \times \left(\check{t}_{k+1} - \check{t}_k\right), \tag{13}$$

where κ is the transmission time interval (TTI). Besides, $\mathscr{T}_g \subset \mathscr{T}$ is known to all CVs that have SLAs with the server. Denote the available CV pool during global round k by $\mathscr{V}_{\check{t}_k}$. Without any loss of generality, denote the global model at the server during global round k by $\boldsymbol{\omega}_k$. The server then broadcasts $\boldsymbol{\omega}_k$ to all CV $v \in \mathscr{V}_{\check{t}_k}$. Denote the local copy of CV v's model at the beginning of round k by $\boldsymbol{\omega}_{v,k}$, i.e., $\boldsymbol{\omega}_{v,k} \leftarrow \boldsymbol{\omega}_k$ for all $v \in \mathscr{V}_{\check{t}_k}$. Upon receiving the global model, each $v \in \mathscr{V}_{\check{t}_k}$ performs local model training to minimize the following objective function

$$\min_{\boldsymbol{\omega}} f_v(\boldsymbol{\omega}, \boldsymbol{\omega}_k) = F_v(\boldsymbol{\omega}) + (\mu/2)\|\boldsymbol{\omega} - \boldsymbol{\omega}_k\|^2, \tag{14}$$

where $F_v(\boldsymbol{\omega})$ is CV v's local empirical loss function on its dataset $\mathscr{D}_v$. An L_2 regularization is added to $F_v(\boldsymbol{\omega})$ to tackle heterogeneity that often arises in FL.

Each $v \in \mathscr{V}_{\check{t}_k}$ trains its local model on its local dataset $\mathscr{D}_v$ for $l_v(k)$ iterations and obtains the following updated model

$$\boldsymbol{\omega}_{v,k+1} = \boldsymbol{\omega}_{v,k} - \delta \sum\nolimits_{l=1}^{l_v(k)} \nabla f_v(\boldsymbol{\omega}, \boldsymbol{\omega}_{v,k}), \tag{15}$$

where δ is the step size. After this local training, the CVs send $\boldsymbol{\omega}_{v,k+1}$'s to the server. The server then performs weight aggregations, which will be discussed in more detail in the following section, and computes the updated global model $\boldsymbol{\omega}_{k+1}$ for the next round. The above processes repeat until the globally trained model reaches an expected level of precision.

D. Delay Calculation

1) Model Training Delay: Denote CV v's CPU computation cycle frequency by $\eta_v(k) \in \{\eta_v^{\min}, \eta_v^{\max}\}$. Assuming CV v requires c_v CPU cycles to process per-bit data, the local computation delay for one local iteration is calculated as [4]

$$\mathrm{t}_{v,\mathrm{itr}}^{\mathrm{cmp}} = c_v \mathrm{D}_v / \eta_v(k), \tag{16}$$

where D_v is the dataset size of v in bits. Then, to perform $l_v(k)$ local iterations during global round k, the total local computation delay of CV v is

$$\mathrm{t}_v^{\mathrm{cmp}}(k) = l_v(k) \cdot \mathrm{t}_{v,\mathrm{itr}}^{\mathrm{cmp}}. \tag{17}$$

2) Communication Delay: Let the model parameters be of M dimension, i.e., $\boldsymbol{\omega} \in \mathbb{R}^M$. Then, the required number of bits for transmitting the model parameters is calculated as

$$S(\boldsymbol{\omega}, \mathrm{FPP}) = \sum\nolimits_{m=1}^{M} [1 + \mathrm{FPP}(m)], \tag{18}$$

where $\mathrm{FPP}(m)$ represents floating point precision. Note that besides $\mathrm{FPP}(m)$ bits to send the m^{th} element of $\boldsymbol{\omega}$, we need 1 additional bits to represent its sign.

This model payload size $S(\boldsymbol{\omega}, \mathrm{FPP})$ is usually on the scale of megabits (Mbs) and takes more than one coherence time and frequency block. Therefore, if a CV starts model offloading from slot $t > \check{t}_k$ after it finishes the local model training during VEFL global round k, the offloading delay is calculated as

$$\mathrm{t}_v^{\mathrm{tx}}(k) = \kappa \cdot \min\left\{T : \kappa(1-\upsilon) \sum\nolimits_{\bar{t}=t}^{T} r_v(\bar{t}) \geq S(\boldsymbol{\omega}, \mathrm{FPP}), T \in \mathbb{Z}^+\right\}. \tag{19}$$

For CV $v \in \mathscr{V}_{\check{t}_k}$, the total delay between two VEFL global rounds is, thus, calculated as follows:

$$\mathrm{t}_v(k) = \mathrm{t}_v^{\mathrm{cmp}}(k) + \mathrm{t}_v^{\mathrm{tx}}(k). \tag{20}$$

E. Energy Consumption

1) Local Model Training Energy Consumption: We calculate the energy consumption of CV $v \in \mathscr{V}_{\check{t}_k}$ for its local model training during round k as

$$\mathrm{e}_v^{\mathrm{cmp}}(k) = l_v(k) \cdot (\zeta/2) c_v \mathrm{D}_v \eta_v(k)^2, \tag{21}$$

where $\zeta/2$ is the effective capacitance of CV v's CPU chip.

2) Communication Energy Consumption: We calculate the uplink transmission energy consumption of the client $v \in \mathscr{V}_{\check{t}_k}$ during round k as

$$\mathrm{e}_v^{\mathrm{tx}}(k) = \sum\nolimits_{\bar{t}=t}^{T} \sum\nolimits_{z=1}^{Z} P_{v,z}(\bar{t}), \tag{22}$$

where T is calculated is (19).

Therefore, we calculate the total energy consumption associated with client v's local computation and uplink communication as follows:

$$\mathrm{e}_v^{\mathrm{tot}}(k) \ = \mathrm{e}_v^{\mathrm{cmp}}(k) + \mathrm{e}_v^{\mathrm{tx}}(k). \tag{23}$$

III. VEFL Convergence Analysis and Problem Formulation

A. Convergence Analysis

We make the following standard assumptions [23], [39].

Assumption 1 (L-Lipschitz Gradient): For all $v \in \mathscr{V}_{\check{t}_k}$ in all k, $F_v(\boldsymbol{\omega})$ is L-Lipschitz gradient for any two parameter vectors $\boldsymbol{\omega}$ and $\boldsymbol{\omega}'$, i.e., $\|\nabla F_v(\boldsymbol{\omega}) - \nabla F_v(\boldsymbol{\omega}')\| \leq L \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|$.

Assumption 2 (B-Dissimilar Gradients): The local gradient at the CVs are at most B-dissimilar from the global gradient $\nabla f(\boldsymbol{\omega})$, i.e., $\|\nabla F_v(\boldsymbol{\omega})\| \leq B \|\nabla f(\boldsymbol{\omega})\|$, for all v.

Assumption 3 (σ-Bounded Hessian): The smallest eigen value of the Hessian matrix is $-\sigma$ for all CVs, i.e., $\nabla^2 F_v \succeq -\sigma \mathbf{I}$. This also implies that the $f_v(\boldsymbol{\omega}, \boldsymbol{\omega}_k)$ in (14) is $\mu' = \mu - \sigma$ strongly convex.

Assumption 4 (γ-Inexact Local Solvers): Local update of the CV v results in γ-inexact solution $\boldsymbol{\omega}_{v,k+1}$ of (14) in all global round k. In other words, the local update of CV v yields $\|\nabla f_v(\boldsymbol{\omega}_{v,k+1}, \boldsymbol{\omega}_k)\| \leq \gamma \|\nabla f_v(\boldsymbol{\omega}_k, \boldsymbol{\omega}_k)\|$, where $\gamma \in [0,1]$. Note that $\gamma = 0$ means solving (14) optimally, while an increased value indicates how the updated model differs from the exact solution.

Additionally, we consider full-batch gradient descent, i.e., each CV performs its model training in its entire dataset, which can be extended to stochastic gradient descent.

Denote the successful trained local model reception from the client v during global round k by

$$\mathbf{1}\left(\mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right) = \begin{cases} 1, & \text{with probability } p_v^{\mathrm{suc}}(k), \\ 0, & \text{with probability } (1 - p_v^{\mathrm{suc}}(k)), \end{cases} \tag{24}$$

where $d_v(k)$ is the distance of v after performing $l_v(k)$ iterations from the gNB.

We consider two cases for device participation, namely, the FDPC and the PDPC. While all CVs with SLAs participate in model training for the former scenario, only a subset of these CVs participates in the latter. In both cases, for the model aggregation, the server can take a client's locally trained model $\boldsymbol{\omega}_{v,k+1}$ only when it successfully receives it within the deadline threshold $\mathrm{t}^{\mathrm{th}}(k)$. As such, for FDPC, we express the server's aggregation rule as

$$\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k + \sum_{v=1}^{V_{\check{t}_k}} \left[\frac{p_v \cdot \mathbf{1}\left(\mathrm{t}_v(k) \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)}{p_v^{\mathrm{suc}}(k)} \times (\boldsymbol{\omega}_{v,k+1} - \boldsymbol{\omega}_k) \right], \tag{25}$$

where $p_v \in (0,1]$ and $\sum_{v=1}^{V_{\check{t}_k}} p_v = 1$. Moreover, $p_v = \bar{p}_v / \sum_{v=1}^{V_{\check{t}_k}} \bar{p}_v$, where $\bar{p}_v$ is calculated as follows:

$$\bar{p}_v = (1-\lambda)[\mathrm{D}_v / \sum\nolimits_{v=1}^{V_{\check{t}_k}} \mathrm{D}_v] + \lambda [\mathrm{t}_v^{\mathrm{soj}}(\check{t}_k) / \sum\nolimits_{v=1}^{V_{\check{t}_k}} \mathrm{t}_v^{\mathrm{soj}}(\check{t}_k)], \tag{26}$$

where $\lambda \in [0,1]$ is a parameter that balances the weight of the sojourn period and dataset size.

Note that the averaging step in (25) is unbiased, i.e., $\mathbb{E}[\boldsymbol{\omega}_{k+1}] = \sum_{v=1}^{V_{\check{t}_k}} p_v \cdot \boldsymbol{\omega}_{v,k+1}$. Besides, we adopt this aggregation weight p_v inspired by the complementary cumulative distribution function (CCDF) of the CVs' expected sojourn periods $\mathrm{t}_v^{\mathrm{soj}}(\check{t}_k)$'s. Particularly, at high speed, $\mathrm{t}_v^{\mathrm{soj}}(\check{t}_k)$ can be very small, as shown in Fig. 5. For example, at a maximum velocity $\mathrm{u}^{\max}$ of 26.82 meter/second (m/s), about 28% CVs have an expected sojourn period smaller than 5 seconds. Besides, about 13% CVs have smaller than 2.5 seconds sojourn periods. As such, the aggregation rule shall also weigh this short sojourn period since these CVs are likely to execute only a few local iterations compared to the other CVs with relatively higher sojourn periods.

One popular way to combat the straggler effect is to aggregate model parameters upon the reception of the first $|\mathscr{C}_k|$, where $\mathscr{C}_k \subseteq \mathscr{V}_{\check{t}_k}$ [40]. In federated edge learning (FEEL), RAT resource is limited. Besides, the CVs are required to perform their own computational tasks, while the server is required to pay a fee if it selects a CV for model training. As such, it is practical to select the subset $\mathscr{C}_k$ at the beginning of VEFL round k. In this pragmatic case, we thus assume that the server uniformly selects $|\mathscr{C}_k|$ CVs out of the original $V_{\check{t}_k}$ CVs without replacement in each VEFL round k. Moreover, we stress that the server decides how many total CVs it can select based on its limited monetary and bandwidth budgets. In other words, we leave this $|\mathscr{C}_k|$ as a design parameter that

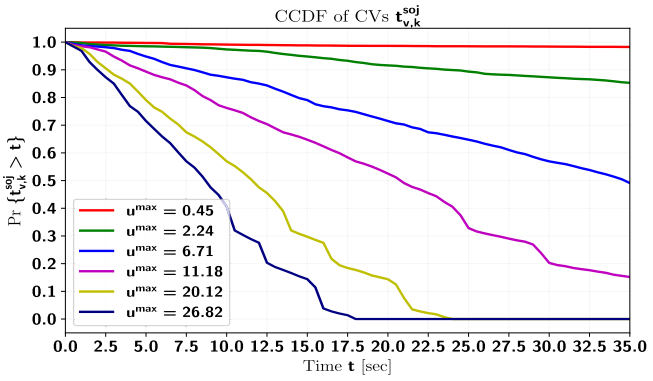

Fig. 5. CCDF of CVs expected sojourn period.

the server determines contingent on the resource-constraint circumstances. Denote the event that CV v is selected during VEFL round k by

$$\mathbf{1}\left(v \in \mathscr{C}_k\right) = \begin{cases} 1, & \text{with probability } q_v(k) \\ 0, & \text{with probability } (1 - q_v(k)). \end{cases} \tag{27}$$

Considering the above factors, the server chooses the following aggregation rule for PDPC.

$$\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k + \sum_{v=1}^{V_{\check{t}_k}} \Big[p_v \cdot \frac{\mathbf{1}\left(v \in \mathscr{C}_k, \mathrm{t}_v(k) \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)}{q_v(k) p_v^{\mathrm{suc}}(k)} \times \left(\boldsymbol{\omega}_{v,k+1} - \boldsymbol{\omega}_k\right) \Big]. \tag{28}$$

When all selected CVs have the same p_v, the averaging step in (28) is unbiased, i.e., $\mathbb{E}\left[\boldsymbol{\omega}_{k+1}\right] = \sum_{v=1}^{V_{\check{t}_k}} p_v \cdot \boldsymbol{\omega}_{v,k+1}$. The VEFL algorithm is summarized in Algorithm 1. The convergence analysis for PDPC is derived in Theorem 1.

Theorem 1: Using our assumptions and aggregation rule (28), the expected loss decrease after one VEFL round is

$$\mathbb{E}\left[f(\boldsymbol{\omega}_{k+1})\right] - f(\boldsymbol{\omega}_k) \leq \left(B\left(1+\gamma\right)/\mu'\right) \times \left(1 + \frac{BL(1+\gamma)}{2\mu'} \sum_{v=1}^{V_{\check{t}_k}} \frac{p_v}{q_v(k) p_v^{\mathrm{suc}}(k)}\right) \left\|\nabla f(\boldsymbol{\omega}_k)\right\|^2. \tag{29}$$

Proof: The proof is left in Appendix A. ∎

Remark 1: From the convergence analysis in (29), we observe that p_v, $q_v(k)$, $p_v^{\mathrm{suc}}(k)$ and $\|\nabla f(\boldsymbol{\omega}_k)\|$ may vary in each VEFL round k. Besides, it is generally impossible to find a closed-form expression for $\|\nabla f(\boldsymbol{\omega}_k)\|$ by relating it to the learning parameters [6], [41]. As such, in each VEFL round, we aim to sub-optimally minimize the loss in (29) by concentrating on $\sum_{v=1}^{V_{\check{t}_k}} p_v/[q_v(k)p_v^{\mathrm{suc}}(k)]$.

Remark 2: Note that FDPC is a special case of PDPC, where all CVs participate in the model training. Moreover, the convergence analysis and problem formulation are similar.

B. Problem Formulation

In our proposed VEFL, we explicitly consider CVs mobility, system heterogeneity—in terms of CPU frequency and energy budget—across CVs, intermittent Uu links, and the server's limited monetary and radio budgets. Denote the fixed monetary budget of the server for global round k by $\Xi(k)$ unit. Denote the cost for each unit of energy by ϕ_v unit/Joule. The energy consumption of a CV depends on its number of local iterations $l_v(k)$, CPU frequency $\eta_v(k)$ and total uplink transmission power to offload the trained model $\boldsymbol{\omega}_{v,k+1}$. Particularly, as part of the SLA, each CV first reports its minimum CPU frequency $\eta_v^{\min}$, maximum CPU frequency $\eta_v^{\max}$, maximum transmission power $P_v^{\max}$, energy budget $\mathrm{e}_v^{\mathrm{bud}}(k)$, dataset size D_v and charging policy $\xi_v(l_v(k), \eta_v(k))$ to the server. More specifically, the CVs have the following charging policy.

$$\xi_v(l_v(k), \eta_v(k)) = \mathrm{e}_v^{\mathrm{tot}}(k)\ \phi_v + \bar{\phi}_v, \tag{30}$$

where $\mathrm{e}_v^{\mathrm{tot}}(k)$ is calculated in (23). As such, $\mathrm{e}_v^{\mathrm{tot}}(k)\ \phi_v$ is the CV's expected operational cost, and $\bar{\phi}_v$ is a constant term to ensure its gain upon participating in model training.

Algorithm 1 Vehicular Edge Federated Learning

Input: Total global round K
for $k = 1$ *to* K **do**
  Broadcast $\boldsymbol{\omega}_k$ to the client CV set $\mathscr{Y}_k$
    /* $\mathscr{Y}_k = \mathscr{V}_{\check{t}_k}$ and $\mathscr{Y}_k = \mathscr{C}_k$, respectively for FDPC and PDPC */
  for $v \in \mathscr{Y}_k$ *in parallel* **do**
    Train the received global model for $l_v(k)$ local iterations to get the locally trained model $\boldsymbol{\omega}_{v,k+1}$
    Offload $\boldsymbol{\omega}_{v,k+1}$ back to the server using the allocated TTIs and pRBs by the gNB
  end
  Server aggregates client CVs models to get the updated global model $\boldsymbol{\omega}_{k+1}$ using (25) and (28), respectively, for FDPC and PDPC
end
Output: Global ML model $\boldsymbol{\omega}_K$

To this end, denote $\mathbf{1}(k) = \left[\mathbf{1}\left(1 \in \mathscr{C}_k\right), \ldots, \mathbf{1}\left(V_{\check{t}_k} \in \mathscr{C}_k\right)\right]^T \in \mathbb{R}^{V_{\check{t}_k}}$, $\boldsymbol{l}(k) = [l_1(k), \ldots, l_{V_{\check{t}_k}}(k)]^T \in \mathbb{R}^{V_{\check{t}_k}}$, $\boldsymbol{\eta}(k) = \left[\eta_1(k), \ldots, \eta_{V_{\check{t}_k}}(k)\right]^T \in \mathbb{R}^{V_{\check{t}_k}}$, $\mathbf{I}(t) = [\mathrm{I}_1(t), \ldots, \mathrm{I}_{V_{\check{t}_k}}(t)]^T \in \mathbb{R}^{V_{\check{t}_k}}$, $\check{\mathbf{I}}_v(t) = [\mathrm{I}_{v,1}(t), \ldots, \mathrm{I}_{v,Z}(t)]^T \in \mathbb{R}^Z$, $\check{\mathbf{I}}(t) = [\check{\mathbf{I}}_1(t), \ldots, \check{\mathbf{I}}_{V_{\check{t}_k}}(t)]^T \in \mathbb{R}^{V_{\check{t}_k} \times Z}$, $\mathbf{p}_v(t) = [P_{v,1}(t), \ldots, P_{v,Z}(t)]^T \in \mathbb{R}^Z$ and $\mathbf{p}(t) = [\mathbf{p}_1(t), \ldots, \mathbf{p}_{V_{\check{t}_k}}(t)]^T \in \mathbb{R}^{V_{\check{t}_k} \times Z}$. Based on Theorem 1 and Remark 1, in each VEFL round k, we want to sub-optimally minimize the loss by minimizing the relaxed objective $\sum_{v=1}^{V_{\check{t}_k}} p_v/[q_v(k)p_v^{\mathrm{suc}}(k)]$. Note that $q_v(k)$ in the denominator is essentially the expectation of CV selection $\mathbf{1}(v \in \mathscr{C}_k)$, which is a VEFL parameter. Besides, p_v^{suc} is the CV's local model reception successful probability that depends on the RAT configurations. Instead of solving $\sum_{v=1}^{V_{\check{t}_k}} p_v/[q_v(k)p_v^{\mathrm{suc}}(k)]$ directly, we pose it as relaxed a linear objective $\sum_{v=1}^{V_{\check{t}_k}} \mathbf{1}(v \in \mathscr{C}_k) \cdot p_v^{\mathrm{suc}}(k)$ that we want to maximize. Note that maximization of total user participation in a similar fashion is also common in the literature [42], [43], [44]. Besides, this practical objective function also works well in simulation. Recall that CSI is not fixed, the server has a limited monetary budget, and the gNB has a limited BWP. On the one hand, knowing $p_v^{\mathrm{suc}}(k)$ at the beginning of a VEFL round is impossible. On the other hand, the CV selection indicator functions appear in many of our constraints in the following. Therefore, using the binary CV selection indicator function $\mathbf{1}(v \in \mathscr{C}_k)$ in the objective function serves two primary purposes: congruent problem decomposition in the sequel and elimination of optimization parameter $q_v(k)$. As such, we want to solve the following relaxed problem.

$$\underset{\mathbf{1}(k), \boldsymbol{l}(k), \boldsymbol{\eta}(k), \mathbf{I}(t), \check{\mathbf{I}}(t), \mathbf{p}(t)}{\text{maximize}} \quad \sum_{v=1}^{V_{\check{t}_k}} \mathbf{1}(v \in \mathscr{C}_k) \cdot p_v^{\mathrm{suc}}(k), \tag{31}$$

$$\text{s.t. } (C_1) \quad \sum_{v=1}^{V_{\check{t}_k}} \mathbf{1}(v \in \mathscr{C}_k) = |\mathscr{C}_k|, \tag{31a}$$

$$(C_2) \quad \mathbf{1}(v \in \mathscr{C}_k)\mathrm{t}_v(k) \leq \mathrm{t}^{\mathrm{th}}(k), \ \forall v, \ \forall k, \tag{31b}$$

$$(C_3) \quad \mathbf{1}(v \in \mathscr{C}_k)\mathrm{t}_v(k) \leq \mathbf{1}(v \in \mathscr{C}_k)\mathrm{t}_v^{\mathrm{soj}}(\check{t}_k), \ \forall v, \ \forall k, \tag{31c}$$

$$(C_4)\ \ \mathbf{1}(v \in \mathscr{C}_k)\mathrm{e}_v^{\mathrm{tot}}(k) \le \mathbf{1}(v \in \mathscr{C}_k)\mathrm{e}_v^{\mathrm{bud}}(k),\ \forall v,\ \forall k, \quad (31d)$$

$$(C_5)\ \ \sum\nolimits_{v=1}^{V_{\check{t}_k}} \mathbf{1}(v \in \mathscr{C}_k)\xi(l_v(k),\eta_v(k)) \le \Xi(k),\ \forall k, \quad (31e)$$

$$(C_6)\ \ \mathbf{1}(v \in \mathscr{C}_k)\cdot l_v(k)\ \ge l_k^{\mathrm{des}},\ l_v(k)\ \in \mathbb{Z}^+,\ \forall k,\ \forall v \quad (31f)$$

$$(C_7)\ \ \sum\nolimits_{v\in\mathscr{C}_k} \mathrm{I}_{v,z}(t) = 1,\ \forall t,\ \forall v \in \mathscr{C}_k,\ \forall z, \quad (31g)$$

$$(C_8)\ \ \sum\nolimits_{z=1}^{Z}\sum\nolimits_{v\in\mathscr{C}_k} \mathrm{I}_{v,z}(t) = Z,\ \forall t,\ \forall v \in \mathscr{C}_k,\ \forall z, \quad (31h)$$

$$(C_9)\ \ \sum\nolimits_{v\in\mathscr{C}_k} \mathrm{I}_v(t) \le Z,\ \forall t,\ \forall v \in \mathscr{C}_k, \quad (31i)$$

$$(C_{10})\ \ \{\mathrm{I}_v(t), \mathrm{I}_{v,z}(t)\} \in \{0,1\},\ \forall t,\ \forall v \in \mathscr{C}_k,\ \forall z, \quad (31j)$$

$$(C_{11})\ \ 0 \le \mathrm{I}_{v,z}(t)P_{v,z}(t) \le P_v^{\max},\ \forall t,\ \forall z,\ \forall v \in \mathscr{C}_k \quad (31k)$$

$$(C_{12})\ \ \sum\nolimits_{z=1}^{Z} \mathrm{I}_{v,z}(t)P_{v,z}(t) \le P_v^{\max},\ \forall t,\ \forall z,\ v \in \mathscr{C}_k, \quad (31l)$$

where constraint (31a) restricts the server to choose $|\mathscr{C}_k|$ CVs in VEFL global round k. The second constraint in C_2 restricts the total delay to be within the deadline budget. Constraint C_3 in (31c) ensures that the total computation and transmission time is within the expected worst-case sojourn period. Besides, constraint (31d) is taken to restrict the total energy consumption to be within the total energy budgets of the CVs. C_5 in (31e) ensures that the server's total expense cannot exceed its budget $\Xi(k)$. Furthermore, constraint C_6 is for the non-negative integer values of the local iteration numbers $l_v(k)$'s. Constraint C_7 in (31g) ensures that one pRB is allocated to only one CV and constraint C_8 in (31h) ensures all pRBs are allocated to all CVs. C_9 ensures that the total number of scheduled CVs in a slot does not exceed the total available pRBs. Constraint C_{10} is for the binary decision variables. Finally, constraints C_{11} and C_{12} ensures that the allocated power over pRB z and over all Z pRBs are less than the maximum possible transmission power of the CV, respectively.

Remark 3: The VEFL parameters, i.e., $\mathbf{1}(k)$, $\boldsymbol{l}(k)$ and $\boldsymbol{\eta}(k)$, and the RAT parameters, i.e., $\mathbf{I}(t)$, $\check{\mathbf{I}}(t)$ and $\mathbf{p}(t)$, are coupled in problem (31). One of the main challenges of this problem is that the CVs must know their $l_v(k)$'s and $\eta_v(k)$'s when the VEFL round k starts. However, to solve (31) for these values, the server must also know $\mathrm{t}_v^{\mathrm{tx}}(k)$'s and $\mathrm{e}_v^{\mathrm{tx}}(k)$'s, which is impossible at the beginning of the VEFL round because the Uu links change in each slot. Moreover, the server also needs to know the charging policies $\xi_v(l_v(k),\eta_v(k))$'s, which depend on the total energy consumption $\mathrm{e}_v^{\mathrm{tot}}(k)$'s. As such, in the following, we decompose this joint problem into a VEFL parameter optimization sub-problem and a RAT parameter optimization sub-problem. More specifically, the VEFL parameter optimization sub-problem uses the worst-case estimation of $\mathrm{t}_v^{\mathrm{tx}}(k)$'s and $\mathrm{e}_v^{\mathrm{tx}}(k)$'s at the beginning of each VEFL round k. On the other hand, the server solves the RAT parameter optimization sub-problem when it requests for the trained models of the CVs.

IV. Problem Transformations and Solutions

This section demonstrates our problem decomposition, followed by the solutions. First, we illustrate the problem decomposition steps for the two sub-problems discussed in Remark 3 through Fig. 6. As the original VEFL

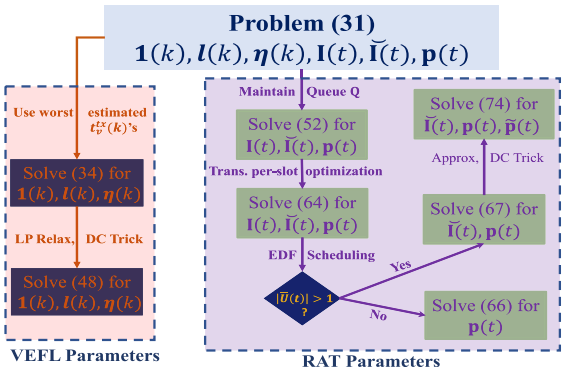

Fig. 6. Problem decomposition steps.

parameter optimization sub-problem (c.f. (34)) is challenging, we perform linear programming (LP) relaxation and transform it into a convex optimization problem (c.f. (48)) that we solve to obtain the optimized $\mathbf{1}(k)$, $\boldsymbol{l}(k)$ and $\boldsymbol{\eta}(k)$ values. Moreover, the server monitors the payload buffers of the CVs and optimizes the RAT parameters to maximize the probability of successful reception $p_v^{\mathrm{suc}}(k)$'s of the CVs' trained models (c.f. (52)). However, since the CSI changes in each slot, the $p_v^{\mathrm{suc}}(k)$'s can only be determined after completing the VEFL round. As such, we convert the problem into a per-slot utility maximization problem (c.f. (64)). To that end, the earliest deadline first (EDF) based scheduling [45] is adopted, followed by the resource allocation optimization. When more than one CV is scheduled, the RAT parameter optimization problem (c.f. (67)) is non-convex. Thus, we perform LP relaxation, use the DC techniques and transform it into a relaxed convex optimization problem (c.f. (74)) that we solve to optimize the pRB and power allocations.

The server considers the following upper-bounded transmission energy consumption to calculate its per round cost at the beginning of each VEFL round k.

$$\hat{\xi}_v(l_v(k),\eta_v(k)) = \left[\mathrm{e}_v^{\mathrm{cmp}}(k) + \kappa P_v^{\max}\cdot \bar{\mathrm{t}}_v^{\mathrm{tx}}\right]\phi_v + \bar{\phi}_v, \quad (32)$$

where $\bar{\mathrm{t}}_v^{\mathrm{tx}}$ is the worst case total number of required TTIs, which is calculated in (33).

$$\bar{\mathrm{t}}_v^{\mathrm{tx}} = \left\lceil S(\boldsymbol{\omega},\mathrm{FPP})/(\kappa(1-\upsilon)\omega\cdot\mathbb{E}\left[\log_2(1+\bar{\Gamma}_v)\right]\times\tilde{z}(k))\right\rceil, \quad (33)$$

where $\lceil\cdot\rceil$ is the *ceil* operator and $\bar{\Gamma}_v = \min\{\{\Gamma_{v,z}\}_{z=1}^{Z}\}$ is the worst-case SNR over the worst channel.[5] Besides, $\tilde{z}(k) = Z/V_{\check{t}_k}$ if $Z < V_{\check{t}_k}$ and $\tilde{z}(k) = 1$ otherwise.

A. PDPC: VEFL Parameter Optimizations

Given the upper-bounded communication delay and energy consumption, the server wishes to jointly optimize $\mathbf{1}(k)$, $\boldsymbol{l}(k)$ and $\boldsymbol{\eta}(k)$. Intuitively, the trained model $\boldsymbol{\omega}_{v,k}$ is expected to deliver better performance if CV v trains it for more local iterations. Besides, more training samples usually help enhance a trained model's test performance. However, since the CVs are not stationary, short sojourn periods can also play

[5]In practical 3GPP networks, one may use the historical CSI information to get an estimated worst-case channel condition.

a critical role. As such, we want to configure $\mathbf{1}(k)$, $\boldsymbol{l}(k)$ and $\boldsymbol{\eta}(k)$ under the joint consideration of the sojourn periods and the dataset sizes. Considering the worst-case communication delay, energy consumption and cost, we optimize the weighted sum of CVs' local training iterations, which is given as

$$\underset{\mathbf{1}(k),\boldsymbol{l}(k),\boldsymbol{\eta}(k)}{\text{maximize}} \ \sum\nolimits_{v=1}^{V_{\check{t}_k}} \mathbf{1}(v \in \mathscr{C}_k) l_v(k)\ \Theta_v, \tag{34}$$

$$\text{s.t. } C_1, C_6, \tag{34a}$$

$$(\tilde{C}_2) \quad \mathbf{1}(v \in \mathscr{C}_k)\mathrm{t}_v^{\mathrm{cmp}}(k) + \kappa \mathbf{1}(v \in \mathscr{C}_k)\bar{\mathrm{t}}_v^{\mathrm{tx}} \leq \mathbf{1}(v \in \mathscr{C}_k)\min\{\mathrm{t}^{\mathrm{th}}(k), \mathrm{t}_v^{\mathrm{soj}}(\check{t}_k)\} \quad \forall v, k, \tag{34b}$$

$$(\tilde{C}_3) \quad \mathbf{1}(v \in \mathscr{C}_k)\mathrm{e}_v^{\mathrm{cmp}}(k) + \kappa \mathbf{1}(v \in \mathscr{C}_k)\bar{\mathrm{t}}_v^{\mathrm{tx}} P_v^{\max} \leq \mathbf{1}(v \in \mathscr{C}_k) \cdot \mathrm{e}_v^{\mathrm{bud}}(k), \ \forall v, k, \tag{34c}$$

$$(\tilde{C}_5) \quad \sum\nolimits_{v=1}^{V_{\check{t}_k}} \mathbf{1}(v \in \mathscr{C}_k)\hat{\xi}(l_v(k), \eta_v(k)) \leq \Xi(k), \ \forall k, \tag{34d}$$

where the constraints are taken for the same reasons as in (31) and $\Theta_v = \theta_v / \sum_{v=1}^{V_{\check{t}_k}} \theta_v$, where θ_v is calculated in (35).

$$\theta_v = (1 - \bar{\lambda})[\mathrm{D}_v / \sum\nolimits_{v=1}^{V_{\check{t}_k}} \mathrm{D}_v] + \bar{\lambda}[\mathrm{t}_v^{\mathrm{soj}}(\check{t}_k) / \sum\nolimits_{v=1}^{V_{\check{t}_k}} \mathrm{t}_v^{\mathrm{soj}}(\check{t}_k)], \tag{35}$$

where $\bar{\lambda} \in [0, 1]$ is a weighting parameter. A higher value of $\bar{\lambda}$ puts more weight on the estimated sojourn period, while a smaller value puts more weight on the dataset size. Note that if the server solves (34), it is expected to have $p_v^{\mathrm{suc}}(k) = 1$ since we consider the worst-case estimate for the transmission delay, energy consumption and cost. In other words, the sub-problem (34) retains the original problem's constraints and is expected to maximize the original objective function in each VEFL round.

Problem (34) has binary, integer and multiplicative decision variables and is NP-hard. Moreover, the CPU frequency and local iteration numbers are inexorably related, affecting the CV selection process. To tackle these grand challenges, we do standard LP relaxation on $l_v(k)$, which gives the following modified problem.

$$\underset{\mathbf{1}(k),\boldsymbol{l}(k),\boldsymbol{\eta}(k)}{\text{maximize}} \ \sum\nolimits_{v=1}^{V_{\check{t}_k}} \mathbf{1}(v \in \mathscr{C}_k) l_v(k)\ \Theta_v, \tag{36}$$

$$\text{s.t. } C_1, \tilde{C}_2, \tilde{C}_3, \tilde{C}_5, \tag{36a}$$

$$(\tilde{C}_6) \quad \mathbf{1}(v \in \mathscr{C}_k) \cdot l_v(k) \ \geq l_k^{\mathrm{des}}, \quad \forall v, k, \tag{36b}$$

To that end, we now handle the multiplicative $\mathbf{1}(v \in \mathscr{C}_k)$ and $l_v(k)$ variables. To tackle this non-linearity, let us define the following new variable:

$$\tilde{l}_v(k) \coloneqq l_v(k)\ \mathbf{1}(v \in \mathscr{C}_k). \tag{37}$$

Since $\mathbf{1}(v \in \mathscr{C}_k) \in \{0, 1\}$ and $l_v^{\min} \leq l_v(k) \leq l_v^{\max}$, we can equivalently write this new variable with the following inequalities:

$$l_v^{\min}\mathbf{1}(v \in \mathscr{C}_k) \leq \tilde{l}_v(k) \leq l_v^{\max}\mathbf{1}(v \in \mathscr{C}_k), \tag{38a}$$

$$l_v^{\min}(1 - \mathbf{1}(v \in \mathscr{C}_k)) \leq l_v(k) - \tilde{l}_v(k) \leq l_v^{\max}(1 - \mathbf{1}(v \in \mathscr{C}_k)), \tag{38b}$$

$$0 \leq \tilde{l}_v(k) \leq l_v^{\max}. \tag{38c}$$

Moreover, using the DC trick, the binary CV selection decision variables can equivalently be represented as follows:

$$\sum\nolimits_{v=1}^{V_{\check{t}_k}} \mathbf{1}(v \in \mathscr{C}_k) - \sum\nolimits_{v=1}^{V_{\check{t}_k}} (\mathbf{1}(v \in \mathscr{C}_k))^2 \leq 0, \tag{39a}$$

$$0 \leq \mathbf{1}(v \in \mathscr{C}_k) \leq 1, \quad \forall v \in \mathscr{V}_{\check{t}_k} \tag{39b}$$

Equation (39a) is non-convex. However, since it is the difference between two convex functions, we can equivalently write the optimization problem as follows:

$$\underset{\mathbf{1}(k),\tilde{\boldsymbol{l}}_k,\boldsymbol{l}(k),\boldsymbol{\eta}(k)}{\text{minimize}} \ -\sum_{v=1}^{V_{\check{t}_k}} \tilde{l}_v(k)\Theta_v + \bar{\vartheta}\Big[\sum_{v=1}^{V_{\check{t}_k}} \mathbf{1}(v \in \mathscr{C}_k) - \sum_{v=1}^{V_{\check{t}_k}} (\mathbf{1}(v \in \mathscr{C}_k))^2\Big], \tag{40}$$

$$\text{s.t. (38a), (38b)(38c), (39b)}, \tag{40a}$$

$$(\tilde{C}_1) \quad \sum\nolimits_{v=1}^{V_{\check{t}_k}} \mathbf{1}(v \in \mathscr{C}_k) \leq |\mathscr{C}_k|, \quad \forall k, \tag{40b}$$

$$(\tilde{C}_2) \quad (\tilde{l}_v(k)c_v\mathrm{D}_v)/\eta_v(k) + \mathbf{1}(v \in \mathscr{C}_k)\kappa\bar{\mathrm{t}}_v^{\mathrm{tx}} \leq \mathbf{1}(v \in \mathscr{C}_k)\min\{\mathrm{t}^{\mathrm{th}}(k), \mathrm{t}_v^{\mathrm{soj}}(\check{t}_k)\} \quad \forall v, k, \tag{40c}$$

$$(\tilde{C}_3) \quad \tilde{l}_v(k)(\zeta/2)c_v\mathrm{D}_v\eta_v(k)^2 + \mathbf{1}(v \in \mathscr{C}_k)\kappa\bar{\mathrm{t}}_v^{\mathrm{tx}}P_v^{\max} \leq \mathbf{1}(v \in \mathscr{C}_k)\mathrm{e}_v^{\mathrm{bud}}(k), \ \forall v, k, \tag{40d}$$

$$(\tilde{C}_5) \quad \sum\nolimits_{v=1}^{V_{\check{t}_k}} \big([\tilde{l}_v(k)(\zeta/2)c_v\mathrm{D}_v\eta_v(k)^2 + \mathbf{1}(v \in \mathscr{C}_k) \times \kappa P_v^{\max}\bar{\mathrm{t}}_v^{\mathrm{tx}}]\phi_v + \mathbf{1}(v \in \mathscr{C}_k)\bar{\phi}_v\big) \leq \Xi(k), \ \forall k, \tag{40e}$$

$$(\tilde{C}_6) \quad \tilde{l}_v(k) \geq l_k^{\mathrm{des}}, \quad \forall v, k, \tag{40f}$$

where $\bar{\vartheta} \gg 0$ is a penalty function that forces $\mathbf{1}(v \in \mathscr{C}_k)$ to be either 0 or 1.

Note that (40) is a DC problem. We use successive convex approximation (SCA) to approximate the second quadratic term inside the penalty function as follows:

$$\begin{aligned}\sum\nolimits_{v=1}^{V_{\check{t}_k}} (\mathbf{1}(v \in \mathscr{C}_k))^2 &\geq \sum\nolimits_{v=1}^{V_{\check{t}_k}} (\mathbf{1}(v \in \mathscr{C}_k, i))^2 \\ &+ \sum\nolimits_{v=1}^{V_{\check{t}_k}} 2 \cdot \mathbf{1}(v \in \mathscr{C}_k, i)[\mathbf{1}(v \in \mathscr{C}_k) - \mathbf{1}(v \in \mathscr{C}_k, i)] \\ &= H(\mathbf{1}(v \in \mathscr{C}_k, i)),\end{aligned} \tag{41}$$

where $\mathbf{1}(v \in \mathscr{C}_k, i)$ is an initial feasible point. Moreover, we can linearize the first term in $\tilde{C}_2$ as follows:

$$\begin{aligned}\frac{\tilde{l}_v(k)c_v\mathrm{D}_v}{\eta_v(k)} &\approx \frac{\tilde{l}_v(k,i)c_v\mathrm{D}_v}{\eta_v(k,i)} + \frac{c_v\mathrm{D}_v}{\eta_v(k,i)}[\tilde{l}_v(k) - \tilde{l}_v(k,i)] \\ &+ \frac{\tilde{l}_v(k,i)c_v\mathrm{D}_v}{-(\eta_v(k,i))^2}[\eta_v(k) - \eta_v(k,i)] \\ &= A_1(\tilde{l}_v(k), \eta_v(k)),\end{aligned} \tag{42}$$

where $\tilde{l}_v(k,i)$ and $\eta_v(k,i)$ are two initial feasible points. Similarly, we can linearize the multiplicative term in $\tilde{C}_3$ and $\tilde{C}_5$ as follows:

$$\begin{aligned}&\tilde{l}_v(k)(\zeta/2)c_v\mathrm{D}_v\eta_v(k)^2 \\ &\approx \tilde{l}_v(k,i)(\zeta/2)c_v\mathrm{D}_v\eta_v(k,i)^2 \\ &\quad + (\zeta/2)c_v\mathrm{D}_v\eta_v(k,i)^2[\tilde{l}_v(k) - \tilde{l}_v(k,i)] \\ &\quad + \tilde{l}_v(k,i)\zeta c_v\mathrm{D}_v\eta_v(k,i)[\eta_v(k) - \eta_v(k,i)] \\ &= A_2(\tilde{l}_v(k), \eta_v(k)).\end{aligned} \tag{43}$$

Moreover, (40f) may not always be possible due to practical limitations. As such, we find the upper-bounded $\tilde{l}_v(k)$ based

on the time constraints as follows:

$$\begin{aligned}\tilde{l}_v(k) \leq \big[&\mathbf{1}(v\in\mathscr{C}_k,i)\min\{\mathrm{t}^{\mathrm{th}}(k),\mathrm{t}_v^{\mathrm{soj}}(\breve{t}_k)\}\\&+[(\tilde{l}_v(k,i)c_v\mathrm{D}_v)/(-\eta_v(k,i)^2)][\eta_v(k)-\eta_v(k,i)]\\&-\mathbf{1}(v\in\mathscr{C}_k,i)\kappa\bar{\mathrm{t}}_v^{\mathrm{tx}}\big]\times[\eta_v(k,i)/(c_v\mathrm{D}_v)]\\&=B_1(\tilde{l}_v(k),\eta_v(k)).\end{aligned}\tag{44}$$

Furthermore, CV's energy constraint provides the following upper bound.

$$\begin{aligned}\tilde{l}_v(k)\leq\big[&\mathbf{1}(v\in\mathscr{C}_k,i)\mathrm{e}_v^{\mathrm{bud}}(k)\\&+(\zeta/2)c_v\mathrm{D}_v\eta_v(k,i)^2[\tilde{l}_v(k)-\tilde{l}_v(k,i)]\\&-\mathbf{1}(v\in\mathscr{C}_k,i)\kappa\bar{\mathrm{t}}_v^{\mathrm{tx}}P_v^{\max}\\&+\tilde{l}_v(k,i)(\zeta/2)c_v\mathrm{D}_v\eta_v(k,i)^2\big]/[\tilde{l}_v(k,i)\zeta c_v\mathrm{D}_v\eta_v(k,i)]\\&=B_2(\tilde{l}_v(k),\eta_v(k)).\end{aligned}\tag{45}$$

Therefore, combining both time and energy constraints, the upper-bounded $\tilde{l}_v(k)$ is obtained as follows:

$$\tilde{l}_v(k)\leq\min\{B_1(\tilde{l}_v(k),\eta_v(k)),B_2(\tilde{l}_v(k),\eta_v(k))\}.\tag{46}$$

Similarly, we can write the lower bound as follows:

$$\tilde{l}_v(k)\geq\min\Big\{l_k^{\mathrm{des}},\min\{B_1(\tilde{l}_v(k),\eta_v(k)),B_2(\tilde{l}_v(k),\eta_v(k))\}\Big\}.\tag{47}$$

To that end, we optimize the upper bound of problem (40) by successively optimizing the following problem:

$$\begin{aligned}\underset{\mathbf{1}(k),\tilde{\boldsymbol{l}}_k,\boldsymbol{l}(k),\boldsymbol{\eta}(k)}{\text{minimize}}\quad&-\sum_{v=1}^{V_{\breve{t}_k}}\tilde{l}_v(k)\Theta_v+\bar{\vartheta}\Big[\sum_{v=1}^{V_{\breve{t}_k}}\mathbf{1}(v\in\mathscr{C}_k)\\&-H(\mathbf{1}(v\in\mathscr{C}_k,i))\Big],\end{aligned}\tag{48}$$

$$\text{s.t. (38a), (38b), (38c), (39b), (46), (47)}\tag{48a}$$

$$(\tilde{C}_1)\quad\sum\nolimits_{v=1}^{V_{\breve{t}_k}}\mathbf{1}(v\in\mathscr{C}_k,i)\leq|\mathscr{C}_k|,\quad\forall k,\tag{48b}$$

$$\begin{aligned}(\tilde{C}_5)\quad\sum\nolimits_{v=1}^{V_{\breve{t}_k}}\big([A_2(\tilde{l}_v(k),\eta_v(k))+\kappa\mathbf{1}(v\in\mathscr{C}_k,i)P_v^{\max}\bar{\mathrm{t}}_v^{\mathrm{tx}}]\phi_v\\+\mathbf{1}(v\in\mathscr{C}_k,i)\bar{\phi}_v\big)\leq\Xi(k),\quad\forall k,\end{aligned}\tag{48c}$$

where $H(\mathbf{1}(v\in\mathscr{C}_k,i))$ is the global underestimation of $\sum_{v=1}^{V_{\breve{t}_k}}(\mathbf{1}(v\in\mathscr{C}_k))^2$ and is calculated in (41). Moreover, $A_2(\tilde{l}_v(k),\eta_v(k))$ is calculated in (43). The server solves (48) to jointly optimize the VEFL parameters and conveys the $l_v^*(k)$'s and $\eta_v^*(k)$'s to the selected CVs as part of the SLAs. Note that problem (48) is convex and can be solved efficiently using existing solvers such as CVX [46]. We use Algorithm 2 to solve (48) iteratively. Since (48) has $4V_{\breve{t}_k}$ decision variables, $2+6V_{\breve{t}_k}$ constraints, and Algorithm 2 runs for a maximum of I iterations, the computation time complexity of our proposed solution is $\mathscr{O}\big(64IV_{\breve{t}_k}^3(2+6V_{\breve{t}_k})\big)$ [47].

Remark 4: Note that for FDPC, as all CVs are to be selected, there are no binary $\mathbf{1}(v\in\mathscr{C}_k)$ variables. Apart from the DC trick, we follow a similar approach as in (34) to jointly optimize $\boldsymbol{l}(k)$ and $\boldsymbol{\eta}(k)$ by maximizing the weighted sum of the CVs' local iterations, where we put the weight based on the aggregation weight $p_v=\bar{p}_v/\sum_{v=1}^{V_{\breve{t}_k}}\bar{p}_v$. We omitted the optimization problem for brevity. Moreover, the following RAT parameter optimization process is also used for FDPC.

Algorithm 2 Iterative Joint CV Selection and Local Iteration Selection Process

Input: Initial feasible set $\mathbf{1}(v\in\mathscr{C}_k,i)$, $\tilde{\boldsymbol{l}}(k,i)$, $\boldsymbol{\eta}(k,i)$, $i=0$, maximum iteration I, precision level ϵ^{prec}, initial penalty ϑ_0

Repeat:

  $i\leftarrow i+1$; $\bar{\vartheta}\leftarrow\vartheta_0+i$

  Solve (48) using $\bar{\vartheta}$, $\mathbf{1}(v\in\mathscr{C}_k,i-1)$, $\tilde{\boldsymbol{l}}(k,i-1)$ and $\boldsymbol{\eta}(k,i-1)$ to find $\mathbf{1}(v\in\mathscr{C}_k,i)$, $\tilde{\boldsymbol{l}}(k,i)$ and $\boldsymbol{\eta}(k,i)$

Until converge with ϵ^{prec} precision or $i=I$

Output: CV selection set $\mathbf{1}(k)$, local iteration set $\boldsymbol{l}(k)$ and CPU frequencies $\boldsymbol{\eta}(k)$

B. RAT Parameter Optimization

The server initiates CVs' trained models reception from slot $\tau(k)=\min\{\tau_1(k),\ldots,\tau_{|\mathscr{C}_k|}(k)\}$, where $\tau_v(k)$ is calculated in (49), to minimize its operating cost.

$$\tau_v(k)=\begin{cases}\breve{t}_k+\lfloor\mathrm{t}_v^{\mathrm{soj}}(\breve{t}_k)/\kappa\rfloor-\bar{\mathrm{t}}_v^{\mathrm{tx}},&\text{if } \mathrm{t}_v^{\mathrm{soj}}(\breve{t}_k)<\mathrm{t}^{\mathrm{th}}(k),\\\breve{t}_{k+1}-\bar{\mathrm{t}}_v^{\mathrm{tx}},&\text{otherwise.}\end{cases}\tag{49}$$

Intuitively, the RAT resources are required only when there are uplink payloads. As such, (49) calculates the slot at which a CV finishes its local model training, and $\tau(k)$ ensures that the server does not start the model receptions if no CV finishes its local model training.

Upon solving (48), the server broadcasts the global model $\boldsymbol{\omega}_k$, $\boldsymbol{l}(k)$, $\boldsymbol{\eta}(k)$ and the transmission-reception starting slot $\tau(k)$. Then, after performing $l_v(k)$ local iterations, CV v needs to offload its trained model $\boldsymbol{\omega}_{v,k+1}$ to the server. From slot $\tau(k)$ and onward until the $\breve{t}_{k+1}$, the server needs to perform CV scheduling and resource allocations to successfully receive the payloads within each CV's specific deadline constraint. Denote these optimization slots for the server by the set $\mathscr{T}_k=\{t\}_{t=\tau(k)}^{\breve{t}_{k+1}}$.

Note that all CVs have the same payload of $S(\boldsymbol{\omega},\mathrm{FPP})$ bits to offload to the server since they receive the same model. The server maintains a virtual remaining-local-model-payload buffer of the CVs $\mathbf{Q}(t)\triangleq[Q_1(t),\ldots,Q_{|\mathscr{C}_k|}(t)]$, where CV v's remaining payload buffer $Q_v(t)$ evolves as follows:

$$Q_v(t+1)=[Q_v(t)-\kappa\cdot r_v(t)]^+,\quad\forall\, t,\tag{50}$$

where $[\cdot]^+$ means $\max\{\cdot,0\}$ and $r_v(t)$ is the uplink data rate calculated in (11). Considering the above factors, we can calculate the probability of successful reception of the CV's trained model $\boldsymbol{\omega}_{v,k+1}$ as follows:

$$p_v^{\mathrm{suc}}(k)\;=1-[Q_v(\breve{t}_{k+1})/S(\boldsymbol{\omega},\mathrm{FPP})].\tag{51}$$

We stress that this $p_v^{\mathrm{suc}}(k)$ depends on the CSI, RAT parameters and the quality of the worst-case required number of slots $\bar{\mathrm{t}}_v^{\mathrm{tx}}$, calculated in (33).

Given that the VEFL parameters are solved using (48), we aim to solve the following sub-problem for jointly optimizing the RAT parameters.

$$\underset{\mathbf{x}(t)}{\text{maximize}}\quad\sum\nolimits_{v=1}^{|\mathscr{C}_k|}p_v^{\mathrm{suc}}(k),\tag{52}$$

$$\text{s.t. } C_7 - C_{12}, \tag{52a}$$

$$\mathrm{t}_v^{\mathrm{tx}}(k) \le \kappa \cdot \min\left\{\breve{t}_k + \left\lfloor \mathrm{t}_v^{\mathrm{soj}}(\breve{t}_k)/\kappa \right\rfloor - \tau_v(k), \breve{t}_{k+1} - \tau_v(k)\right\}, \tag{52b}$$

where $\mathbf{x}(t) = [\mathbf{I}(t), \breve{\mathbf{I}}(t), \mathbf{p}(t)]$. Constraints in (52a) are taken for the same reasons as in the original problem in (31). Besides, constraint (52b) ensures that the entire trained model $\boldsymbol{\omega}_{v,k+1}$ has to be received within the remaining $\min\{\breve{t}_k + \lfloor \mathrm{t}_v^{\mathrm{soj}}(\breve{t}_k)/\kappa \rfloor - \tau_v(k), \breve{t}_{k+1} - \tau_v(k)\}$ slots.

Remark 5: The maximization of $\sum_{v=1}^{|\mathscr{C}_k|} p_v^{\mathrm{suc}}(k)$ in (52) is not straightforward since the CSI varies in each slot t, and the CVs require multiple transmission slots to offload their trained models. Besides, while $p_v^{\mathrm{suc}}(k)$ is calculated at the end of the current VEFL round k, based on (51), the RAT parameters must be optimized in each slot t. Note that the radio resources are available for the entire VEFL round. Intuitively, the server should seek a cost-effective way to perform the VEFL training and trained model reception to optimize the associated monetary cost, as defined in (30). As such, we aim to devise a per-slot-based long-term energy efficiency (EE) maximization problem to maximize the probability of successful model reception implicitly. In other words, to maximize $p_v^{\mathrm{suc}}(k)$, the server aims to maximize the transmission EE, which shall also ensure optimized cost $\xi_v(l_v(k), \eta_v(k))$.

Without any loss of generality, we define EE as follows:

$$\beta(t) = \Big(\sum_{v=1}^{|\mathscr{C}_k|} r_v(t)\Big) / \Big(\sum_{v=1}^{|\mathscr{C}_k|} \sum_{z=1}^{Z} P_{v,z}(t)\Big). \tag{53}$$

Note that (53) is the standard EE of a wireless network that calculates the tradeoff between the total sum rate and the total energy expense [48]. The expected average EE during round k is calculated as

$$\bar{\beta}(k) = \bar{\mathrm{r}}(k) / \bar{\mathrm{P}}(k), \tag{54}$$

where $\bar{\mathrm{r}}(k) = \left[1/(\breve{t}_{k+1} - \tau(k))\right] \sum_{t=\tau(k)}^{\breve{t}_{k+1}} \mathbb{E}\left[\sum_{v=1}^{|\mathscr{C}_k|} r_v(t)\right]$ and $\bar{\mathrm{P}}(k) = \left[1/(\breve{t}_{k+1} - \tau(k))\right] \sum_{t=\tau(k)}^{\breve{t}_{k+1}} \mathbb{E}\left[\sum_{v=1}^{|\mathscr{C}_k|} \sum_{z=1}^{Z} P_{v,z}(t)\right]$. We jointly optimize the RAT parameters to maximize the EE as

$$\underset{\mathbf{x}(t)}{\text{maximize}} \quad \bar{\beta}(k), \tag{55}$$

$$\text{s.t. } C_7 - C_{12}, \tag{55a}$$

$$\mathrm{t}_v^{\mathrm{tx}}(k) \le \kappa \cdot \min\left\{\breve{t}_k + \left\lfloor \mathrm{t}_v^{\mathrm{soj}}(\breve{t}_k)/\kappa \right\rfloor - \tau_v(k), \breve{t}_{k+1} - \tau_v(k)\right\}, \tag{55b}$$

where the constraints are taken for the same reasons as in (52).

Remark 6: By maximizing the EE, problem (55) ensures that the monetary cost for the uplink transmission is optimized, while constraint (55b) ensures that all local $\boldsymbol{\omega}_{v,k+1}$'s are fully received. Therefore, if problem (55) is solved, the server should ensure the maximization of $\sum_{v=1}^{|\mathscr{C}_k|} p_v^{\mathrm{suc}}(k)$ in a cost effective way, which provides the expected loss minimization of the VEFL.

The objective function in (55) is fractional, non-convex and challenging to solve. In practice, the Dinkelbach method [49] is widely used to transform the fractional objective function equivalently into a linear one. Denote the optimal solution for (55) by $\mathbf{x}^*(t) = [\mathbf{I}^*(t), \breve{\mathbf{I}}^*(t), \mathbf{p}^*(t)]$ and the corresponding optimal long-term EE by $\bar{\beta}(k)^* = \bar{\mathrm{r}}(k)^* / \bar{\mathrm{P}}(k)^*$. Then, using the Dinkelbach approach, we transform the fractional objective function to the following equivalent form [50]:

$$\bar{\beta}(k) = \bar{\mathrm{r}}(k) / \bar{\mathrm{P}}(k) = \bar{\mathrm{r}}(k) - \bar{\beta}(k)^* \bar{\mathrm{P}}(k). \tag{56}$$

However, since $\bar{\beta}(k)^*$ is unknown beforehand, available historical information is usually utilized. Denote $\bar{\beta}(k,t)$ as

$$\bar{\beta}(k,t) := \Big[\sum_{\hat{t}=\tau(k)}^{t-1} \sum_{v=1}^{|\mathscr{C}_k|} r_v(\hat{t})\Big] / \Big[\sum_{\hat{t}=\tau(k)}^{t-1} \sum_{v=1}^{|\mathscr{C}_k|} \sum_{z=1}^{Z} P_{v,z}(\hat{t})\Big], \tag{57}$$

where $\bar{\beta}(k, \tau(k)) = 0$.

To that end, plugging (56) and (57) into the objective function of (55), we get the modified objective $\bar{\mathrm{r}}(k) - \bar{\beta}(k,t)\bar{\mathrm{P}}(k)$. Then, we can rewrite problem (55) as follows:

$$\underset{\mathbf{x}(t)}{\text{minimize}} \quad -\bar{\mathrm{r}}(k) + \bar{\beta}(k,t)\bar{\mathrm{P}}(k), \tag{58}$$

$$\text{s.t. } C_7 - C_{12}, (55\text{b}), \tag{58a}$$

Note that problems (55) and (58) are equivalent [50], and similar treatment is widely used in the literature [48], [50], [51], [52].

Clearly, problem (58) is stochastic due to the intermittent Uu links and is challenging to solve. As such, we leverage the Lyapunov optimization framework to transform (55) into an online and per-slot-wise deterministic optimization problem. Let us define a quadratic Lyapunov function as

$$L(\mathbf{Q}(t)) := (1/2) \sum_{v=1}^{|\mathscr{C}_k|} Q_v(t)^2. \tag{59}$$

Then, we can write the conditional Lyapunov drift as

$$\Delta(\mathbf{Q}(t)) = \mathbb{E}\left[L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t)) | \mathbf{Q}(t)\right]. \tag{60}$$

To that end, we express a Lyapunov drift-plus-penalty function as follows:

$$\Delta_C(\mathbf{Q}(t)) = \Delta(\mathbf{Q}(t)) + C \cdot \mathbb{E}\Big[-\sum_{v=1}^{|\mathscr{C}_k|} r_v(t) + \bar{\beta}(k,t) \sum_{v=1}^{|\mathscr{C}_k|} \sum_{z=1}^{Z} P_{v,z}(t) | \mathbf{Q}(t)\Big], \tag{61}$$

where $C \in [0, +\infty]$ is a control parameter that adjusts the trade-off between EE and payload buffer backlogs.

Remark 7: We want to minimize the upper-bounded Lyapunov drift-plus-penalty function in (61). The right-hand side suggests minimizing the drift, i.e., payload buffer size, while the added penalty controls the EE of the system.

Lemma 1: For an arbitrary $\mathbf{x}(t) = [\mathbf{I}(t), \breve{\mathbf{I}}(t), \mathbf{p}(t)]$, the Lyapunov drift-plus-penalty function is upper-bounded as

$$\Delta_C(\mathbf{Q}(t)) \le \varpi - \mathbb{E}\left[\sum_{v=1}^{|\mathscr{C}_k|} \kappa r_v(t) Q_v(t) | \mathbf{Q}(t)\right] + C \cdot \mathbb{E}\Big[-\sum_{v=1}^{|\mathscr{C}_k|} r_v(t) + \bar{\beta}(k,t) \sum_{v=1}^{|\mathscr{C}_k|} \sum_{z=1}^{Z} P_{v,z}(t) | \mathbf{Q}(t)\Big], \tag{62}$$

where $\varpi \ge 0$ does not depend on the queue state.

Proof: The proof is left in Appendix B. ∎

From Lemma 1, we find the following per-slot utility function that we want to minimize:

$$\mathrm{U}(t)=C\bar{\beta}(k,t)\sum_{v=1}^{|\mathscr{C}_k|}\sum_{z=1}^{Z}P_{v,z}(t) - \sum_{v=1}^{|\mathscr{C}_k|} r_v(t)\big[\kappa Q_v(t)+C\big]. \quad (63)$$

Using (10) and (11), we can write

$$\mathrm{U}(t)=C\bar{\beta}(k,t)\sum_{v=1}^{|\mathscr{C}_k|}\sum_{z=1}^{Z}P_{v,z}(t)-\omega(1-\upsilon)\sum_{v=1}^{|\mathscr{C}_k|}\mathrm{I}_v(t) \times\Big[\sum_{z=1}^{Z}\log_2\big(1+(\mathrm{I}_{v,z}(t)P_{v,z}(t)\|\mathbf{h}_{v,z}(t)\|^2)/(\omega\varsigma^2)\big)\Big](\kappa Q_v(t)+C).$$

As such, we can pose the following per-slot online optimization problem that can be solved opportunistically.

$$\underset{\mathbf{x}(t)}{\text{minimize}}\ \mathrm{U}(t), \quad (64)$$

$$\text{s.t. } C_7-C_{12},(55b), \quad (64a)$$

where the constraints are taken for the same reasons as in (58).

Recall that CV v starts offloading its trained model from slot $\tau_v(k)$. Therefore, in slot $t\in\mathscr{T}_k$, denote the CVs that need to offload their trained models by the set $\mathscr{U}(t)$. Since the gNB can schedule at most Z CVs in a slot for their uplink transmissions and all $\boldsymbol{\omega}_{v,k+1}$'s need to be received within slot $\breve{t}_{k+1}$ to meet the deadline threshold, the gNB has to perform a scheduling decision when $|\mathscr{U}(t)|>Z$. As such, to ensure (55b), the medium access control (MAC) scheduler schedules the CVs based on the remaining deadlines and payload buffer status $\mathbf{Q}_v(t-1)$. Particularly, we adopt the widely used real-time operating system's EDF-based scheduling[6] [45]. Note that if EDF cannot guarantee zero deadline violation, no other algorithm can [53]. Algorithm 3 provides the scheduled CV set $\bar{\mathscr{U}}(t)$ during slot t.

Per this scheduling policy, a CV is scheduled in at least $\big\lfloor (Z/|\mathscr{C}_k|)\times\min\big\{\breve{t}_k+\big\lfloor \mathrm{t}_v^{\mathrm{soj}}(\breve{t}_k)/\kappa\big\rfloor,\breve{t}_{k+1}\big\}-\tau_v(k)\ \big\rfloor$ slots. As such, to satisfy constraint (55b), we enforce

$$r_v(t)\geq \begin{cases} \frac{Q_v(t-1)}{\kappa\times\min\left\{\breve{t}_k+\left\lfloor \mathrm{t}_v^{\mathrm{soj}}(\breve{t}_k)/\kappa\right\rfloor-\tau_v(k),\breve{t}_{k+1}-\tau_v(k)\right\}}, & \text{if } |\mathscr{C}_k|\leq Z,\\ \frac{Q_v(t-1)}{\kappa\times\left\lfloor (Z/|\mathscr{C}_k|)\times\min\left\{\breve{t}_k+\left\lfloor \mathrm{t}_v^{\mathrm{soj}}(\breve{t}_k)/\kappa\right\rfloor,\breve{t}_{k+1}\right\}-\tau_v(k)\ \right\rfloor}, & \text{else,}\end{cases} \quad (65)$$

where $t=\tau_v(k),\ldots,\min\{\breve{t}_k+\big\lfloor \mathrm{t}_v^{\mathrm{soj}}(\breve{t}_k)/\kappa\big\rfloor,\breve{t}_{k+1}\}$.

To that end, note that in a particular slot t, if only 1 CV requires offloading its trained model, it shall have all pRBs, i.e., $\mathrm{I}_{v,z}(t)=1,\forall z\in\mathscr{Z}$. The server only needs to optimize the transmission power of the scheduled CV. As such, in this special case, we pose the following optimization problem:

$$\underset{\mathbf{p}_v(t)}{\text{minimize}}\ \mathrm{U}(t), \quad (66)$$

$$\text{s.t. } C_{11},C_{12},(65),\mathrm{I}_{v,z}(t)=1,\quad \forall z. \quad (66a)$$

However, when $|\mathscr{U}(t)|>1$, the MAC scheduler schedules the CVs in the set $\bar{\mathscr{U}}(t)$ based on Algorithm 3. We, therefore, reformulate optimization problem (64) as

$$\underset{\bar{\mathbf{x}}(t)}{\text{minimize}}\ \bar{\mathrm{U}}(t), \quad (67)$$

$$\text{s.t. } C6,C7,C10,C11,(65),\mathrm{I}_{v,z}(t)\in\{0,1\},\quad \forall v\in\bar{\mathscr{U}}(t),z, \quad (67a)$$

[6]Similar scheduling is also widely used in deadline-constrained applications [32], [53], [54].

Algorithm 3 CV Scheduling in Each TTI

```
Determine the CV set 𝒰(t) eligible for scheduling
Set empty set 𝒰̄(t)
if |𝒰(t)| ≤ Z then
    𝒰̄(t) ← 𝒰(t)
    Set I_v(t) = 1, ∀v ∈ 𝒰(t)
else
    𝒰̄(t) ← indmin_[Z]{T_1^thr, ..., T_|𝒰(t)|^thr}, where
        T_v^thr = min{ ť_k + ⌊t_v^soj(ť_k)/κ⌋ − t, ť_{k+1} − t }
        /* indmin_[Z]{𝒬} means the index of
        the first Z smallest entry of 𝒬
        */
    Set I_v(t) = 1 ∀v ∈ 𝒰̄(t)
end
Output: Scheduled CV set 𝒰̄(t)
```

where $\bar{\mathbf{x}}(t)=[\breve{\mathbf{I}}(t),\mathbf{p}(t)]$ is the decision variables set for all $v\in\bar{\mathscr{U}}(t)$ and $\bar{\mathrm{U}}(t)$ is calculated as

$$\bar{\mathrm{U}}(t)=C\bar{\beta}(k,t)\sum_{v\in\bar{\mathscr{U}}(t)}\sum_{z=1}^{Z}P_{v,z}(t)-\omega(1-\upsilon) \times\sum_{v\in\bar{\mathscr{U}}(t)}\Big[\sum_{z=1}^{Z}\log_2(1+(\mathrm{I}_{v,z}(t)P_{v,z}(t)\|\mathbf{h}_{v,z}(t)\|^2)/(\omega\varsigma^2))\Big] \times[C+\kappa Q_v(t)].$$

Problem (67) has binary and multiplicative decision variables, making it non-convex. To handle these complexities, we introduce the following new variable.

$$\tilde{P}_{v,z}(t)\ :=\mathrm{I}_{v,z}(t)P_{v,z}(t). \quad (68)$$

Now, since $\mathrm{I}_{v,z}(t)\in\{0,1\}$ and $0\leq P_{v,z}(t)\leq P_v^{\max}$, $\tilde{P}_{v,z}(t)$ is equivalent to the following inequalities:

$$0\cdot\mathrm{I}_{v,z}(t)\leq\tilde{P}_{v,z}(t)\ \leq P_v^{\max}\cdot\mathrm{I}_{v,z}(t), \quad (69a)$$

$$0\cdot[1-\mathrm{I}_{v,z}(t)]\leq P_{v,z}(t)-\tilde{P}_{v,z}(t)\ \leq P_v^{\max}\cdot[1-\mathrm{I}_{v,z}(t)], \quad (69b)$$

$$0\leq\tilde{P}_{v,z}(t)\ \leq P_v^{\max}, \quad (69c)$$

Note that the binary decision variable $\mathrm{I}_{v,z}(t)\in\{0,1\}$ is equivalent to $\mathrm{I}_{v,z}(t)-(\mathrm{I}_{v,z}(t))^2=0$. Then, using the DC trick, we equivalently represent this binary decision constraint as [47], [55]:

$$\sum_{v\in\bar{\mathscr{U}}(t)}\sum_{z=1}^{Z}\mathrm{I}_{v,z}(t)-\sum_{v\in\bar{\mathscr{U}}(t)}\sum_{z=1}^{Z}(\mathrm{I}_{v,z}(t))^2\leq 0, \quad (70a)$$

$$0\leq\mathrm{I}_{v,z}(t)\leq 1,\quad \forall v\in\bar{\mathscr{U}}(t),z\in\mathscr{Z}. \quad (70b)$$

To that end, using (68-70b), we equivalently rewrite (67) as

$$\underset{\tilde{\mathbf{x}}(t)}{\text{minimize}}\ \tilde{\mathrm{U}}(t), \quad (71)$$

$$\text{s.t. } C_7,C_8,(69a),(69b),(69c),(70a),(70b), \quad (71a)$$

$$\sum_{z=1}^{Z}\tilde{P}_{v,z}(t)\ \leq P_v^{\max},\quad \forall t,\ \forall z, \quad (71b)$$

$$\tilde{r}_v(t)\geq\frac{Q_v(t-1)}{\kappa\times\big\lfloor(Z/|\mathscr{C}_k|)\times(\min\{\breve{t}_k+\big\lfloor \mathrm{t}_v^{\mathrm{soj}}(\breve{t}_k)/\kappa\big\rfloor,\breve{t}_{k+1}\}-t)\big\rfloor}, \quad (71c)$$

where $\tilde{\mathbf{x}}(t)=[\breve{\boldsymbol{I}}(t),\mathbf{p}(t),\tilde{\mathbf{p}}(t)]$, $\tilde{\mathbf{p}}(t)=[\tilde{\mathbf{p}}_1(t),\ldots,\tilde{\mathbf{p}}_{|\mathscr{C}_k|}(t)]^T$, $\tilde{\mathbf{p}}_v(t)=[\tilde{P}_v^1(t),\ldots,\tilde{P}_{v,z}(t)]^T$, $\tilde{\mathrm{U}}(t)=C\bar{\beta}(k,t)\sum_{v\in\bar{\mathscr{U}}(t)}\sum_{z=1}^{Z}\tilde{P}_{v,z}(t)-\omega(1-$

$\upsilon)\sum_{v\in\bar{\mathscr{U}}(t)}[\sum_{z=1}^{Z}\log_2(1+(\tilde{P}_{v,z}(t)\ \|\mathbf{h}_{v,z}(t)\|^2)/(\omega\varsigma^2))][C+\kappa Q_v(t)]$ and $\tilde{r}_v(t) = \omega(1-\upsilon)\sum_{z=1}^{Z}[\log_2(1+(\tilde{P}_{v,z}(t)\ \|\mathbf{h}_{v,z}(t)\|^2)/(\omega\varsigma^2))]$.

Notice that constraint (70a) makes problem (71) non-convex. Particularly, constraint (70a) is the difference between two convex functions. Thus, this constraint can be incorporated into the objective function to act as a penalty when violated. Particularly, for a sufficiently large constant ϑ, optimization problem (71) can be equivalently [55] represented as

$$\underset{\tilde{\mathbf{x}}(t)}{\text{minimize}}\ \tilde{\mathrm{U}}(t)+\vartheta\Big[\sum_{v\in\bar{\mathscr{U}}(t)}\sum_{z=1}^{Z}\mathrm{I}_{v,z}(t)-\sum_{v\in\bar{\mathscr{U}}(t)}\sum_{z=1}^{Z}(\mathrm{I}_{v,z}(t))^2\Big], \quad (72)$$

$$\text{s.t. } C_7, C_8, (69a), (69b), (69c), (70b), (71b), (71c). \quad (72a)$$

Intuitively, $\vartheta \gg 0$ penalizes the objective function when any $\mathrm{I}_{v,z}(t)$ is not either 0 or 1. Therefore, for a hefty ϑ, the pRB allocation parameter will need to be either 0 or 1 to minimize the objective function [47], [55].

Remark 8: The optimization problem (72) belongs to the DC function programming because all four terms in the objective functions are convex, and the constraints span a convex set.

To that end, we use SCA to approximate the second term of (70a). Using Taylor expansion, we write the following for a feasible point $\mathrm{I}_{v,z}(t,j)$.

$$\sum_{v\in\bar{\mathscr{U}}(t)}\sum_{z=1}^{Z}(\mathrm{I}_{v,z}(t))^2 \geq \sum_{v\in\bar{\mathscr{U}}(t)}\sum_{z=1}^{Z}(\mathrm{I}_{v,z}(t,j))^2 + \sum_{v\in\bar{\mathscr{U}}(t)}\sum_{z=1}^{Z}2\mathrm{I}_{v,z}(t,j)[\mathrm{I}_{v,z}(t)-\mathrm{I}_{v,z}(t,j)] = H(\tilde{\mathbf{I}}(t,j)), \quad (73)$$

where $\tilde{\mathbf{I}}(t,j)$ is the set of all feasible points for all v and z. As such, given an initial feasible $\tilde{\mathbf{x}}(t,j)$, we obtain the upper bound of (72) via optimizing the following convex optimization problem.

$$\underset{\tilde{\mathbf{x}}(t,j)}{\text{minimize}}\ \tilde{\mathrm{U}}(t)\ +\vartheta\Big[\sum_{v\in\bar{\mathscr{U}}(t)}\sum_{z=1}^{Z}\mathrm{I}_{v,z}(t)-H(\tilde{\mathbf{I}}(t,j))\Big], \quad (74)$$

$$\text{s.t. } C_7, C_8, (69a), (69b), (69c), (70b),(71b), (71c), \quad (74a)$$

where $\tilde{\mathbf{x}}(t,j)=[\tilde{\mathbf{I}}(t,j),\mathbf{p}(t,j),\tilde{\mathbf{p}}(t,j)]$ and $H(\tilde{\mathbf{I}}(t,j))$ is the global underestimation of $\sum_{v\in\bar{\mathscr{U}}(t)}\sum_{z=1}^{Z}(\mathrm{I}_{v,z}(t))^2$.

Note that (74) is convex and can be solved efficiently using existing solvers such as CVX [46]. Particularly, we use Algorithm 4 and CVX to solve (74) iteratively. Note that (74) has $3(|\bar{\mathscr{U}}(t)|\times Z)$ decision variables and $4(|\bar{\mathscr{U}}(t)|\times Z)+2|\bar{\mathscr{U}}(t)|+Z+1$ constraints. Besides, in the worst-case, $|\bar{\mathscr{U}}(t)|=Z$. Moreover, we need to run Algorithm 3 to get the required scheduling decisions before solving (74). As such the time complexity of running Algorithm 4 for J iterations is $\mathscr{O}\big(27JZ^6(4Z^2+3Z+1)+|\mathscr{U}(t)|^2+Z\big)$ [47]. Moreover, our SCA-based solutions of (48) and (74) converge to locally optimal solutions of the original problems (34) and (64), respectively [47], [55].

Remark 9 (VEFL Summary): At the beginning of a VEFL round k, each CV reports its $\eta_v^{\min}$, $\eta_v^{\max}$, P_v^{max}, $\mathrm{e}_v^{\mathrm{bud}}(k)$, D_v and $\xi_v(l_v(k),\eta_v(k))$ to the server as part of the SLA. The server solves (48) to jointly optimize $\mathbf{1}(k)$, $\boldsymbol{l}(k)$ and $\boldsymbol{\eta}(k)$. It then sends the most recent global model $\boldsymbol{\omega}_k$, $l_v^*(k)$'s and $\eta_v^*(k)$'s to the selected CVs. Each CV uses the optimized $\eta_v^*(k)$ to train the received ML model for $l_v^*(k)$ local rounds on its local dataset. The server and the CVs work together to receive the CVs' locally trained models $\boldsymbol{\omega}_{v,k+1}$'s. More specifically, the server keeps track of the remaining payload buffer $\mathbf{Q}(t)$, which essentially tells it how much of $\boldsymbol{\omega}_{v,k+1}$'s are yet to be received. Recall that the server estimated the worst-case required number of offloading time slots $\bar{\mathrm{t}}_v^{\mathrm{tx}}$ and determined the VEN slot $\tau(k)$ from which it shall start receiving the $\boldsymbol{\omega}_{v,k+1}$'s. The server makes the payload offloading scheduling decisions based on Algorithm 3 from this slot $\tau(k)$. It then solves (66) to optimize the transmission power if only one CV is scheduled. Otherwise, it solves (74) to optimize the pRB and power allocation jointly. The scheduled CVs then receives their $\mathrm{I}_{v,z}^*(t)$'s and $P_{v,z}^*(t)$'s from the server, and continue offloading their $\boldsymbol{\omega}_{v,k+1}$'s. To that end, the server updates the global model based on its aggregation rule defined in (28) and repeats the above processes for the next round.

Algorithm 4 Iterative pRB and Power Allocation Process

Input: Initial feasible set $\tilde{\mathbf{x}}(t,j)$, $j=0$, maximum iteration J, precision level ϵ^{prec}, initial penalty ϑ_0

Repeat:
- $j \leftarrow j+1$
- $\vartheta \leftarrow \vartheta_0 + j$
- Solve (74) using ϑ and $\tilde{\mathbf{x}}(t,j-1)$ to find $\tilde{\mathbf{x}}(t,j)$

Until converge with ϵ^{prec} precision or $j=J$

Output: pRB allocation and power allocation set $\tilde{\mathbf{x}}^*(t)$

V. Simulation Results and Discussions

A. Simulation Setting

For our RoI, we use the real-world map information of downtown Raleigh, NC, USA, from OpenStreetMap.[7] We then use SUMO to model practical microscopic mobility as presented in Section II-A. We repeat our simulation process for 5 times to get the average performance. The VEN is simulated for 2000 seconds in each of these repeats. There are 293, 297, 299, 292 and 287 unique CVs, respectively in these 5 repeats. Besides, the total CVs present in the VEN slots varies. Each CV's minimum and maximum CPU frequencies are selected randomly from $[1\times10^3, 5\times10^3]$ Hz and $[1.9\times10^9, 2.8\times10^9]$ Hz, respectively. The energy budget, required CPU cycle to process per bit data, per unit energy cost and ϕ_vs are randomly chosen from $[20,30]$ Joules, $[20,30]$, $[5,10]$ units and $[10,20]$ units, respectively. For the RAT parameters, we consider $r=500$ meter, $Z=10$, $\omega=1.8$ MHz,[8] $\bar{n}=1$, subcarrier spacing 30 KHz and $\kappa=0.5$ ms. We assume the gNB has 4 antennas,

[7] https://www.openstreetmap.org

[8] With numerology 1, the pRB size is 360 KHz. However, for the ease of faster simulation, we considered that the pRB size is 5×360 KHz.

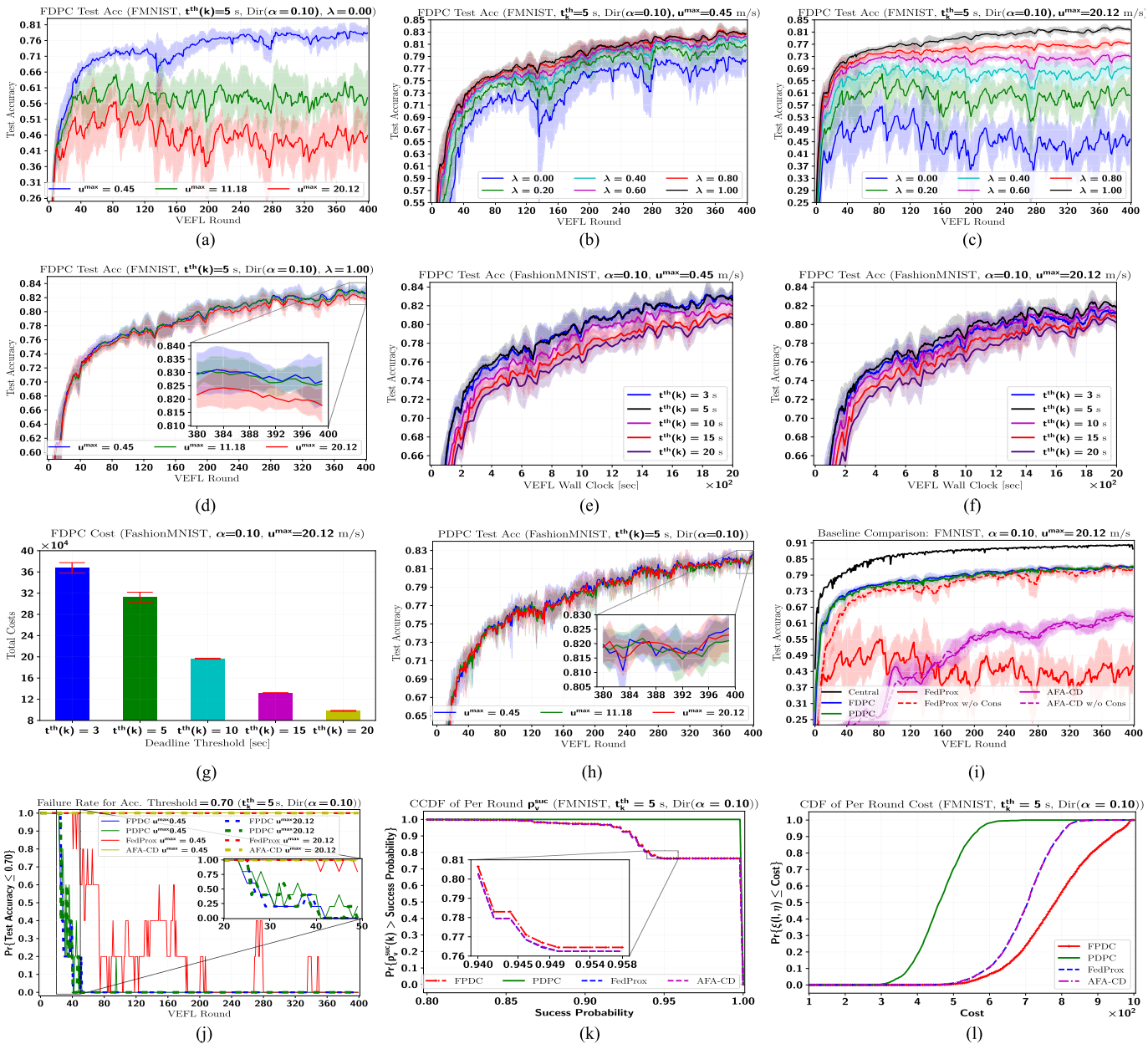

Fig. 7. (a) FDPC - impact of CV velocity: $t^{th}(k) = 5$, Dir($\alpha = 0.1$), $\lambda = 0$. (b) FDPC - impact of λ: $t^{th}(k) = 5$, Dir($\alpha = 0.1$), $u^{max} = 0.45$ m/s. (c) FDPC - impact of λ: $t^{th}(k) = 5$, Dir($\alpha = 0.1$), $u^{max} = 20.12$ m/s. (d) FDPC - performance when $\lambda = 1$: $t^{th}(k) = 5$, Dir($\alpha = 0.1$). (e) FDPC - Impact of $t^{th}(k)$: Dir($\alpha = 0.1$), $\lambda = 1$, $u^{max} = 0.45$ m/s. (f) FDPC - Impact of $t^{th}(k)$: Dir($\alpha = 0.1$), $\lambda = 1$, $u^{max} = 20.12$ m/s. (g) FDPC: total costs for different $t^{th}(k)$. (h) PDPC: Impact of CV velocity: $t^{th}(k) = 5$, Dir($\alpha = 0.1$). (i) Performance comparison: $t^{th}(k) = 5$s, $u^{max} = 20.12$m/s, Dir($\alpha = 0.1$). (j) Failure rate of achieving 70% test accuracy: $t^{th}(k) = 5$ s, $u^{max} = 0.45$ m/s, Dir($\alpha = 0.1$). (k) CCDF of success probability comparison: $t^{th}(k) = 5$, Dir($\alpha = 0.1$). (l) CDF of per round cost comparison: $t^{th}(k) = 5$, Dir($\alpha = 0.1$).

whereas the CVs have single antennas. For model training and testing, we use a) MNIST [56], b) FashionMNIST [57] c) German Traffic Sign Recognition Benchmark (GTSRB) [58] and d) CIFAR-10 [59] datasets. We use a simple convolutional neural network (CNN) with two convolutional layers, each followed by max-pooling layers, one fully connected layer and the output layer. For the non-IID data distribution, we use symmetric Dirichlet distribution Dir(α) with the concentration parameter α and use a similar approach as in [4] to distribute the labels across the unique CVs. Note that a smaller α means the label distribution across CVs is more skewed.

To this end, we study the impact of different system parameters using the FashionMNIST dataset for FDPC and PDPC. We will also use the other three datasets to compare the performances later in Section V-D. Note that we have omitted the results of the VEFL and RAT parameter optimizations for brevity.

B. Simulation Results: FDPC

1) FDPC-Impact of Velocity and λ: First, we validate our aggregation rule's necessity for different velocities. Note that in (26), if we put zero weight on the estimated sojourn period, then the server essentially aggregates the model weights

solely based on the dataset sizes of the CVs. When the FEEL clients are stationary, it makes sense to aggregate the model parameters based on the dataset sizes. However, when the clients are mobile, which is the case for CVs, solely aggregating based on dataset size may not yield the best results. This is particularly true when mobility is relatively high. The expected sojourn period of the CV decreases as the velocity increases. Therefore, a CV has less time to perform its local iterations as it must offload the trained model before moving out of the gNB's coverage area. This essentially means that the performance of the trained global model may not yield good test accuracy. A similar trend is also observed in our simulation results. Fig. 7(a) shows how test accuracy varies across the VEFL rounds for different CV velocities for $\alpha = 0.1$. We observe that the performance decreases when $\mathrm{u}^{\max}$ increases. Note that the solid lines in the figures show the mean values, while the shaded strip is the standard deviation. These simulation results also reveal that the deviation also increases with increased velocity.

In addition, it is beneficial to put more weight on the expected sojourn period. Our results in Fig. 7(b) and Fig. 7(c) also validate this claim. When $\mathrm{u}^{\max} = 0.45$ m/s, we observe that $\lambda = 0.8$ and $\lambda = 1$ yield nearly identical performance in Fig. 7(b). Besides, when $\mathrm{u}^{\max} = 20.12$ m/s, $\lambda = 1$ clearly provides the best test accuracy, which is reflected in Fig. 7(c). Fig. 7(d) verifies that our proposed aggregation policy works for all $\mathrm{u}^{\max}$s. Furthermore, the standard deviations in the test accuracy are also negligible for all velocities.

2) FDPC-Impact of $\mathrm{t}^{\mathrm{th}}(k)$: The impact of the deadline threshold $\mathrm{t}^{\mathrm{th}}(k)$ between two VEFL rounds can also affect the learning performance. Intuitively, having a smaller $\mathrm{t}^{\mathrm{th}}(k)$ means the server can run more VEFL rounds within a fixed time duration. Therefore, the trained model's test accuracy is expected to be better with a smaller $\mathrm{t}^{\mathrm{th}}(k)$. However, although this holds at low velocity, it is not straightforward at a higher $\mathrm{u}^{\max}$ since the short sojourn period may lead to a smaller $l_v(k)$. Therefore, under high mobility, the server shall carefully decide the deadline threshold $\mathrm{t}^{\mathrm{th}}(k)$ to get a reasonably trained model from the participating CVs. Our simulation results also reveal similar trends in Fig. 7(e) and Fig. 7(f). Particularly, when $\mathrm{u}^{\max} = 0.45$ m/s, we observe that $\mathrm{t}^{\mathrm{th}} = 3$ s and $\mathrm{t}^{\mathrm{th}} = 5$ s yield almost similar performances. However, at $\mathrm{u}^{\max} = 20.12$ m/s, we observe that $\mathrm{t}^{\mathrm{th}}(k) = 3$ s clearly does not provide better results over $\mathrm{t}^{\mathrm{th}}(k) = 5$ s and $\mathrm{t}^{\mathrm{th}}(k) = 10$ s. Furthermore, a shorter $\mathrm{t}^{\mathrm{th}}(k)$ also causes more VEFL rounds, which causes more expenses for the server. Fig. 7(g) shows how the total cost $\sum_{k=1}^{K}\sum_{v=1}^{V_{t_k}} \xi(l_v(k), \eta_v(k))$ vary for different $\mathrm{t}^{\mathrm{th}}(k)$. Here, we stress that a smaller $\mathrm{t}^{\mathrm{th}}(k)$ can be desirable for the server if it requires a trained global model that delivers a certain level of accuracy quickly at the cost of higher expenses. As such, depending on its operational needs, it may choose the desired deadline threshold.

C. Simulation Results: PDPC

Note that the length of subset $\mathscr{C}_k$ is a design parameter that the system administrator can choose. Based on our simulation setting, we found that $|\mathscr{C}_k| = 8$ provides the best results for MNIST and GTSRB datasets, while $|\mathscr{C}_k| = 10$ and $|\mathscr{C}_k| = 12$ work best for the FashionMNIST and CIFAR-10 datasets, respectively. Moreover, our framework is general, where the server can adjust this value in each VEFL round.

1) PDPC-Impact of Velocity: We now validate that our PDPC solution is robust against different velocities. Recall that we maximize a weighted summation of $l_v(k)$s, where we choose the weight based on expected sojourn periods and dataset sizes of the CVs. Particularly, since PDPC allows the server to choose the CVs at the beginning of the VEFL rounds, we put equal weights on the sojourn periods and the dataset sizes, i.e., $\bar{\lambda} = 0.5$ in (35). Therefore, a proper joint optimization for $\mathbf{1}(v \in \mathscr{C}_k)$, $l_v(k)$ and $\eta_v(k)$ can combat the effect of short sojourn periods under high mobility. Our simulation results also validate this. In Fig. 7(h), we observe that the test accuracies are very similar for $\mathrm{u}^{\max} = 0.45$ m/s, $\mathrm{u}^{\max} = 11.18$ m/s and $\mathrm{u}^{\max} = 20.12$ m/s. Particularly, when $\mathrm{u}^{\max} = 0.45$ m/s and the VEFL rounds are $k = 100$, $k = 200$, $k = 300$ and $k = 400$, the test accuracies are 76.71%, 79.19%, 81.17% and 82.53%, respectively. For the same VEFL rounds, when $\mathrm{u}^{\max} = 11.18$ m/s, the test accuracies are 76.17%, 78.56%, 80.67% and 82.11%, respectively. Besides, when $\mathrm{u}^{\max} = 20.12$ m/s, the test accuracies are 76.08%, 78.74%, 81.12% and 82.3%, respectively.

Note that PDPC performs similarly to FDPC with respect to different $\mathrm{t}^{\mathrm{th}}(k)$, which we skip presenting for brevity.

D. Performance Comparisons

We now compare the performances of our proposed FDPC and PDPC VEFL schemes with the state-of-the-art FedProx [23] and anarchic federated averaging - cross-device (AFA-CD) [60] baselines. For FedProx and AFA-CD, we consider two cases, namely, (1) all constraints are present, and (2) no constraints are enforced. Note that the latter is the ideal case that does not consider any delay, energy, monetary or radio resource constraints. Besides, for the baselines with system constraints, we assume that the server expends an equal amount of its budget for all CVs. Furthermore, taking the maximum CPU frequencies $\eta_v^{\max}$'s, the $l_v(k)$'s are chosen to satisfy the time and energy constraints. Moreover, we use the same RAT solution for offloading the CVs' trained models. Note that the baselines with constraints are expected to perform worse when the velocity increases since they do not consider mobility in the model weights aggregation. We also compare the results with the centralized ML, which serves as the performance upper bound as the server can access all training data.

To that end, we compare these schemes with $\mathrm{u}^{\max} = 20.12$ m/s in Fig. 7(i). When VEFL rounds are $k = 200$, $k = 300$ and $k = 400$, the test accuracies with FDPC are 78.91%, 81.43% and 81.77%, respectively. On the other hand, PDPC delivers 78.74%, 81.12% and 82.30% test accuracies for the same respective VEFL rounds. By contrast, for the same VEFL rounds, when all constraints are present, FedProx returns 35.89%, 39.94% and 45.07%, while AFA-CD yields 55.03%, 60.51% and 63.59 % test accuracies. Moreover, in the ideal case, i.e., without (W/O) constraints, FedProx delivers

TABLE I
TEST ACCURACY WITH TRAINED $\boldsymbol{\omega}_K$

Dataset	Dir(α)	FDPC (Proposed)	PDPC (Proposed)	FedProx [23] with Constraints	AFA-CD [60] with Constraints	FedProx [23] W/O Constraints	AFA-CD [60] W/O Constraints	Centralized ML
MNIST	0.1	0.9677 ± 0.0023	$\mathbf{0.9691} \pm 0.0024$	0.4566 ± 0.1497	0.7543 ± 0.112	0.9723 ± 0.0012	0.8285 ± 0.0209	0.9905
	0.9	0.9779 ± 0.0022	$\mathbf{0.9787} \pm 0.0013$	0.8906 ± 0.0191	0.8248 ± 0.0994	0.982 ± 0.0017	0.8994 ± 0.0049	
	10	$\mathbf{0.9816} \pm 0.0009$	$\mathbf{0.9816} \pm 0.0006$	0.9216 ± 0.0033	0.8294 ± 0.1025	0.9838 ± 0.0011	0.9136 ± 0.0031	
FMNIST	0.1	0.8177 ± 0.0064	$\mathbf{0.823} \pm 0.006$	0.4507 ± 0.0888	0.6359 ± 0.0262	0.8098 ± 0.0181	0.6319 ± 0.0161	0.8879
	0.9	$\mathbf{0.857} \pm 0.0024$	0.8527 ± 0.0032	0.6457 ± 0.0251	0.6933 ± 0.0058	0.8649 ± 0.0033	0.6943 ± 0.0136	
	10	$\mathbf{0.8641} \pm 0.0019$	0.8526 ± 0.0033	0.6846 ± 0.0142	0.7061 ± 0.003	0.8744 ± 0.0021	0.7057 ± 0.0033	
GTSRB	0.1	0.4838 ± 0.0123	$\mathbf{0.488} \pm 0.0049$	0.104 ± 0.026	0.1247 ± 0.0144	0.5568 ± 0.0118	0.1246 ± 0.0139	0.9166
	0.9	0.614 ± 0.0078	$\mathbf{0.6165} \pm 0.0187$	0.2045 ± 0.0191	0.1628 ± 0.0146	0.6622 ± 0.0122	0.1592 ± 0.0106	
	10	0.6159 ± 0.0164	$\mathbf{0.6438} \pm 0.0182$	0.2195 ± 0.0183	0.1602 ± 0.0144	0.6873 ± 0.0085	0.1524 ± 0.0077	
CIFAR-10	0.1	$\mathbf{0.5043} \pm 0.0122$	0.4966 ± 0.0081	0.2047 ± 0.0211	0.256 ± 0.02141	0.5043 ± 0.0213	0.2484 ± 0.0168	0.7394
	0.9	0.589 ± 0.006	$\mathbf{0.595} \pm 0.0081$	0.2803 ± 0.022	0.3229 ± 0.0045	0.6124 ± 0.0129	0.3282 ± 0.0038	
	10	0.6175 ± 0.0022	$\mathbf{0.6202} \pm 0.009$	0.3305 ± 0.0097	0.3304 ± 0.0097	0.6293 ± 0.0026	0.3274 ± 0.0026	

77.75%, 80.09% and 80.98%, and AFA-CD returns 56.26%, 59.04% and 63.19% test accuracies. These results suggest that our proposed FDPC and PDPC schemes significantly outperform the baselines. It is worth noting that FedProx W/O constraints performs similarly to our solutions with constraints, while the AFA-CD fails to achieve good performance. This is due to the fact that FedProx also incorporates a similar local loss function for the clients, whereas AFA-CD does not incorporate any proximal terms and employs a different aggregation rule. Besides, the gap with the centralized ML is expected due to the inherent nature of the system model. More specifically, all training data samples are distributed among the CVs without repetitions. As such, once a CV moves out of the communication area, its training data samples are no longer available. Moreover, these CVs may only perform a few local rounds before moving out of the gNB's coverage.

Besides, unlike in the ideal case, under all system constraints, FedProx suffers from potential model divergence. In such a constrained case, notice that the best test accuracy of 55.06% is achieved with FedProx when $k = 81$. However, $\boldsymbol{\omega}_k$ does not converge and fluctuates rapidly, as shown in Fig. 7(i). It is even more evident in Fig. 7(j), which shows the failure rate of achieving a test accuracy of 70% for different $\mathrm{u}^{\max}$. We can see that FDPC achieves a higher test accuracy than the accuracy threshold for all velocities after about 51 VEFL rounds with probability 1. On the other hand, we notice a few oscillations with PDPC. Particularly, the probability of having a higher test accuracy than the threshold becomes one after about 51 VEFL rounds, notwithstanding a 20% drop at VEFL round $k = 95$. However, from $k = 96$ onward, PDPC delivers a higher test accuracy than the threshold with probability 1. Contrary to these guaranteed performances with FDPC and PDPC, the performance of FedProx with constraints is unreliable. Even at low mobility, $\mathrm{u}^{\max} = 0.45$ m/s, we observe that FedProx's performance oscillates. This baseline shows that the test accuracy can be less than the threshold 20% of the time, even at VEFL round $k = 349$ with relatively low mobile CVs. Furthermore, AFA-CD fails to achieve the desired 70% accuracy.

Although PDPC yields slightly higher performance oscillations than FDPC initially, it is still a practical solution in resource-constrained circumstances. To further study its benefits, we now examine the CCDF of $\mathrm{p}_v^{\mathrm{suc}}(k)$. Recall that for FDPC, FedProx and AFA-CD, all CVs with SLAs receive the global model and participate in the training process. However, due to mobility, some CVs may not finish even a single local iteration within the allocated deadline. Besides, the energy constraint can also potentially restrict some CVs from training and offloading their models. On the other hand, in PDPC, this can be mitigated by appropriately selecting the subset $\mathscr{C}_k$. In any case, if the server does not receive a CV's local model within $\mathrm{t}^{\mathrm{th}}(k)$, it considers that as a failure. Our simulation results in Fig. 7(k) suggest that in all VEFL rounds, a CV successfully offloads its locally trained model to the server with at least 0.9 probability 97.3%, 100%, 97.1% and 97% percent of the times for FDPC, PDPC, FedProx and AFA-CD, respectively. Moreover, a CV has a success probability of at least 0.99 for 76.45% and 100% of the VEFL rounds, respectively, in FDPC, PDPC, and 76.25% of the VEFL rounds for FedProx and AFA-CD, with all constraints present.

PDPC not only guarantees a 100% trained models reception but also yields a lesser expense for the server. This is due to the fact that the server only needs to pay the fees to the selected CVs in the subset $\mathscr{C}_k$. Our simulation results in Fig. 7(l) show that, in a VEFL round, the server's cost is less than 700 units for about 23.75%, 100%, 46.8% and 47.3% of the times, respectively, for FDPC, PDPC, FedProx and AFA-CD. Note that PDPC not only requires a lesser expense but also delivers near identical test accuracies to FDPC.

Table I shows the performance comparisons on MNIST, FashionMNIST, GTSRB and CIFAR-10 datasets with the obtained global model after $K = 400$ VEFL rounds when $\mathrm{u}^{\max} = 20.12$ m/s, $\mathrm{t}^{\mathrm{th}} = 5$ s and $\Xi(k) = 1000$ units. Under the resource-constrained scenario, it can be observed that both FDPC and PDPC perform similarly while the baselines lag significantly. Furthermore, even without constraints, AFA-CD fails to perform reasonably in GTSRB and CIFAR-10 datasets. Moreover, the proposed schemes achieve performance comparable to FedProx W/O constraints, which validates the effectiveness of our proposed solutions under extreme resource constraints.

VI. Conclusion

A novel vehicular edge federated learning framework that leverages a 5G-NR RAT and the limited computation powers of the moving CVs is proposed in this work. Under delay, energy and cost constraints, first, subset CV selection, local iterations and CPU frequencies are jointly optimized given the estimated worst-case sojourn period and communication delay and cost. Then, the RAT parameters are jointly optimized using an online per-slot-based stochastic optimization technique. Extensive simulation results suggest that the server can combat high mobility by aggregating the trained model parameters using a weighted combination of sojourn periods and dataset sizes in FDPC, whereas by appropriately selecting a subset of CVs in PDPC. Moreover, the proposed method outperforms the state-of-the-art FedProx and AFA-CD algorithms under the same constraints in all the examined scenarios.

Appendix A
Proof of Theorem 1

Using Taylor expansion, we can write the following:

$$f(\boldsymbol{\omega}_{k+1}) = f(\boldsymbol{\omega}_k) + \langle \nabla f(\boldsymbol{\omega}_k), \boldsymbol{\omega}_{k+1} - \boldsymbol{\omega}_k \rangle + \frac{1}{2!} (\boldsymbol{\omega}_{k+1} - \boldsymbol{\omega}_k)^T \nabla^2 f(\boldsymbol{\omega}_k) (\boldsymbol{\omega}_{k+1} - \boldsymbol{\omega}_k) + h(o), \quad (75)$$
$$\overset{(a)}{\leq} f(\boldsymbol{\omega}_k) + \underbrace{\langle \nabla f(\boldsymbol{\omega}_k), \boldsymbol{\omega}_{k+1} - \boldsymbol{\omega}_k \rangle}_{\text{Term 1}} + \underbrace{\frac{L}{2} \|\boldsymbol{\omega}_{k+1} - \boldsymbol{\omega}_k\|^2}_{\text{Term 2}},$$

where $h(o)$ represents higher-order terms. We reach to (a) by ignoring $h(o)$ and using *Assumption* 1.

Bound Term 2: Let $\hat{\boldsymbol{\omega}}_{v,k+1} = \arg\min f_v(\boldsymbol{\omega}, \boldsymbol{\omega}_k)$. Then, due to μ'-strong convexity of $f_v(\boldsymbol{\omega}, \boldsymbol{\omega}_k)$, we can write

$$\begin{aligned}
&\|\hat{\boldsymbol{\omega}}_{v,k+1} - \boldsymbol{\omega}_{v,k+1}\| \\
&\leq \frac{1}{\mu'} \|\nabla f_v(\hat{\boldsymbol{\omega}}_{v,k+1}, \boldsymbol{\omega}_k) - \nabla f_v(\boldsymbol{\omega}_{v,k+1}, \boldsymbol{\omega}_k)\|, \\
&= (1/\mu') \|\nabla f_v(\boldsymbol{\omega}_{v,k+1}, \boldsymbol{\omega}_k)\| \overset{(a)}{\leq} (\gamma/\mu') \|\nabla f_v(\boldsymbol{\omega}_k, \boldsymbol{\omega}_k)\|, \\
&= (\gamma/\mu') \|\nabla F_v(\boldsymbol{\omega}_k)\| \overset{(b)}{\leq} [(B\gamma)/\mu'] \|\nabla f(\boldsymbol{\omega}_k)\|, \quad (76)
\end{aligned}$$

where we obtain (a) and (b) using *Assumption* 4, i.e., γ-inexact local solvers and *Assumption* 2, i.e., B-dissimilarity, respectively. Similarly, we can write

$$\|\hat{\boldsymbol{\omega}}_{v,k+1} - \boldsymbol{\omega}_k\| \leq \left(\frac{1}{\mu'}\right) \|\nabla F_v(\boldsymbol{\omega}_k)\| = \left(\frac{B}{\mu'}\right) \|\nabla f(\boldsymbol{\omega}_k)\|. \quad (77)$$

Then, using triangle inequality we can write

$$\|\boldsymbol{\omega}_{v,k+1} - \boldsymbol{\omega}_k\| \leq [(B(1+\gamma))/\mu'] \|\nabla f(\boldsymbol{\omega}_k)\|. \quad (78)$$

Now using (78) and (28), we get

$$\begin{aligned}
&\|\boldsymbol{\omega}_{k+1} - \boldsymbol{\omega}_k\|^2 \\
&= \left\| \sum_{v=1}^{V_{t_k}} \left[p_v \cdot \frac{\mathbf{1}\left(v \in \mathscr{C}_k, \mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)}{q_v(k)\, p_v^{\mathrm{suc}}(k)} \right. \right. \\
&\left. \left. \times (\boldsymbol{\omega}_{v,k+1} - \boldsymbol{\omega}_k) \right] \right\|^2 \\
&\overset{(a)}{\leq} \sum_{v=1}^{V_{t_k}} p_v \cdot \left\| \frac{\mathbf{1}\left(v \in \mathscr{C}_k, \mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)}{q_v(k) p_v^{\mathrm{suc}}(k)} \right. \\
&\left. \times (\boldsymbol{\omega}_{v,k+1} - \boldsymbol{\omega}_k) \right\|^2, \\
&\overset{(b)}{=} \sum_{v=1}^{V_{t_k}} p_v \cdot \left(\frac{\mathbf{1}\left(v \in \mathscr{C}_k, \mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)}{q_v(k) p_v^{\mathrm{suc}}(k)} \right)^2 \\
&\times \|\boldsymbol{\omega}_{v,k+1} - \boldsymbol{\omega}_k\|^2, \\
&\overset{()}{=} \frac{B^2(1+\gamma)^2}{\mu'^2} \|\nabla f(\boldsymbol{\omega}_k)\|^2 \\
&\times \sum_{v=1}^{V_{t_k}} p_v \cdot \left(\frac{\mathbf{1}\left(v \in \mathscr{C}_k, \mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)}{q_v(k) p_v^{\mathrm{suc}}(k)} \right)^2, \quad (79)
\end{aligned}$$

where (a) follows from Jensen's inequality due to the convexity of $\|\cdot\|^2$. (b) stems using the fact that $\frac{\mathbf{1}\left(v \in \mathscr{C}_k, \mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)}{q_v(k) p_v^{\mathrm{suc}}(k)}$ is scalar. Moreover, we get to (c) using (78).

Bound Term 1: Using the aggregation rule defined in (28), we can write

$$\begin{aligned}
&\langle \nabla f(\boldsymbol{\omega}_k), \boldsymbol{\omega}_{k+1} - \boldsymbol{\omega}_k \rangle = \left\langle \nabla f(\boldsymbol{\omega}_k), \sum_{v=1}^{V_{t_k}} \left[p_v \cdot \right. \right. \\
&\left. \left. \frac{\mathbf{1}\left(v \in \mathscr{C}_k, \mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)}{q_v(k) p_v^{\mathrm{suc}}(k)} (\boldsymbol{\omega}_{v,k+1} - \boldsymbol{\omega}_k) \right] \right\rangle, \\
&= \sum_{v=1}^{V_{t_k}} p_v \cdot \frac{\mathbf{1}\left(v \in \mathscr{C}_k, \mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)}{q_v(k) p_v^{\mathrm{suc}}(k)} \\
&\times \langle \nabla f(\boldsymbol{\omega}_k), \boldsymbol{\omega}_{v,k+1} - \boldsymbol{\omega}_k \rangle, \\
&\leq B(1+\gamma)/(\mu') \|\nabla f(\boldsymbol{\omega}_k)\|^2 \sum_{v=1}^{V_{t_k}} p_v \\
&\times \frac{\mathbf{1}\left(v \in \mathscr{C}_k, \mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)}{q_v(k) p_v^{\mathrm{suc}}(k)}. \quad (80)
\end{aligned}$$

Then, plugging (80) and (79) in (75), we get the following

$$\begin{aligned}
f(\boldsymbol{\omega}_{k+1}) &\leq f(\boldsymbol{\omega}_k) + B(1+\gamma)/(\mu') \|\nabla f(\boldsymbol{\omega}_k)\|^2 \\
&\times \sum_{v=1}^{V_{t_k}} p_v \cdot \frac{\mathbf{1}\left(v \in \mathscr{C}_k, \mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)}{q_v(k) p_v^{\mathrm{suc}}(k)} \\
&+ \frac{LB^2(1+\gamma)^2}{2\mu'^2} \|\nabla f(\boldsymbol{\omega}_k)\|^2 \sum_{v=1}^{V_{t_k}} p_v \\
&\cdot \left(\frac{\mathbf{1}\left(v \in \mathscr{C}_k, \mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)}{q_v(k) p_v^{\mathrm{suc}}(k)} \right)^2. \quad (81)
\end{aligned}$$

Therefore, we can calculate the expected training loss decrease in one global round by taking the expectation of (81) as

$$\begin{aligned}
&\mathbb{E}[f(\boldsymbol{\omega}_{k+1})] - f(\boldsymbol{\omega}_k) \\
&\leq \mathbb{E}\left\{ B(1+\gamma)/(\mu') \|\nabla f(\boldsymbol{\omega}_k)\|^2 \right. \\
&\left. \times \sum_{v=1}^{V_{t_k}} p_v \cdot \frac{\mathbf{1}\left(v \in \mathscr{C}_k, \mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)}{q_v(k) p_v^{\mathrm{suc}}(k)} \right\} \\
&+ \mathbb{E}\left\{ \frac{LB^2(1+\gamma)^2}{2\mu'^2} \|\nabla f(\boldsymbol{\omega}_k)\|^2 \right.
\end{aligned}$$

$$\times \sum_{v=1}^{V_{t_k}} p_v \cdot \left(\frac{\mathbf{1}\left(v \in \mathscr{C}_k, \mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)}{q_v(k) p_v^{\mathrm{suc}}(k)} \right)^2 \Bigg\},$$

$$= \frac{B\left(1+\gamma\right)}{\mu'} \left\|\nabla f(\boldsymbol{\omega}_k)\right\|^2 + \frac{LB^2(1+\gamma)^2}{2\mu'^2} \left\|\nabla f(\boldsymbol{\omega}_k)\right\|^2$$
$$\times \sum_{v=1}^{V_{t_k}} p_v \cdot \frac{\mathbb{E}\left\{\left[\mathbf{1}\left(v \in \mathscr{C}_k, \mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)\right]^2\right\}}{q_v(k)^2 p_v^{\mathrm{suc}}(k)^2},$$

$$\overset{(a)}{=} \frac{B\left(1+\gamma\right)}{\mu'} \left\|\nabla f(\boldsymbol{\omega}_k)\right\|^2 + \frac{LB^2(1+\gamma)^2}{2\mu'^2} \left\|\nabla f(\boldsymbol{\omega}_k)\right\|^2$$
$$\times \sum\nolimits_{v=1}^{V_{t_k}} [p_v / (q_v(k)^2 p_v^{\mathrm{suc}}(k)^2)]$$
$$\times \mathbb{E}\Big\{ \left[\mathbf{1}\left(v \in \mathscr{C}_k\right) \times \mathbf{1}\left(\mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)\right]^2 \Big\},$$

$$\overset{(b)}{=} \frac{B\left(1+\gamma\right)}{\mu'} \left\|\nabla f(\boldsymbol{\omega}_k)\right\|^2 + \frac{LB^2(1+\gamma)^2}{2\mu'^2} \left\|\nabla f(\boldsymbol{\omega}_k)\right\|^2$$
$$\times \sum\nolimits_{v=1}^{V_{t_k}} [p_v / (q_v(k)^2 p_v^{\mathrm{suc}}(k)^2)] \times$$
$$\mathbb{E}\Big\{ \left[\mathbf{1}\left(v \in \mathscr{C}_k\right)\right]^2 \times \mathbb{E}\left[\mathbf{1}\left(\mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)\right]^2 \Big\},$$

$$= \frac{B\left(1+\gamma\right)}{\mu'} \left\|\nabla f(\boldsymbol{\omega}_k)\right\|^2 + \frac{LB^2(1+\gamma)^2}{2\mu'^2} \left\|\nabla f(\boldsymbol{\omega}_k)\right\|^2$$
$$\times \sum\nolimits_{v=1}^{V_{t_k}} [p_v / (q_v(k) p_v^{\mathrm{suc}}(k))],$$

$$= \frac{B(1+\gamma)}{\mu'} \left[1 + \frac{BL(1+\gamma)}{2\mu'} \sum_{v=1}^{V_{t_k}} \frac{p_v}{q_v(k) p_v^{\mathrm{suc}}(k)}\right] \left\|\nabla f(\boldsymbol{\omega}_k)\right\|^2, \quad (82)$$

where we write (a) using the fact that $\mathbf{1}\left(v \in \mathscr{C}_k\right)$ and $\mathbf{1}\left(\mathrm{t}_v(k) \ \leq \mathrm{t}^{\mathrm{th}}(k) | d_v(k) \leq \mathrm{r}\right)$ are independent. Besides, in (b) the outer and inner expectations are with respect to CV sampling and successful trained model receptions, respectively.

Appendix B
Proof of Lemma 1

Using the remaining payload buffer in (50), we can write

$$Q_v(t+1)^2 = \left([Q_v(t) - \kappa \cdot r_v(t)]^+\right)^2,$$
$$\leq Q_v(t)^2 + \kappa^2 r_v(t)^2 - 2\kappa r_v(t) Q_v(t),$$
$$Q_v(t+1)^2 - Q_v(t)^2$$
$$\leq \kappa^2 r_v(t)^2 - 2\kappa r_v(t) Q_v(t),$$
$$(1/2) \sum_{v=1}^{|\mathscr{C}_k|} \left[Q_v(t+1)^2 - Q_v(t)^2\right]$$
$$\leq -\sum_{v=1}^{|\mathscr{C}_k|} \kappa r_v(t) Q_v(t) + (1/2) \sum_{v=1}^{|\mathscr{C}_k|} \kappa^2 r_v(t)^2, \quad (83)$$

where the last inequality is obtained by dividing both sides by 2 and summing up the inequalities for all $v \in \mathscr{V}_{t_k}$. Then, using (11), we can write

$$L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t))$$
$$\leq -\sum_{v=1}^{|\mathscr{C}_k|} \kappa r_v(t) Q_v(t)$$
$$+ (\kappa^2 \omega^2 (1-\upsilon)^2/2) \sum_{v=1}^{|\mathscr{C}_k|} \left[\mathrm{I}_v(t)^2 \Big(\sum_{z=1}^{Z} \log_2(1+\Gamma_{v,z}(t)) \Big)^2\right],$$
$$\overset{(a)}{\leq} -\sum_{v=1}^{|\mathscr{C}_k|} \kappa r_v(t) Q_v(t) + (\kappa^2 \omega^2 (1-\upsilon)^2/2) \sum_{v=1}^{|\mathscr{C}_k|}$$
$$\times \left[\mathrm{I}_v(t)^2 \cdot Z \sum_{z=1}^{Z} \left(\log_2(1+\Gamma_{v,z}(t))\right)^2\right],$$
$$\overset{(b)}{\leq} -\sum_{v=1}^{|\mathscr{C}_k|} \kappa r_v(t) Q_v(t) + [(Z\kappa^2 \omega^2 (1-\upsilon)^2)/(\ln 2)^2]$$
$$\times \Big[\sum_{v=1}^{|\mathscr{C}_k|} \mathrm{I}_v(t)^2 \cdot \sum_{z=1}^{Z} (\mathrm{I}_{v,z}(t) \cdot P_{v,z}(t) \left\|\mathbf{h}_{v,z}(t)\right\|^2)/(\omega \varsigma^2)\Big], \quad (84)$$

where (a) comes from Cauchy-Schwarz inequality on real numbers $\left(\sum_{z=1}^{Z} a_z \cdot 1\right)^2 \leq \left(\sum_{z=1}^{Z} a_z^2\right) \cdot \left(\sum_{z=1}^{Z} 1^2\right)$ and (b) stems from $(\log_2(1+a))^2 \leq (2a/(\ln 2)^2)$.

Now, since both $\mathrm{I}_v(t)$ and $\mathrm{I}_{v,z}(t)$ are binary indicator functions with maximum value of 1, we can write the following:

$$\frac{Z\kappa^2\omega^2(1-\upsilon)^2}{(\ln 2)^2} \sum_{v=1}^{|\mathscr{C}_k|} \mathrm{I}_v(t)^2 \cdot \sum_{z=1}^{Z} \frac{\mathrm{I}_{v,z}(t) \cdot P_{v,z}(t) \left\|\mathbf{h}_{v,z}(t)\right\|^2}{\omega \varsigma^2}$$
$$\overset{(a)}{\leq} \frac{Z\kappa^2\omega^2(1-\upsilon)^2}{(\ln 2)^2} \sum_{v=1}^{|\mathscr{C}_k|} 1^2 \cdot \sum_{z=1}^{Z} \frac{1 \cdot \frac{P_v^{\max}}{Z} \left\|\mathbf{h}_{v,z}(t)\right\|^2}{\omega \varsigma^2},$$
$$= \frac{\kappa^2\omega^2(1-\upsilon)^2}{(\ln 2)^2} \sum_{v=1}^{|\mathscr{C}_k|} \sum_{z=1}^{Z} (P_v^{\max} \left\|\mathbf{h}_{v,z}(t)\right\|^2)/(\omega \varsigma^2) = \varpi, \quad (85)$$

where, in (a), we use the fact that $P_v^{\max}$ is equally distributed among all pRBs.

Then, using ϖ, adding $C \cdot (-\sum_{v=1}^{|\mathscr{C}_k|} r_v(t) + \bar{\beta}(k,t) \sum_{v=1}^{|\mathscr{C}_k|} \sum_{z=1}^{Z} P_{v,z}(t))$ on both sides of (83) and taking expectation on both sides conditioned on the queue state $Q_v(t)$, we reach to (62).

References

[1] A. Qayyum, M. Usama, J. Qadir, and A. Al-Fuqaha, "Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 998–1026, 2nd Quart., 2020.

[2] H. Ye, L. Liang, G. Y. Li, J. Kim, L. Lu, and M. Wu, "Machine learning for vehicular networks: Recent advances and application examples," *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 94–101, Jun. 2018.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[4] R. Jin, X. He, and H. Dai, "Communication efficient federated learning with energy awareness over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 5204–5219, Jul. 2022.

[5] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, May 2020.

[6] M. Salehi and E. Hossain, "Federated learning in unreliable and resource-constrained cellular wireless networks," *IEEE Trans. Commun.*, vol. 69, no. 8, pp. 5136–5151, Aug. 2021.

[7] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[8] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.

[9] T. Zeng et al., "Multi-task federated learning for traffic prediction and its application to route planning," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jul. 2021, pp. 451–457.

[10] M. F. Pervej et al., "Mobility, communication and computation aware federated learning for Internet of Vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2022, pp. 750–757.

[11] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1146–1159, Nov. 2020.

[12] X. Huang, P. Li, R. Yu, Y. Wu, K. Xie, and S. Xie, "FedParking: A federated learning based parking space estimation with parked vehicle assisted edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9355–9368, Sep. 2021.

[13] X. Zhou, W. Liang, J. She, Z. Yan, and K. Wang, "Two-layer federated learning with heterogeneous model aggregation for 6G supported Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5308–5317, Jun. 2021.

[14] Q. Kong et al., "Privacy-preserving aggregation for federated learning-based navigation in vehicular fog," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8453–8463, Dec. 2021.

[15] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, and M. S. Hossain, "Mobility-aware proactive edge caching for connected vehicles using federated learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5341–5351, Aug. 2021.

[16] G. Wang, F. Xu, H. Zhang, and C. Zhao, "Joint resource management for mobility supported federated learning in Internet of Vehicles," *Future Gener. Comput. Syst.*, vol. 129, pp. 199–211, Apr. 2022.

[17] F. Sun, Z. Zhang, S. Zeadally, G. Han, and S. Tong, "Edge computing-enabled Internet of Vehicles: Towards federated learning empowered scheduling," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 10088–10103, Sep. 2022.

[18] S. B. Prathiba, G. Raja, S. Anbalagan, K. Dev, S. Gurumoorthy, and A. P. Sankaran, "Federated learning empowered computation offloading and resource management in 6G-V2X," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3234–3243, Sep. 2022.

[19] T. Zeng, O. Semiari, M. Chen, W. Saad, and M. Bennis, "Federated learning on the road autonomous controller design for connected and autonomous vehicles," *IEEE Trans. Wireless Commun.*, vol. 21, no. 12, pp. 10407–10423, Dec. 2022.

[20] H. Xiao, J. Zhao, Q. Pei, J. Feng, L. Liu, and W. Shi, "Vehicle selection and resource optimization for federated learning in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11073–11087, Aug. 2021.

[21] A. Taïk, Z. Mlika, and S. Cherkaoui, "Clustered vehicular federated learning: Process and optimization," *IEEE Trans. Intel. Transp. Syst.*, vol. 23, no. 12, pp. 25371–25383, Dec. 2022.

[22] S. Liu, J. Yu, X. Deng, and S. Wan, "FedCPF: An efficient-communication federated learning approach for vehicular edge computing in 6G communication networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1616–1629, Feb. 2022.

[23] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. MLSys*, vol. 2, 2020, pp. 429–450.

[24] F. Liang, Q. Yang, R. Liu, J. Wang, K. Sato, and J. Guo, "Semi-synchronous federated learning protocol with dynamic aggregation in Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4677–4691, May 2022.

[25] Z. Yang, X. Zhang, D. Wu, R. Wang, P. Zhang, and Y. Wu, "Efficient asynchronous federated learning research in the Internet of Vehicles," *IEEE Internet Things J.*, vol. 10, no. 9, pp. 7737–7748, May 2023.

[26] Q. Chen, Z. You, and H. Jiang, "Semi-asynchronous hierarchical federated learning for cooperative intelligent transportation systems," 2021, *arXiv:2110.09073*.

[27] X. Lin, "An overview of 5G advanced evolution in 3GPP release 18," *IEEE Commun. Standards Mag.*, vol. 6, no. 3, pp. 77–83, Sep. 2022.

[28] *Technical Specification Group Services & System Aspects; Study on 5G System Support for AI/ML-Based Services (Release 18)*, document TR 23.700-80, 3rd Generation Partnership Project, Version 18.0.0, Dec. 2022.

[29] *Technical Specification Group Radio Access Network; NR; Physical Channels and Modulation*, document 3GPP TS 38.211 3rd Generation Partnership Project, Version 16.2.0, Release 16, Jul. 2020.

[30] P. A. Lopez et al., "Microscopic traffic simulation using SUMO," in *Proc. ITSC*, Nov. 2018, pp. 2575–2582.

[31] T. Kim et al., "MoDEMS: Optimizing edge computing migrations for user mobility," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 3, pp. 675–689, Mar. 2023.

[32] T. Guo and A. Suárez, "Enabling 5G RAN slicing with EDF slice scheduling," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2865–2877, Mar. 2019.

[33] *Technical Specification Group Services and System Aspects; System architecture for the 5G System (5GS); Stage 2 (Release 18)*, document TS 23.501, Version 18.0.0, 3rd Generation Partnership Project, Dec. 2022.

[34] *Technical Specification Group Services and System Aspects; Procedures for the 5G System (5GS); Stage 2 (Release 18)*, document TS 23.502, Version 18.0.0, 3rd Generation Partnership Project, Dec. 2022.

[35] R. R. Roy, *Handbook of Mobile Ad Hoc Networks for Mobility Models*. New York, NY, USA: Springer, doi: 10.1007/978-1-4419-6050-4.

[36] *Technical Specification Group Radio Access Network; NR;Physical Layer Procedures for Control*, document 3GPP TS 38.213, Version 17.0.0, Release 17, 3rd Generation Partnership Project, Dec. 2021.

[37] *Technical Specification Group Radio Access Network; Study on Channel Model for Frequencies From 0.5 to 100 GHz*, document 3GPP TR 38.901, Version 16.1.0, Release 16, 3rd Generation Partnership Project, Dec. 2019.

[38] *Technical Specification Group Radio Access Network; NR; Physical Layer Procedures for Data (Release 17)*, document TS 38.214, Version 17.4.0, 3rd Generation Partnership Project, Dec. 2022.

[39] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. V. Poor, "Fast-convergent federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 201–218, Jan. 2021.

[40] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in *Proc. ICLR*, 2020, pp. 1–26.

[41] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[42] X. Ma, H. Sun, and R. Q. Hu, "Scheduling policy and power allocation for federated learning in NOMA based MEC," in *Proc. IEEE GLOBECOM*, Dec. 2020, pp. 1–7.

[43] R. Hamdi, M. Chen, A. B. Said, M. Qaraqe, and H. V. Poor, "Federated learning over energy harvesting wireless networks," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 92–103, Jan. 2022.

[44] Z. Wang et al., "Federated learning via intelligent reflecting surface," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 808–822, Feb. 2022.

[45] G. C. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. New York, NY, USA: Springer, doi: 10.1007/978-1-4614-0676-1.

[46] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2909–2913, Jan. 2016.

[47] Y. Sun, D. W. K. Ng, Z. Ding, and R. Schober, "Optimal joint power and subcarrier allocation for full-duplex multicarrier non-orthogonal multiple access systems," *IEEE Trans. Commun.*, vol. 65, no. 3, pp. 1077–1091, Mar. 2017.

[48] B. Liu, C. Liu, and M. Peng, "Resource allocation for energy-efficient MEC in NOMA-enabled massive IoT networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 1015–1027, Apr. 2021.

[49] W. Dinkelbach, "On nonlinear fractional programming," *Manage. Sci.*, vol. 13, no. 7, pp. 492–498, Mar. 1967.

[50] Y. Zhang, C. Li, T. H. Luan, Y. Fu, W. Shi, and L. Zhu, "A mobility-aware vehicular caching scheme in content centric networks: Model and optimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3100–3112, Apr. 2019.

[51] B. Xu, P. Zhu, J. Li, D. Wang, and X. You, "Joint long-term energy efficiency optimization in C-RAN with hybrid energy supply," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11128–11138, Oct. 2020.

[52] H. Hu, W. Song, Q. Wang, R. Q. Hu, and H. Zhu, "Energy efficiency and delay tradeoff in an MEC-enabled mobile IoT network," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 15942–15956, Sep. 2022.

[53] P. Guan and X. Deng, "Maximize potential reserved task scheduling for URLLC transmission and edge computing," in *Proc. VTC-Fall*, Nov. 2020, pp. 1–5.

[54] M. F. Pervej, R. Jin, S.-C. Lin, and H. Dai, "Efficient content delivery in user-centric and cache-enabled vehicular edge networks with deadline-constrained heterogeneous demands," 2022, *arXiv:2202.07792*.

[55] E. Che, H. D. Tuan, and H. H. Nguyen, "Joint optimization of cooperative beamforming and relay assignment in multi-user wireless relay networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 10, pp. 5481–5495, Oct. 2014.
[56] Y. LeCun, C. Cortes, and C. Burges. (2010). MNIST handwritten digit database. ATT Labs. [Online]. Available: https://yann.lecun.com/exdb/mnist
[57] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[58] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German traffic sign recognition benchmark: A multi-class classification competition," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul./Aug. 2011, pp. 1453–1460.
[59] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., Apr. 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
[60] H. Yang, X. Zhang, P. Khanduri, and J. Liu, "Anarchic federated learning," in *Proc. ICML*, 2022, pp. 25331–25363.

Md Ferdous Pervej (Graduate Student Member, IEEE) received the B.Sc. degree in electronics and telecommunication engineering from the Rajshahi University of Engineering and Technology, Rajshahi, Bangladesh, in 2014, and the M.S. degree in electrical engineering from Utah State University, Logan, UT, USA, in 2019. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, North Carolina State University, Raleigh, NC, USA.

He was with the Mitsubishi Electric Research Laboratories, Cambridge, MA, USA, in summer 2021, where he developed collaborative and privacy-preserving machine learning (ML) solutions for the Internet of Vehicles. From May 2022 to December 2022, he was with the Futurewei Wireless Research and Standards, Schaumburg, IL, USA, where he developed ML solutions for beam management for 5G new radio. His primary research interests include 5G, edge networks, vehicle-to-everything communication, edge caching/computing, ML for wireless networks, and wireless networks for distributed and on-device ML.

Richeng Jin (Member, IEEE) received the B.S. degree in information and communication engineering from Zhejiang University, Hangzhou, China, in 2015, and the Ph.D. degree in electrical engineering from North Carolina State University, Raleigh, NC, USA, in 2020.

He was a Post-Doctoral Researcher of electrical and computer engineering with North Carolina State University from 2021 to 2022. He is currently a Faculty Member with the Department of Information and Communication Engineering, Zhejiang University. His research interests include wireless AI, game theory, and security and privacy in machine learning/artificial intelligence and wireless networks.

Huaiyu Dai (Fellow, IEEE) received the B.E. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1996 and 1998, respectively, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, USA, in 2002.

He was with Bell Labs, Lucent Technologies, Holmdel, NJ, in summer 2000, and with AT&T Labs Research, Middletown, NJ, in summer 2001. He is currently a Professor of electrical and computer engineering with North Carolina State University, Raleigh, holding the title of University Faculty Scholar. His research interests include communications, signal processing, networking, and computing. His current research focuses on machine learning and artificial intelligence for communications and networking, multilayer and interdependent networks, dynamic spectrum access and sharing, as well as security and privacy issues in the above systems.

Prof. Dai is a member of the Executive Editorial Committee for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He was a co-recipient of best paper awards from the 2010 IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2010), the 2016 IEEE INFOCOM BIGSECURITY Workshop, and the 2017 IEEE International Conference on Communications (ICC 2017). He received the Qualcomm Faculty Award in 2019. He has served as an Area Editor for IEEE TRANSACTIONS ON COMMUNICATIONS and the Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING. He also serves as an Area Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.