Risk-Sensitive Task Fetching and Offloading for Vehicular Edge Computing

Sadeep Batewela, *Student Member, IEEE*, Chen-Feng Liu, *Student Member, IEEE*,
Mehdi Bennis, *Senior Member, IEEE*, Himal A. Suraweera, *Senior Member, IEEE*,
and Choong Seon Hong, *Senior Member, IEEE*

***Abstract*—This letter studies an ultra-reliable low latency communication problem focusing on a vehicular edge computing network in which vehicles either fetch and synthesize images recorded by surveillance cameras or acquire the synthesized image from an edge computing server. The notion of *risk-sensitive* in financial mathematics is leveraged to define a reliability measure, and the studied problem is formulated as a risk minimization problem for each vehicle's end-to-end (E2E) task fetching and offloading delays. Specifically, by resorting to a joint utility and policy estimation-based learning algorithm, a distributed risk-sensitive solution for task fetching and offloading is proposed. Simulation results show that our proposed solution achieves performance improvements up to 40% variance reduction and steeper distribution tail of the E2E delay over an averaged-based baseline.**

***Index Terms*—5G and beyond, vehicular edge computing, URLLC, risk-sensitive learning.**

I. Introduction

By APPLYING the mobile edge computing (MEC) architecture [1] to vehicular networks, vehicular edge computing (VEC) [2] provides computation service for vehicles to execute time-critical applications. The main problem is that the latency exacerbates when all vehicles intend to access the MEC server's computational resources. To this end, some works have studied the offloading and resource allocation mechanisms in various VEC scenarios [3]–[5]. However, the majority of the VEC literature aimed at reducing the latency, whereas reliably delivering messages for traffic safety maintenance is critical in VEC networks. To enable ultra-reliable low latency communication (URLLC), improving the delay performance on average is not sufficient [6] but requires further breaking down the statistics of delay, i.e., mean, variance, skewness, etc. To this end, we resort to the *risk* measure, a notion adopted in financial mathematics [7], for wireless communication, where risk is synonymous with losing valuable information due to the randomness of wireless transmission. For example, the stochastic channel quality incurs delay variation, and a higher variance in the delay can result in an urgent-message loss, making traffic safety at stake.

In this letter, we consider a VEC scenario in which computation tasks are remotely generated at surveillance cameras. Next, the vehicle decides either to fetch the tasks from the cameras for local computation or to obtain the computed results from the MEC server. In the latter case, tasks are offloaded from the cameras and computed at the server. Motivated by the aforementioned shortcomings, the studied problem is cast as a risk minimization problem for each vehicle's end-to-end (E2E) delay. Moreover, the variance and higher-order statistics of the E2E delay are taken into account in the risk-minimization problem. Since all vehicles' delays are coupled together in the considered system, each vehicle requires other vehicles' wireless channel information, which is not available in practice, to solve the problem. To address this issue, we leverage upon a joint utility and policy estimation-based learning framework [8], [9] and propose a distributed risk-sensitive approach for task fetching and offloading. Numerical results show that in contrast to the average-based system design, our proposed solution achieves lower variance and steeper distribution tail of the E2E delay albeit higher average performance. The superiority of the proposed approach in denser networks is validated through comparison with several baseline schemes.

II. System Model

We consider an urban VEC network around the road intersection, where a set $\mathcal{V}$ of V vehicular user equipments (VUEs) locate on the intersecting roads. A set $\mathcal{C}$ of C cameras are installed at the intersection to record images for aiding traffic safety. We assume that each camera has a limited angle of view, and all cameras' images need to be synthesized to have the full view of the intersection. Since VUEs are at different locations, we simply assume that each VUE requires a dedicated synthesized image which is different from the other VUEs' required synthesized images. The VUE has its own computational capability to synthesize the images that are

This work was supported in part by the Academy of Finland project CARMA, in part by the Academy of Finland project MISSION, in part by the Academy of Finland project SMARTER, in part by the INFOTECH project NOOR, in part by the Kvantum Institute strategic project SAFARI, and in part by the Nokia Foundation.

S. Batewela is with Nokia Networks, 33100 Tampere, Finland (e-mail: sadeep.batewela_vidanelage@nokia.com).

C.-F. Liu is with the Centre for Wireless Communications, University of Oulu, 90014 Oulu, Finland (e-mail: chen-feng.liu@oulu.fi).

M. Bennis is with the Centre for Wireless Communications, University of Oulu, 90014 Oulu, Finland, and also with the Department of Computer Science and Engineering, Kyung Hee University, Yongin 17104, South Korea (e-mail: mehdi.bennis@oulu.fi).

H. A. Suraweera is with the Department of Electrical and Electronic Engineering, University of Peradeniya, Peradeniya 20400, Sri Lanka (e-mail: himal@ee.pdn.ac.lk).

C. S. Hong is with the Department of Computer Science and Engineering, Kyung Hee University, Yongin 17104, South Korea (e-mail: cshong@khu.ac.kr).

wirelessly fetched from cameras. Moreover, due to VUEs' limited computation capabilities, an MEC server, connected with all cameras via fiber, is deployed to provide the computational service for the VUEs. In this regard, images are first delivered to the MEC server through the fiber links. After synthesizing, the MEC server wirelessly sends the synthesized images to the VUEs. Let us further denote each VUE i's task-fetching and offloading decision as $\alpha_i \in \{0,1\}, \forall\, i \in \mathcal{V}$, in which the VUE fetches images and synthesize them locally if $\alpha_i = 0$. Otherwise, synthesizing is executed at the MEC server. The task-fetching and offloading decision in the considered VEC system is schematically shown in Fig. 1.

When $\alpha_i = 0$, the VUE first fetches the images from the cameras. We assume that each camera has a dedicated bandwidth $W_{\rm c}$ to broadcast its image to all task-fetching VUEs, denoted by $\mathcal{V}_{\rm f} = \{i \in \mathcal{V} |\, \alpha_i = 0\}$. In order to ensure that all VUEs in $\mathcal{V}_{\rm f}$ can correctly receive the same image over the full bandwidth $W_{\rm c}$, each camera $j \in \mathcal{C}$ broadcasts its data with the rate $R_j = W_{\rm c} \log_2\big(1 + \frac{P_{\rm c} h_{j,\min}}{W_{\rm c} N_0}\big)$, considering the minimal channel gain among all corresponding camera-VUE links, i.e., $h_{j,\min} = \min\limits_{i \in \mathcal{V}_{\rm f}} h_{ji}$. Here, h_{ji} is the channel gain, including path loss and channel fading, between camera j and VUE i. We further assume that all wireless channel gains belong to the finite sets such that channels stay static[1] during task fetching and offloading. $P_{\rm c}$ is the camera's transmit power which is identical for all cameras, and N_0 is the power spectral density of the additive white Gaussian noise. Then given the camera's image size A, the transmission delay from camera j is $T_j^{\rm Tx} = A/R_j$. Since the VUE i has to wait till it receives all cameras' images before synthesizing, the net transmission delay for each VUE is $T^{\rm Fet} = \max\limits_{j \in \mathcal{C}} T_j^{\rm Tx}$. Here, we note that the transmission delay $T^{\rm Fet}$ is affected by all the task-fetching VUEs' wireless channels as per $h_{j,\min} = \min\limits_{i \in \mathcal{V}_{\rm f}} h_{ji}$. Assuming that all camera's images are in the same size, we calculate the computation delay for image synthesizing at VUE i as $T_i^{\rm Comp} = CAL/f_i$ in which L and f_i (in the unit of cycle/sec) are, respectively, the required central processing unit (CPU) cycles per bit for computation, i.e., the processing density, and VUE i's local CPU-cycle frequency [1]. When $\alpha_i = 0$, VUE i's E2E delay $T_i^{\rm E2E}$ includes the transmission delay and local computation delay, i.e., $T_i^{\rm E2E} = T^{\rm Fet} + T_i^{\rm Comp}$.

For $\alpha_i = 1$, let us first analogously denote the set of VUEs with the task-offloading decision as $\mathcal{V}_{\rm o} = \{i \in \mathcal{V} |\, \alpha_i = 1\}$. Additionally, since all cameras deliver their images to the MEC server through fiber links, we neglect the transmission delays from cameras to the MEC server. After obtaining these images, the MEC server equally divides its total CPU-cycle frequency $f_{\rm s}$ into $|\mathcal{V}_{\rm o}|$ parts in order to simultaneously synthesize the $|\mathcal{V}_{\rm o}|$ VUEs' dedicated images. Thus, the computational delay at the MEC server is expressed as $T_{\rm s}^{\rm Comp} = CAL|\mathcal{V}_{\rm o}|/f_{\rm s}$. Subsequently, the MEC server orthogonally sends the synthesized image to each VUE $i \in \mathcal{V}_{\rm o}$ with the equally allocated bandwidth $W_{\rm s}/|\mathcal{V}_{\rm o}|$. The allocated transmit power for VUE i is denoted by P_i with the constraint

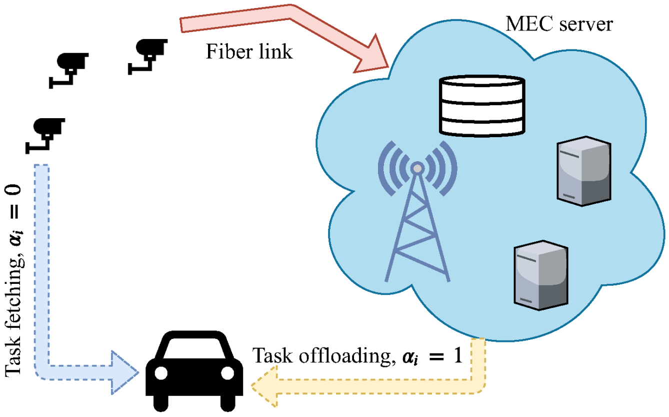

MEC server
Fiber link
Task fetching, $\alpha_i = 0$
Task offloading, $\alpha_i = 1$

Fig. 1. Schematic illustration of the task-fetching and offloading decision in a vehicular setting.

$$\sum_{i \in \mathcal{V}_{\rm o}} P_i = P_{\max} \text{ and } P_i \geq 0,\ \forall\, i \in \mathcal{V}_{\rm o}, \tag{1}$$

where $P_{\max}$ is the MEC server's transmit power budget. The power allocation mechanism will be detailed in Section III-B. Accordingly, the corresponding downlink (DL) rate (from the MEC server) to the VUE i is expressed as $R_i = \frac{W_{\rm s}}{|\mathcal{V}_{\rm o}|} \log_2\big(1 + \frac{P_i h_{{\rm s}i} |\mathcal{V}_{\rm o}|}{W_{\rm s} N_0}\big)$, where $h_{{\rm s}i}$ is the DL channel gain from the MEC server to VUE i. Then given the synthesized image size B, the DL transmission delay of the VUE i is $T_i^{\rm DL} = B/R_i$. When $\alpha_i = 1$, VUE i's E2E delay includes the computation delay at the MEC server and the DL transmission delay, i.e., $T_i^{\rm E2E} = T_{\rm s}^{\rm Comp} + T_i^{\rm DL}$. In summary, VUE i's E2E latency is expressed as

$$T_i^{\rm E2E} = \begin{cases} \max\limits_{j \in \mathcal{C}} \left\{ \dfrac{A}{W_{\rm c} \log_2\Big(1 + \frac{P_{\rm c} h_{j,\min}}{W_{\rm c} N_0}\Big)} \right\} + \dfrac{CAL}{f_i}, \\ \hfill \text{when } \alpha_i = 0, \\ \dfrac{CAL|\mathcal{V}_{\rm o}|}{f_{\rm s}} + \dfrac{B|\mathcal{V}_{\rm o}|}{W_{\rm s} \log_2\Big(1 + \frac{P_i h_{{\rm s}i} |\mathcal{V}_{\rm o}|}{W_{\rm s} N_0}\Big)}, \\ \hfill \text{when } \alpha_i = 1. \end{cases} \tag{2}$$

Furthermore, we note that the E2E delay in (2) is random.

III. Problem Formulation

A. Risk Minimization for the VUE's End-to-End Delay

As mentioned in Section I, risk is considered as our concerned reliability metric. In this regard, we focus on the entropic risk measure[2] $\frac{1}{\rho} \ln(\mathbb{E}[\exp(\rho T_i^{\rm E2E})])$ and formulate a risk minimization problem for each VUE $i \in \mathcal{V}$ as follows:

$$\underset{\Pr(\alpha_i|\mathbf{h}_i) \geq 0}{\text{minimize}} \quad \frac{1}{\rho} \ln(\mathbb{E}[\exp(\rho T_i^{\rm E2E})]) \tag{3a}$$

$$\text{subject to} \quad \sum_{\alpha_i \in \{0,1\}} \Pr(\alpha_i|\mathbf{h}_i) = 1,\ \ \forall\, \mathbf{h}_i \in \mathcal{H}_i, \tag{3b}$$

where $\mathbb{E}[\cdot]$ is the expectation operator, and $\mathbf{h}_i = [h_{ji}, h_{{\rm s}i} : j \in \mathcal{C}] \in \mathcal{H}_i, \forall\, i \in \mathcal{V}$, denotes VUE i's channel vector with a finite set $\mathcal{H}_i$. By taking the Maclaurin series

[1] For example, by quantizing the continuous channel fading values in consecutive coherence time blocks into the same level such that the corresponding transmission rate is sustainable.

[2] Because entropic risk measure directly incorporates the statistics of the delay, i.e., mean, variance, skewness, etc.

expansion, we can express the objective function (3a) as $\frac{1}{\rho}\ln(\mathbb{E}[\exp(\rho T_i^{\mathrm{E2E}})]) = \mathbb{E}[T_i^{\mathrm{E2E}}] + \frac{\rho}{2!}\mathrm{Var}(T_i^{\mathrm{E2E}}) + \frac{\rho^2\mu_3}{3!} + \cdots$ in which $\mu_3 = \mathbb{E}[(T_i^{\mathrm{E2E}} - \mathbb{E}[T_i^{\mathrm{E2E}}])^3]$ is the third central moment of T_i^{E2E}, and the skewness of T_i^{E2E} is equal to $\frac{\mu_3}{(\mathrm{Var}(T_i^{\mathrm{E2E}}))^{3/2}}$. In other words, the mean, variance, skewness, and other higher-order statistics of the E2E delay are taken into account in the objective. Moreover, the risk-sensitivity parameter $\rho > 0$ reflects the weight of higher-order statistics in the formulated risk minimization problem. Owing to the randomness of $\mathbf{h}_i$ and other unavailable information, e.g., the other VUEs' channels, each VUE i focuses on the probabilistic policy $\Pr(\alpha_i|\mathbf{h}_i)$ of the task-fetching and offloading decision in problem (3). Since $\frac{1}{\rho}\ln(\cdot)$ in (3a) is a monotonically increasing function, we can directly remove it and focus on an equivalent problem

$$\underset{\Pr(\alpha_i|\mathbf{h}_i)\geq 0}{\text{minimize}} \ \mathbb{E}[\exp(\rho T_i^{\mathrm{E2E}})] \ \text{ subject to (3b).} \tag{4}$$

Analogously, by taking the Maclaurin series expansion, i.e., $\mathbb{E}[\exp(\rho T_i^{\mathrm{E2E}})] = 1 + \rho\mathbb{E}[T_i^{\mathrm{E2E}}] + \frac{\rho^2}{2!}\mathbb{E}[(T_i^{\mathrm{E2E}})^2] + \frac{\rho^3}{3!}\mathbb{E}[(T_i^{\mathrm{E2E}})^3] + \cdots$, we can see that the higher-order moments/statistics are still considered in the objective function of the equivalent problem.

B. Transmit Power Allocation at the MEC Server

Referring to the motivation of considering (4), we formulate the MEC server's power allocation problem as

$$\underset{P_i}{\text{minimize}} \ \sum_{i\in\mathcal{V}_{\mathrm{o}}} \exp(\rho T_i^{\mathrm{DL}}) \ \text{ subject to (1).} \tag{5}$$

In the objective, we consider the DL transmission delay since the allocated transmit power only affects this delay. Following the results in [10, Eq. (3.10)], we can verify that $\exp(\rho T_i^{\mathrm{DL}}) = \exp\left(\frac{\rho B|\mathcal{V}_{\mathrm{o}}|}{W_{\mathrm{s}}\log_2\left(1+\frac{P_i h_{\mathrm{s}i}|\mathcal{V}_{\mathrm{o}}|}{W_{\mathrm{s}}N_0}\right)}\right)$ is a convex function with respect to P_i. Hence, we directly apply the Karush–Kuhn–Tucker (KKT) conditions and derive the following optimal solution. For the DL transmission, the MEC server allocates the transmit power $P_i^* > 0, \forall\, i \in \mathcal{V}_{\mathrm{o}}$, which satisfies $\frac{\theta\kappa_i\exp\left(\frac{\theta}{\ln(1+P_i^*\kappa_i)}\right)}{(1+P_i^*\kappa_i)[\ln(1+P_i^*\kappa_i)]^2} = \nu$ with $\theta = \frac{\rho B|\mathcal{V}_{\mathrm{o}}|\ln 2}{W_{\mathrm{s}}}$ and $\kappa_i = \frac{h_{\mathrm{s}i}|\mathcal{V}_{\mathrm{o}}|}{W_{\mathrm{s}}N_0}$. Here, ν is chosen such that $\sum_{i\in\mathcal{V}_{\mathrm{o}}} P_i^* = P_{\max}$.

IV. Distributed Risk-Sensitive Approach for Task Fetching and Offloading

Note that each VUE i's objective $\mathbb{E}[\exp(\rho T_i^{\mathrm{E2E}})]$ in problem (4) is affected by not only its own policy of the task fetching and offloading decision but also the other VUEs' as per (1) and (2). Hence, the studied risk minimization for all VUEs is a multi-agent problem, where the stable[3] policy of (4) is desired at each VUE. To solve (4), each VUE i requires the full information of $|\mathcal{V}_{\mathrm{o}}|$, $\mathcal{V}_{\mathrm{f}}$, and $\mathbf{h}_{i'}, \forall\, i' \in \mathcal{V}_{\mathrm{f}} \setminus i$, as per (2). In other words, the VUE needs to know all possible values of $\exp(\rho T_i^{\mathrm{E2E}})$ in the objective function. However, the required information is cumbersome to fetch in practice. To address this issue while achieving the stable policy, we consider a joint utility and policy estimation-based learning framework, i.e., the distributed no-regret learning algorithm [9], which iteratively estimates the average impact of the aforementioned unavailable information on $\exp(\rho T_i^{\mathrm{E2E}})$ and further obtain a task-fetching and offloading solution to problem (4). The main steps of the distributed no-regret learning algorithm are outlined as follows:

- Each VUE i is given an initial policy $\Pr(\alpha_i|\mathbf{h}_i)$.
- In each iteration t, VUE i observes the independent and identically distributed (i.i.d.) realization $\mathbf{h}_i(t)$ and then makes a decision $\alpha_i(t)$ based on its policy $\Pr(\alpha_i|\mathbf{h}_i)$.
- Subsequently, cameras and the MEC server are implicitly informed about the VUE compositions of $\mathcal{V}_{\mathrm{f}}(t)$ and $\mathcal{V}_{\mathrm{o}}(t)$, respectively, and send the images to the corresponding VUEs. Once the synthesized image is available, each VUE i can observe its E2E delay $T_i^{\mathrm{E2E}}(t)$ and calculate the utility $-\exp(\rho T_i^{\mathrm{E2E}}(t))$.
- Each VUE calculates the average utilities over all past iterations by 1) always making the same decision α_i; and 2) following its iteration-variant policy $\Pr(\alpha_i|\mathbf{h}_i)$. The *regret* of not insisting on a specific decision is the difference between the average utilities of 1) and 2).
- VUE i iteratively updates its policy such that the likelihood of making a specific decision is proportional to the corresponding regret calculated over the past iterations.

Let us explain the details in the remainder of this section. Firstly, we denote all unavailable information of VUE i as $\boldsymbol{\omega}_i$. Since the objective function of problem (4) is affected by α_i, $\mathbf{h}_i$, and $\boldsymbol{\omega}_i$, we define VUE i's utility in the t-th iteration as $u_i(t) = u_i(\alpha_i(t), \mathbf{h}_i(t), \boldsymbol{\omega}_i(t)) = -\exp(\rho T_i^{\mathrm{E2E}}(t))$. Additionally, for notational clarity, we use α_i^m and $\mathbf{h}_i^l$ to express the specific realizations of $\alpha_i(t)$ and $\mathbf{h}_i(t)$, respectively. The *state* realizations of $\mathbf{h}_i(t)$ in all iterations are i.i.d., whereas the *action* realization of $\alpha_i(t)$ is generated based on some iteration-variant distribution $\Pr(\alpha_i|\mathbf{h}_i) = \pi_{\mathbf{h}_i}(t-1;\alpha_i)$. Subsequently, considering the state realization $\mathbf{h}_i^l$, the conditional *regret* of the action α_i^m in iteration t is defined as

$$r_{\mathbf{h}_i^l}(t;\alpha_i^m) = \frac{1}{\sum_{\tau=1}^{t}\mathbb{1}_{\{\mathbf{h}_i(\tau)=\mathbf{h}_i^l\}}} \times \sum_{\tau=1}^{t}\left[u_i(\alpha_i^m, \mathbf{h}_i^l, \boldsymbol{\omega}_i(\tau)) - u_i(\tau)\right] \times \mathbb{1}_{\{\mathbf{h}_i(\tau)=\mathbf{h}_i^l\}}, \tag{6}$$

where $\mathbb{1}_{\{\cdot\}}$ represents the indicator function. In (6), we calculate the average utility over the past iteration instants with $\mathbf{h}_i(\tau) = \mathbf{h}_i^l$ and constantly changing $\alpha_i(\tau)$, and the average utility of fixing the specific action α_i^m over these instants. The conditional regret $r_{\mathbf{h}_i^l}(t;\alpha_i^m)$ is interpreted as the average utility improvement by fixing the decision α_i^m in the iteration instants with $\mathbf{h}_i(\tau) = \mathbf{h}_i^l$. However, since the realization of $\boldsymbol{\omega}_i(\tau)$ is unknown, it is not feasible to calculate $u_i(\alpha_i^m, \mathbf{h}_i^l, \boldsymbol{\omega}_i(\tau))$ and the conditional regret (6). To deal with this issue, the VUE estimates the expected utility with respect to $\boldsymbol{\omega}_i$ given α_i^m and $\mathbf{h}_i^l$, i.e., $\mathbb{E}_{\boldsymbol{\omega}_i}[u_i(\alpha_i^m, \mathbf{h}_i^l)]$. To this end, the expected utility $\mathbb{E}_{\boldsymbol{\omega}_i}[u_i(\alpha_i^m, \mathbf{h}_i^l)]$ and conditional regret

[3] While all the other VUEs follow their stable policies, the VUE deviating from the stable policy cannot further decrease its objective $\mathbb{E}[\exp(\rho T_i^{\mathrm{E2E}})]$.

Algorithm 1 Distributed No-Regret Learning Algorithm

```
1: Initialize $t = 1$ and set initial values for $\hat{u}_{\mathbf{h}_i^l}(0;\alpha_i^m)$, $\hat{r}_{\mathbf{h}_i^l}(0;\alpha_i^m)$, $\pi_{\mathbf{h}_i^l}(0;\alpha_i^m)$, $\forall\, i \in \mathcal{V}, \alpha_i^m \in \{0,1\}, \mathbf{h}_i^l \in \mathcal{H}_i$.
2: repeat
3:     Observing a realization $\mathbf{h}_i(t) = \mathbf{h}_i^l$, each VUE $i \in \mathcal{V}$ makes a decision $\alpha_i(t) = \alpha_i^m$ based on its $\Pr(\alpha_i(t) = \alpha_i^m | \mathbf{h}_i(t) = \mathbf{h}_i^l) = \pi_{\mathbf{h}_i^l}(t-1;\alpha_i^m)$.
4:     The MEC server allocates the transmit power for the VUE set $\mathcal{V}_{\mathrm{o}}(t)$ as per Section III-B.
5:     Each VUE $i$ observes its $T_i^{\mathrm{E2E}}(t)$ and locally updates (7)–(10).
6:     $t \leftarrow t + 1$.
7: until Stopping criteria are satisfied.
```

$r_{\mathbf{h}_i^l}(t;\alpha_i^m)$ are estimated, in iteration t, as per

$$\hat{u}_{\mathbf{h}_i^l}(t;\alpha_i^m) = \hat{u}_{\mathbf{h}_i^l}(t-1;\alpha_i^m) + \eta_u(t) \times \mathbb{1}_{\{\mathbf{h}_i(t)=\mathbf{h}_i^l\}} \times \mathbb{1}_{\{\alpha_i(t)=\alpha_i^m\}} \times \big[u(t) - \hat{u}_{\mathbf{h}_i^l}(t-1;\alpha_i^m)\big], \quad (7)$$

$$\hat{r}_{\mathbf{h}_i^l}(t;\alpha_i^m) = \hat{r}_{\mathbf{h}_i^l}(t-1;\alpha_i^m) + \eta_r(t) \times \mathbb{1}_{\{\mathbf{h}_i(t)=\mathbf{h}_i^l\}} \times \big[\hat{u}_{\mathbf{h}_i^l}(t;\alpha_i^m) - u(t) - \hat{r}_{\mathbf{h}_i^l}(t-1;\alpha_i^m)\big], \quad (8)$$

$\forall\, \alpha_i^m \in \{0,1\}, \mathbf{h}_i^l \in \mathcal{H}_i$. Given all the estimated conditional regrets $\hat{r}_{\mathbf{h}_i^l}(t;\alpha_i^m)$ in the current iteration t, a straightforward solution to the task-fetching and offloading problem is making the decision (for each channel vector value $\mathbf{h}_i^l$) with the maximal regret. On the other hand, we also want to comprehensively explore the impacts of the randomness $\boldsymbol{\omega}_i$ on both $u_i(\alpha_i = 0, \mathbf{h}_i^l)$ and $u_i(\alpha_i = 1, \mathbf{h}_i^l)$. Incorporating these two concerns, the conditional probability distribution for the task-fetching and offloading decision in iteration t can be modeled as [9], $\forall\, \mathbf{h}_i^l \in \mathcal{H}_i$, $\beta(\hat{\mathbf{r}}_{\mathbf{h}_i^l}(t);\alpha_i) = \underset{\Pr(\alpha_i|\mathbf{h}_i^l)}{\arg\max}\Big\{\sum_{\alpha_i^m=0}^{1}\big[\Pr(\alpha_i^m|\mathbf{h}_i^l)\hat{r}_{\mathbf{h}_i^l}(t;\alpha_i^m) - \frac{1}{\xi}\Pr(\alpha_i^m|\mathbf{h}_i^l)\ln\big(\Pr(\alpha_i^m|\mathbf{h}_i^l)\big)\big]\Big\}$ whose solution is given by

$$\beta(\hat{\mathbf{r}}_{\mathbf{h}_i^l}(t);\alpha_i^m) = \frac{\exp\big(\xi \hat{r}^{+}_{\mathbf{h}_i^l}(t;\alpha_i^m)\big)}{\sum_{\tilde{\alpha}_i^m=0}^{1}\exp\big(\xi \hat{r}^{+}_{\mathbf{h}_i^l}(t;\tilde{\alpha}_i^m)\big)}, \quad (9)$$

$\forall\, \alpha_i^m \in \{0,1\}$, with $\hat{\mathbf{r}}_{\mathbf{h}_i^l}(t) = [\hat{r}_{\mathbf{h}_i^l}(t;0), \hat{r}_{\mathbf{h}_i^l}(t;1)]$ and $\hat{r}^{+}_{\mathbf{h}_i^l}(t;\alpha_i^m) = \max\{\hat{r}_{\mathbf{h}_i^l}(t;\alpha_i^m), 0\}$. Here, $\xi > 0$ is the parameter which trades off exploitation (maximizing the expected conditional regret) and exploration (maximizing information entropy). Then combining the estimation rules in (7) and (8) with the distribution (9), we update the task-fetching and offloading policy for VUE i as

$$\pi_{\mathbf{h}_i^l}(t;\alpha_i^m) = \pi_{\mathbf{h}_i^l}(t-1;\alpha_i^m) + \eta_\pi(t) \times \mathbb{1}_{\{\mathbf{h}_i(t)=\mathbf{h}_i^l\}} \times \big[\beta(\hat{\mathbf{r}}_{\mathbf{h}_i^l}(t);\alpha_i^m) - \pi_{\mathbf{h}_i^l}(t-1;\alpha_i^m)\big], \quad (10)$$

$\forall\, \alpha_i^m \in \{0,1\}, \mathbf{h}_i^l \in \mathcal{H}_i$. Additionally, in (7), (8), and (10), the learning rates $\eta_u(t)$, $\eta_r(t)$, and $\eta_\pi(t)$, which satisfy $\lim\limits_{N\to\infty}\sum_{t=1}^{N}\eta_u(t) = \infty$, $\lim\limits_{N\to\infty}\sum_{t=1}^{N}\eta_r(t) = \infty$, $\lim\limits_{N\to\infty}\sum_{t=1}^{N}\eta_\pi(t) = \infty$, $\lim\limits_{N\to\infty}\sum_{t=1}^{N}[\eta_u(t)]^2 < \infty$, $\lim\limits_{N\to\infty}\sum_{t=1}^{N}[\eta_r(t)]^2 < \infty$, $\lim\limits_{N\to\infty}\sum_{t=1}^{N}[\eta_\pi(t)]^2 < \infty$, $\lim\limits_{t\to\infty}\frac{\eta_r(t)}{\eta_u(t)} = 0$, and $\lim\limits_{t\to\infty}\frac{\eta_\pi(t)}{\eta_r(t)} = 0$, can be chosen by referring to p-series. In the next iteration $t+1$, the action realization of $\alpha_i(t+1)$ is generated based on $\Pr(\alpha_i(t+1) = \alpha_i^m|\mathbf{h}_i(t+1) = \mathbf{h}_i^l) = \pi_{\mathbf{h}_i^l}(t;\alpha_i^m)$. Finally, the converged distribution $\pi_{\mathbf{h}_i^l}(\infty;\alpha_i^m)$ provides a stable solution to the studied problem (4). The related proof of stability can be found in [9]. The steps of the distributed no-regret learning algorithm are outlined in Algorithm 1. Moreover, given that $\mathbb{E}_{\boldsymbol{\omega}_i}[\exp(\rho T_i^{\mathrm{E2E}})|\alpha_i, \mathbf{h}_i], \forall\, \alpha_i, \mathbf{h}_i,$ is known, problem (4) is a linear programming problem which includes $2|\mathcal{H}_i|$ optimization variables and $3|\mathcal{H}_i|$ constraints. Here, $|\mathcal{H}_i|$ increases exponentially with the number $|\mathcal{C}|+1$. In Algorithm 1 which addresses the unavailability dilemma of $\mathbb{E}_{\boldsymbol{\omega}_i}[\exp(\rho T_i^{\mathrm{E2E}})|\alpha_i, \mathbf{h}_i]$, $6|\mathcal{H}_i|$ memory elements (for calculating (7), (8), and (10)) are required and updated iteratively.

V. Numerical Results

We consider Rayleigh fading with unit variance and the path loss model $68.5 + 16.1\log_{10} d$ (dB) at the 5.9 GHz carrier frequency [11], where $d \in [1, 100]$ measured in meters is the distance between the camera/MEC server and vehicle. The channel gain is further quantized into two levels. There are four cameras in the simulated network. Considering that the MEC server must have sufficient resources to serve a large number of VUEs, we select $W_{\mathrm{s}} = 20\,\mathrm{MHz}$, $W_{\mathrm{c}} = 100\,\mathrm{kHz}$, $f_{\mathrm{s}} = 2 \times 10^{11}\,$cycle/s, and $f_i = 10^9\,$cycle/s by referring to [11], [12]. To synthesize images, the same objects in different images need to be recognized first. Therefore, we refer to the processing density for face recognition, i.e., $L = 2339\,$cycle/bit [13], in the simulations. Some parameters will be specified while investigating their impacts on performance. The rest simulation parameters are $A = 20\,$kbit, $B = 60\,$kbit, $N_0 = -174\,$dBm/Hz, $P_{\mathrm{c}} = 20\,$dBm, $P_{\max} = 30\,$dBm, $\xi = 10$, $\eta_u(t) = t^{-0.51}$, $\eta_r(t) = t^{-0.52}$, and $\eta_\pi(t) = t^{-0.53}$. We run 10000 iterations in Algorithm 1. For performance comparison, we consider the following four baselines:

1) **Average-based** scheme in which $\mathbb{E}[T_i^{\mathrm{E2E}}]$, $\sum_{i\in\mathcal{V}_{\mathrm{o}}} T_i^{\mathrm{DL}}$, and $u_i(t) = -T_i^{\mathrm{E2E}}(t)$ are considered in (4), (5), and (7), respectively;
2) **Fully-fetching** scheme with $\Pr(\alpha_i = 0|\mathbf{h}_i) = 1, \forall\, i \in \mathcal{V}, \mathbf{h}_i \in \mathcal{H}_i$;
3) **Fully-offloading** scheme with $\Pr(\alpha_i = 1|\mathbf{h}_i) = 1, \forall\, i \in \mathcal{V}, \mathbf{h}_i \in \mathcal{H}_i$ and $\sum_{i\in\mathcal{V}}\exp(\rho T_i^{\mathrm{DL}})$ in (5);
4) **Half-fetching half-offloading** scheme with $\Pr(\alpha_i = 0|\mathbf{h}_i) = \Pr(\alpha_i = 1|\mathbf{h}_i) = 0.5, \forall\, i \in \mathcal{V}, \mathbf{h}_i \in \mathcal{H}_i$ and $\sum_{i\in\mathcal{V}_{\mathrm{o}}}\exp(\rho T_i^{\mathrm{DL}})$ in (5).

In Fig. 2, we show the mean and standard deviation of the E2E delay by varying the risk-sensitive parameter ρ. As per the Maclaurin series expansions of the objective functions in (3a) and (4), the goal is to lower the variance and higher-order statistics which increase with ρ. Consequently, the standard deviation (or the variance) of the E2E delay is a decreasing function of ρ, whereas the average E2E delay increases with ρ. Now let us further look into the the complementary cumulative distribution function (CCDF) of the E2E delay in Fig. 3, where we show the CCDFs of the E2E delay of both the proposed and **average-based** approaches. As shown in the

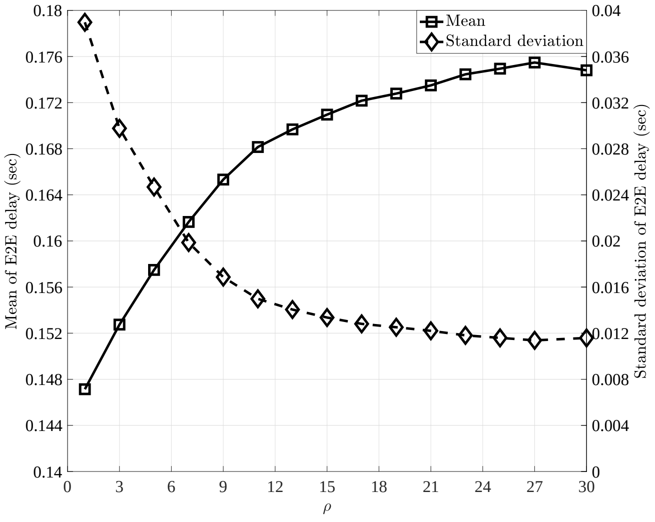


Fig. 2. Mean and standard deviation of the VUE's E2E delay versus ρ with $V = 60$ VUEs.

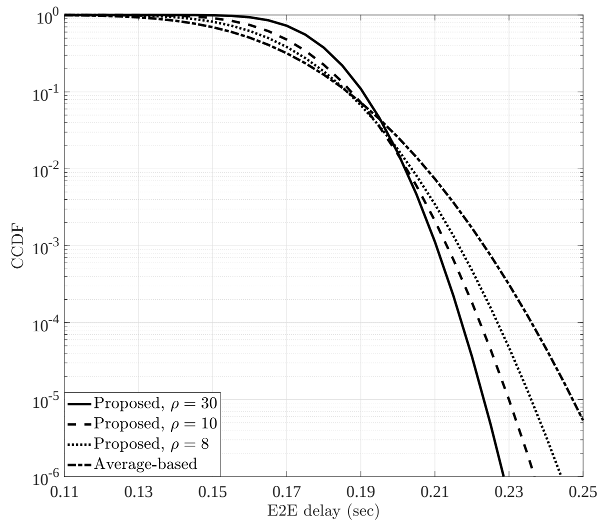
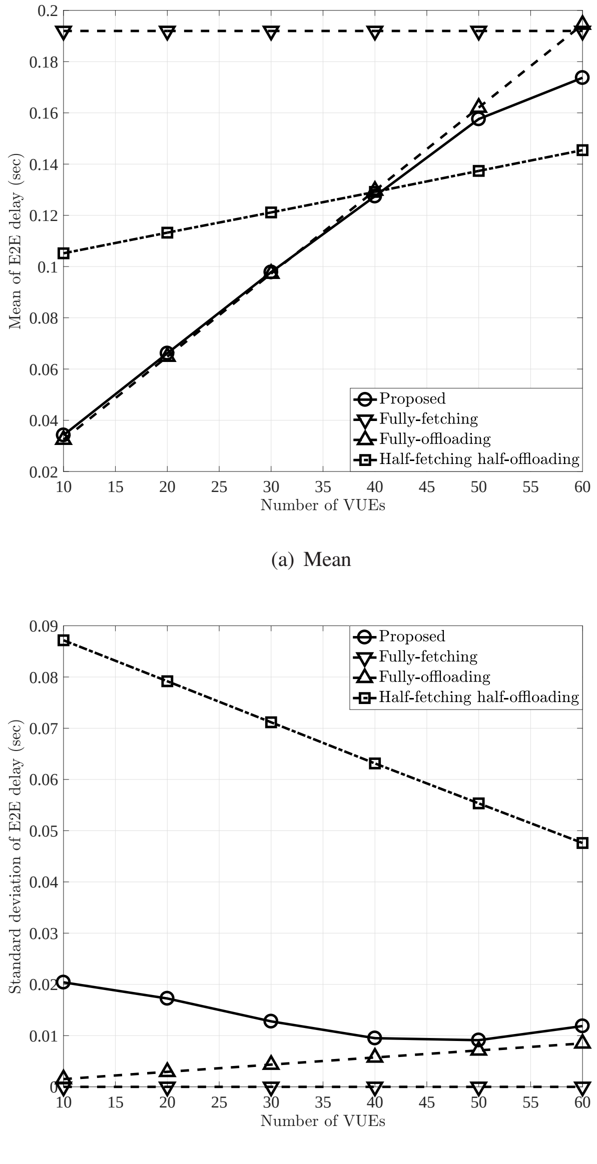


Fig. 3. CCDFs of the VUE's E2E delay for various ρ in our proposed approach and the average-based baseline with $V = 60$ VUEs.



(a) Mean

(b) Standard deviation

Fig. 4. Mean and standard deviation of the VUE's E2E delay in our proposed approach with $\rho = 30$ and the three baselines versus network density.

figure, the CCDF of the proposed approach decays faster resulting in a steeper tail, particularly with larger ρ. This behavior means that our approach exhibits lower occurrence probability of extremely high delay, compared to the **average-based** baseline. In addition, focusing on the region $[10^{-2}, 1]$ of the CCDF, we can see that the proposed approach achieves a lower variance at the expense of higher average performance.

Considering different network densities in Fig. 4, we compare the mean and standard deviation of the VUE's E2E delay of our proposed approach and the three baselines. Since the camera transmits at its lowest rate, and the VUE waits for all cameras' images before synthesizing, the E2E delay in the **fully-fetching** scheme hardly changes, i.e., almost zero variance. However, the average E2E delay in this baseline is the highest due to the camera's lowest transmission rate and the VUE's weak computation capability. When the VUE density is low, more communication and computational resources of the MEC servers are allocated to each VUE, making the **fully-offloading** baseline achieve the lowest average E2E delay. In the low-density regime, our approach gives a solution which is almost equivalent to the **fully-offloading** scheme with the best average performance. In addition, due to the non-zero probability of the task-fetching decision, i.e., $\Pr(\alpha_i = 0|\mathbf{h}_i) \neq 0$, as per (9), the delay variance of our proposed approach is slightly higher than the variance of the **fully-offloading** baseline. When the network becomes dense, less resources will be allocated from the MEC server. In contrast with the **fully-offloading** baseline, our approach increases the probability of the task-fetching decision $\Pr(\alpha_i = 0|\mathbf{h}_i)$ so that the average

E2E delay is improved. The superior mean performance of our approach is rather significant compared with the standard deviation degradation. Finally, due to equal likelihood for both local computation and task offloading, the VUE in the **half-fetching half-offloading** scheme has the highest delay variance regardless of the network density.

VI. Conclusions

In this letter, we have studied the task-fetching and offloading problem in VEC networks towards URLLC. The studied problem was formulated as a risk minimization problem per VUE's E2E delay in which the entropic risk measure is defined as our reliability metric. Leveraging a distributed no-regret learning algorithm, we have proposed a risk-sensitive task-fetching and offloading solution. Numerical results have shown a reduction in delay variance compared to the average-based approach and other baseline schemes.

References

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[2] Z. Ning, X. Wang, and J. Huang, "Mobile edge computing-enabled 5G vehicular networks: Toward the integration of communication and computing," *IEEE Veh. Technol. Mag.*, vol. 14, no. 1, pp. 54–61, Mar. 2019.

[3] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *Proc. IEEE Int. Conf. Commun.*, May 2017, pp. 1–6.

[4] C. Xu, Y. Wang, Z. Zhou, B. Gu, V. Frascolla, and S. Mumtaz, "A low-latency and massive-connectivity vehicular fog computing framework for 5G," in *Proc. IEEE Global Commun. Conf. Workshops*, Dec. 2018, pp. 1–6.

[5] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4377–4387, Jun. 2019.

[6] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proc. IEEE*, vol. 106, no. 10, pp. 1834–1853, Oct. 2018.

[7] H. Föllmer and A. Schied, *Stochastic Finance: An Introduction in Discrete Time*, 4th ed. Berlin, Germany: Walter de Gruyter, 2016.

[8] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3072–3108, 4th Quart., 2019.

[9] S. Samarakoon, M. Bennis, W. Saad, and M. Latva-aho, "Backhaul-aware interference management in the uplink of wireless small cell networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 11, pp. 5813–5825, Nov. 2013.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[11] C.-F. Liu and M. Bennis, "Ultra-reliable and low-latency vehicular transmission: An extreme value theory approach," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1292–1295, Jun. 2018.

[12] A. Samanta, Z. Chang, and Z. Han, "Latency-oblivious distributed task scheduling for mobile edge computing," in *Proc. IEEE Global Commun. Conf.*, Dec. 2018, pp. 1–7.

[13] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2510–2523, Dec. 2015.