



QoS Violation Probability Minimization in Federating Vehicular-Fogs With Cloud and Edge Systems

Binayak Kar, *Member, IEEE*, Kuan-Min Shieh, Yuan-Cheng Lai , Ying-Dar Lin , *Fellow, IEEE*, and Huei-Wen Ferng , *Senior Member, IEEE*

Abstract—Due to the fact that most user equipments (UEs) do not have enough computation power, time-sensitive tasks should be offloaded to other computation resources. In such cases, a cloud is an ideal destination to which computation tasks can be offloaded because of its huge data processing power. However, the distance between a UE and the cloud may increase the latency of data transmission. At the same time, edges and vehicular-fogs (consisting of electric vehicles with computing resources) are closer to UEs and mostly remain under-utilized. To address this issue, a federated architecture with UEs, vehicular-fogs (parking lot fog and traffic intersection fog), edges, and cloud is proposed along with a probabilistic offloading strategy. The quality of service (QoS) violation probability can be minimized by finding the optimal offloading probabilities while satisfying the delay constraint through an optimal offloading probability estimation algorithm based on the subgradient method proposed in this paper. Our results show that the QoS violation probability and the average waiting time can be decreased by 10%-12% and 45%, respectively, for a federated architecture with two vehicular-fogs as compared to an architecture without any fogs. As offloading to edges and cloud has longer communication delays than offloading to vehicular-fogs, the offloading probabilities to the edges in the architecture with two vehicular-fogs in the case of heavy traffic are reduced by nearly 35% as compared to the architecture without any fogs.

Index Terms—Federation, offloading, vehicular-fog, QoS violation probability.

I. INTRODUCTION

IN RECENT years, the intelligent transportation system (ITS) [1] has been developed to improve transportation safety and efficiency. Some applications, such as road safety applications,

have stringent requirements for data processing, including low data transmission latency and low task processing times. However, UEs and electronic vehicles do in general not have enough processing capability to meet those requirements. To address this issue, alternative ways to process tasks efficiently need to be sought with some urgency. Mobile cloud computing (MCC) [2], [3] may be a solution to address this issue, by moving data processing tasks and storage from UEs to the clouds, with the result that processing times become reduced because of the high computing capacity of cloud servers. However, the long-distances between UEs and the cloud may cause an extra data transmission delay. MCC may therefore not be a good choice to process time-sensitive tasks requested by some sensitive applications. On the contrary, mobile edge computing (MEC) [4], [5] may be a better solution for achieving the demand from these applications as they are closer to UEs. UEs can then offload their tasks to a MEC server directly without spending too much time on data transmission to the clouds. In spite of being computation resources, these MECs have certain limitations. Some requests that cannot be handled by the edge can be offloaded to the cloud when they require high computing resources.

Vehicular-fog computing (VFC) [6]–[8] is an emerging technology that turns (electronic) vehicles equipped with computing devices into fog nodes that act as small-scale cloud platforms for vehicles themselves as well as for other connected devices. These vehicles are used not only for transportation but also as a part of communication and computation infrastructure [9], [10]. For example, road safety applications rely on the information shared by vehicles. As the vehicles have computing resources, they can provide services to others and benefit financially by utilizing their available under-utilized resources. There are cases where a vehicle has computation limitations for handling a relatively large task. In such a scenario, a vehicle can collaborate with other vehicles. This collaboration by vehicles is called vehicular-fog [6]. Considering these fogs' dynamic nature, we separate vehicular-fogs into two categories: parking lot fog, where a vehicle can remain for a longer time, and traffic intersection fog, where a vehicle can remain for a maximum of 100–120 seconds.

To gain the benefits from MEC, MCC, and VFC and to utilize the advantages of a federation, a cloud, edges, and vehicular-fogs federated architecture is proposed in this paper. In such a federated architecture, the computation-intensive tasks, i.e., high computing resource tasks generated by UEs, can be offloaded to a MCC server to decrease task processing times,

Manuscript received June 22, 2021; revised September 13, 2021 and October 10, 2021; accepted October 12, 2021. Date of publication October 15, 2021; date of current version December 17, 2021. This work was supported by the Ministry of Science and Technology, Taiwan, under Grants 109-2221-E-011-104-MY3 and 109-2221-E-011-118-MY2. The review of this article was coordinated by Prof. K. Bian. (*Corresponding author: Huei-Wen Ferng.*)

Binayak Kar, Kuan-Min Shieh, and Huei-Wen Ferng are with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan (e-mail: bkar@mail.ntust.edu.tw; m10715043@mail.ntust.edu.tw; hwferng@mail.ntust.edu.tw).

Yuan-Cheng Lai is with the Department of Information Management, National Taiwan University of Science and Technology, Taipei 10607, Taiwan (e-mail: laiyc@cs.ntust.edu.tw).

Ying-Dar Lin is with the Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu 572-1274, Taiwan (e-mail: ydlin@cs.nctu.edu.tw).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TVT.2021.3120413>, provided by the authors.

Digital Object Identifier 10.1109/TVT.2021.3120413

and the latency-sensitive tasks can be offloaded to the MEC or VFC servers to reduce transmission delays. In such a multitier architecture, decision-making on offloading is an important issue. For example, some latency-sensitive tasks may not meet their latency requirements because of long transmission delays when UEs offload all the tasks to a MCC server. Likewise, some computation-sensitive tasks may not meet their latency requirement when UEs process all the tasks themselves because they do not have enough processing capacity. Hence, a threshold in time is associated with the task to show whether the requirement is satisfied or not. If the response time of a task exceeds the given threshold, then the measurement of QoS will be considered to be wicked and named a violation of QoS. To evaluate the efficiency of the whole system, the QoS violation probabilities in the different offloading scenarios are estimated so that the offloading probabilities can be optimized. To address such a decision-making problem on offloading, we then derive a probabilistic offloading strategy. All the decisions made about tasks to determine the computational resources to offload are performed probabilistically. To determine the corresponding offloading probabilities, an optimal offloading probability estimation algorithm is proposed. This algorithm consists of two parts: a main probability estimation and a sub-probability estimation. The main probability estimation (MPE) algorithm is used to determine the combined offloading probabilities of the whole system. In contrast, the sub-probability estimation (SPE) algorithm is used to search for the optimal offloading probability of each path. These proposals are summarized as follows:

- 1) First, we propose a UEs, vehicular-fogs, edges, and cloud federated architecture and analyze the offloading probabilities by applying the queueing theory.
- 2) We formulate a problem to minimize the QoS violation probabilities of the federated architecture with delay as the constraint.
- 3) We propose an optimal offloading probability estimation algorithm to estimate the probabilities. Via Matlab, experimentation demonstrates that our architecture has a lower QoS violation probability and average waiting time than other existing architectures.

The rest of this paper is organized as follows. The studies on different federated architectures and offloading are reviewed in Section II. We propose a federated system with cloud, edges, and two vehicular-fogs along with the QoS violation probability derivation in Section III. Section IV covers our solution and the numerical results are covered in Section V. Finally, Section VI concludes this paper.

II. RELATED WORK

To improve performance and satisfy the demands of UEs in a federated system, a service provider offloads its tasks to others. Although offloading in the federated systems has been studied in the literature before, the existing architectures are mostly limited to two-tier federated systems, such as cloud-edge [12], cloud-fog [14], [20], [21], and edge-fog [16], [17]. Some existing research has different objectives, such as minimizing cost, energy, latency, etc. And some papers have similar objectives

as ours but different architectures [22], [24]. Below is a review of the related research papers dealing with offloading in various federated architectures with different objectives.

Rodrigues *et al.* [11] proposed a Cybertwin-based MEC that includes various metrics, such as user mobility, migration of virtual servers, processing, etc., and compared the performance with the VM-based MEC system model. In [12], an omnidirectional cloud-edge federated architecture was proposed to minimize the total cost with the given latency and capacity constraints. To investigate the tradeoff between the energy consumption and the execution delay in a MEC system, Zhang *et al.* [13] proposed the online dynamic tasks assignment scheduling. They formulated an energy consumption and delay minimization problem for mobile devices with the battery level as a constraint. In [14], a resource allocation and task offloading problem was proposed to satisfy the deadline requirements of heterogeneous real-time tasks in the cloud-fog computing system.

Adhikari *et al.* [15] designed a delay-dependent priority-aware task offloading strategy for the task generated by the Internet of things (IoT) devices. Their strategy assigns a priority to each task based on its deadline and uses a multi-level feedback queue to minimize the waiting time, the starvation problem, the delay-sensitive tasks in queue, and the low-priority tasks. In [16], Lin *et al.* proposed a two-tier federated edge and vehicular-fog architecture, where the roadside units (RSUs) are treated as the manager of fogs, with a mixed integer programming problem to minimize the total cost from the edge perspective while meeting the given latency constraint. In [17], Yen *et al.* proposed the multiple edges to multiple fogs offloading architecture as an extension of [16]. Rodrigues *et al.* [18] proposed a configuration algorithm to decide when and where to offload the tasks for mobile MEC users by considering various parameters such as the time needed to transmit and process the tasks locally and remotely.

Ghosh *et al.* [19] proposed a framework called the mobility-aware Internet of spatial things (Mobi-IoST) to process the information and deliver the result based on the mobility prediction of the user's location. The real-time traffic management with fog-based Internet of vehicles (IoV) systems was proposed in [20] to minimize average response times. Its model includes both parked and moving vehicle-based fog nodes and tackles the fog-enabled offloading optimization problem mathematically. Focusing on energy consumption, latency, and cost, Liu *et al.* [21] proposed a multi-objective optimization problem to minimize energy consumption, delay, and payment cost by determining the optimal offloading probabilities in a cloud and a fog federated environment.

In [22], an analytical model was presented for communication and computation offloading of the fifth-generation (5G) vehicles to anything (V2X) using the M/M/1 model. A sub-gradient based algorithm was proposed to minimize the average packet delay to determine the optimal offloading probabilities. In [23], the behavior of an integrated cloud-fog-edge computing system was investigated by using the Markov model. A computational resource allocation heuristic method was applied to maximize a social welfare metric. Hwang *et al.* [24] applied the queueing theory to analyze a federated architecture, which consists of

TABLE I
NOTATION

Notation	Description
M	Number of UEs covered in each edge
N	Number of edges covered by the cloud
c	Number of servers in the M/M/c model
Traffic	
λ	External task arrival rate originally to a UE
$\lambda^{F1}, \lambda^{F2}, \lambda^E, \lambda^C,$ λ^U	Arrival rates of each computational resource, where $F1$ denotes intersection vehicular-fog, $F2$ denotes parking lot vehicular-fog, E denotes edge server or MEC, C denotes cloud server or MCC, and U denotes UEs
$\lambda^{UF1}, \lambda^{F1U}, \lambda^{UF2},$ $\lambda^{F2U}, \lambda^{UE}, \lambda^{EU},$ $\lambda^{EC}, \lambda^{CE}$	Arrival rate to the communication links between U to $F1$, $F1$ to U , U to $F2$, $F2$ to U , U to E , E to U , E to C , and C to E , respectively
Capacity	
$\mu^U, \mu^{F1}, \mu^{F2}, \mu^E,$ μ^C	Service rate of U , $F1$, $F2$, E , and C , respectively
$\mu^{UF1}, \mu^{F1U}, \mu^{UF2},$ $\mu^{F2U}, \mu^{UE}, \mu^{EU},$ μ^{EC}, μ^{CE}	Service capacities of communication links between U to $F1$, $F1$ to U , U to $F2$, $F2$ to U , U to E , E to U , E to C , and C to E , respectively
Delay	
$W(x)$	CDF of the waiting time in the M/M/c model
$w(x)$	PDF of the waiting time in the M/M/c model
θ	Delay constraint of a computation task
W^X	Average waiting time of the whole system
$W^U, W^{F1}, W^{F2},$ W^E, W^C	Average waiting time of each offloading path when tasks are served by U , $F1$, $F2$, E , and C , respectively
Probability	
$P^{UE}, P^{UF1},$ P^{UF2}, P^{EC}	Offloading probabilities from U to E , U to $F1$, U to $F2$, and E to C , respectively
$P_X(\theta)$	QoS violation probability
$P_X^U(\theta), P_X^E(\theta),$ $P_X^C(\theta)$	Probabilities of the delay of a task served by UE, fog, edge, and cloud servers exceeding constraint θ

mobile devices, edges, and cloud and proposed a closed-form solution to derive the service delay distribution. However, their architecture does not include vehicular-fogs.

In summary, many existing papers do not consider cloud-edge-fog federation together. For those papers consider such an architecture, they do not include the vehicular-fog in their system. In this paper, we consider a federated architecture with the mobile device, edge, cloud, and vehicular-fog together and use the queueing theory to analyze and minimize the QoS violation probability. To the best of our knowledge, such an architecture has not been discussed before.

III. SYSTEM DESIGN AND PROBLEM FORMULATION

In this section, we present a cloud, edges, and vehicular-fogs federated architecture and discuss different offloading scenarios. By applying the queueing analysis, the QoS violation probability in each offloading scenario is derived. The variables used are listed in Table I.

A. Proposed System Architecture

Fig. 1 shows our proposed federated architecture. The computational tasks for the UEs follow a Poisson process with an average arrival rate of λ . These tasks can be either processed by the UEs themselves or offloaded to other systems, such as the MEC or vehicular-fogs through the corresponding links. After

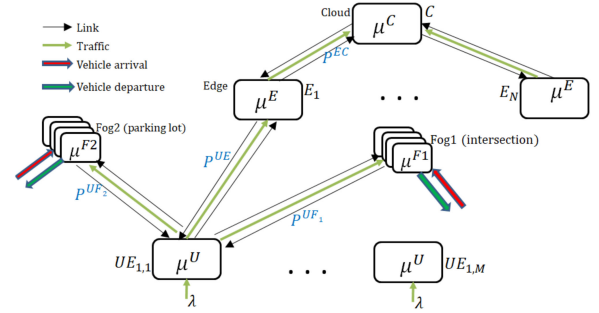


Fig. 1. System architecture of the MCC, MEC, and VFC federated system.

completion, the results will be returned to the UEs. Of course, the MEC and fogs can serve a group of UEs. In some cases, a MEC can offload its tasks to a MCC system. In our architecture, M UEs (with service rate μ^U) within each edge are associated with two fogs, i.e., an intersection vehicular-fog (with service rate μ^{F1}) and a parking lot vehicular-fog (with service rate μ^{F2}), and N edges (with service rate μ^E) are associated with the cloud (with service rate μ^C). We assume that vehicles are dynamically joining or leaving a vehicular-fog because of mobility. In this paper, a probabilistic offloading strategy is chosen to address the decision-making problem. When a UE creates a computation task, it will follow the offloading probabilities to decide where to process the task. Those offloading probabilities are denoted by P^{UE} (probability to offload the task from UEs to edge), P^{UF1} (probability to offload from UEs to the intersection vehicular-fog), P^{UF2} (probability to offload the task from UEs to the parking lot vehicular-fog), and P^{EC} (probability to offload the task from the edge to the cloud). By applying the queueing analysis, the performance of our proposed federated system is evaluated, including the QoS violation probability of each offloading path, which includes the corresponding computing server and links.

B. Problem Formulation

1) *System Model*: When tasks are generated by the UE in our proposed architecture, they can be processed by the UE itself or offloaded to other computation resources, such as an edge server, a cloud server, or a vehicular-fog. This study assumes that each task can be offloaded to one computation resource only and cannot be served partially by different servers. Furthermore, the tasks chosen to offload must be transmitted through links.

Our proposed federated system has five different computing resources: UEs, MEC, MCC, intersection VFC, and parking lot VFC. Therefore, the task can be processed by five different paths and the result can then be returned to the UEs. Each path has three phases that will cost a piece of time. Let us consider the example where a task is offloaded to the MEC server. First, it will take some time to transmit the data required for processing to the MEC server. Then, it needs some time to process data. Finally, the result will be returned from the MEC server, which requires time for transmission. Most of the paths have to transmit data twice and need a task processing period. However, the times of data transmission will be doubled in some cases. Because UEs

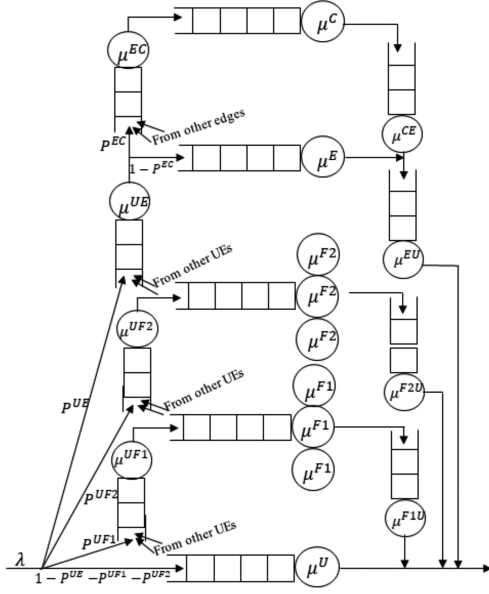


Fig. 2. Queuing network for our proposed federated system.

connect indirectly to the MCC server, the task to be offloaded to the MCC server will first be offloaded to the MEC server and then to the MCC server.

To analyze this system, we apply the queuing theory to model the phases. For each link in the system, MEC, MCC, UEs, an M/M/1 queue is employed, while an M/M/c queue is associated with the VFCs, where c represents the number of computing components (*i.e.*, number of vehicles)¹ in the fog. Note that the use of a parallel processing model could be a realistic scenario in MEC and MCC. However, the M/M/1 model is assumed in this paper to keep our modeling simple although it can not reflect a realistic scenario properly. As shown in Fig. 2, the whole system behaves like an open Jackson network and each offloading path is a tandem queueing system [25]. In this system, each UE can generate tasks by following a Poisson process with an arrival rate of λ . Each computation resource of the MEC and the VFC system is allowed to connect to M UEs within each edge. Furthermore, the task will be offloaded to these computation resources with offloading probabilities P^{UE} , P^{UF1} , and P^{UF2} . The arrival rates for the offloading path served by the MEC, parking lot fog, and intersection fog are then $\lambda^E = \lambda M P^{UE} (1 - P^{EC})$, $\lambda^{F1} = \lambda M P^{UF1}$, and $\lambda^{F2} = \lambda M P^{UF2}$, respectively. The task arrival rate for each UE is $\lambda^U = \lambda (1 - P^{UE} - P^{UF1} - P^{UF2})$. Since the MCC system is allowed to connect to N MEC systems, each MEC may offload the tasks to the MCC with offloading probability P^{EC} . Then, the arrival rate for this offloading path is $\lambda^C = \lambda M N P^{UE} P^{EC}$.

2) *Derivation of the QoS Violation Probability:* For each offloading path, the QoS violation probability is first analyzed. After all the offloading paths have been conducted, the QoS violation probabilities of all the paths are collected to obtain the QoS violation probability of the whole system. In this paper, we consider a homogeneous scenario, where the arrival and service

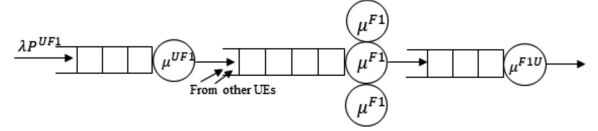


Fig. 3. Offloading path for the tasks served by the VFC server.

rates of all UEs are the same, for simplification. Likewise, similar assumptions apply to communication links, edge servers, and vehicular-fogs.

a) *The task served by the UE:* We assume that the arrival of computation tasks at a UE and the service time follow a Poisson process with arrival rate λ^U and an exponential distribution with rate μ^U . Based on the M/M/1 queueing model, the corresponding delay follows an exponential distribution with rate $\mu^U - \lambda^U$ and the probability that the task delay exceeds the delay constraint denoted by θ is

$$P_X^U(\theta) = P(X_U \geq \theta) = e^{-(\mu^U - \lambda^U)\theta},$$

where X_U is the random variable of the task delay.

b) *The task served by the vehicular-fog:* As shown in Fig. 3, the data required for processing will be transmitted to the server through a link when a computation task is offloaded to the VFC server by the UE. Once completed, the result will be returned to the UE through the link. To find the QoS violation probability of this workflow, the VFC system is treated as an M/M/c queueing model and the link is treated as an M/M/1 queueing model.

Because the offloading path to be analyzed is a tandem queue [25] consisting of three queues (as shown in Fig. 3), we need to apply the convolution technique to obtain the delay distribution of the tandem queue. For simplification, the technique of a Laplace transform is employed. From [25], the overall cumulative distribution function, *i.e.*, CDF, of the delay in an M/M/c queueing model is shown as follows:

$$W(x) = \frac{c(1-\rho) - W_q(0)}{c(1-\rho) - 1} (1 - e^{-v_1 x}) - \frac{1 - W_q(0)}{c(1-\rho) - 1} (1 - e^{-v_2 x}), \quad (1)$$

where

$$W_q(0) = 1 - \frac{r^c p_0}{c!(1-\rho)}, p_0 = \left(\left(\frac{r^c}{c!(1-\rho)} + \sum_{n=0}^{c-1} \frac{r^n}{n!} \right)^{-1}, \right. \\ \left. v_2 = c\mu^F - \lambda, v_1 = \mu^F, r = \frac{\lambda}{\mu}, \rho = \frac{r}{c}. \right.$$

By differentiating (1), the corresponding probability density function, *i.e.*, PDF, denoted by $w(x)$ is shown as follows:

$$w(x) = \frac{c(1-\rho) - W_q(0)}{c(1-\rho) - 1} v_1 e^{-v_1 x} - \frac{1 - W_q(0)}{c(1-\rho) - 1} v_2 e^{-v_2 x} \quad (2)$$

with the following Laplace transform

$$\frac{c(1-\rho) - W_q(0)}{c(1-\rho) - 1} \times \frac{v_1}{s + v_1} - \frac{1 - W_q(0)}{c(1-\rho) - 1} \times \frac{v_2}{s + v_2}.$$

¹Each vehicle in the vehicular-fog is considered as a computing component.

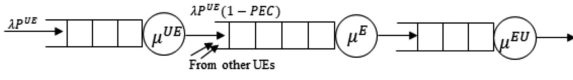


Fig. 4. Offloading path for the task served by the MEC server.

Following the results of [24] and we denoting $v_3 = \mu^{UF} - \lambda^{UF}$ and $v_4 = \mu^{FU} - \lambda^{FU}$ to be the difference between the service rate and the arrival rate of the uploading link and the downloading link, respectively, their Laplace transforms of delay have the following forms:

$$\frac{v_i}{s + v_i}, \quad i = 3, 4.$$

Based on the M/M/1 queueing model, the Laplace transform for the tandem queue is

$$S^*(s) = \left(\frac{c(1-\rho) - W_q(0)}{c(1-\rho) - 1} \times \frac{v_1}{s + v_1} - \frac{1 - W_q(0)}{c(1-\rho) - 1} \times \frac{v_2}{s + v_2} \right) \times \prod_{i=3}^4 \frac{v_i}{s + v_i}. \quad (3)$$

Finally, the QoS violation probability can be derived by inverting the Laplace transform of $S^*(s)$ in (3) and integrating over (θ, ∞) with respect to variable x , i.e.,

$$P_X^F(\theta) = P(X_F \geq \theta) = \int_{\theta}^{\infty} L^{-1}(S^*(s)) dx,$$

where X_F is a random variable associated with the delay and the detailed derivation can be referred to Appendix A. Note that the values of variables v_i , $i = 1, 2, 3, 4$, may lead to different QoS violation probabilities. Therefore, we consider seven cases in Appendix B and give their derivation process in Appendix A. Here, let us discuss the derivation of the QoS violation probability regarding the vehicular-fog for illustration. When there are multiple vehicular-fogs, these derivations can be done separately. In this paper, fog-1 ($F1$) and fog-2 ($F2$) are associated with the traffic intersection fog and the parking lot fog, respectively.

c) The task served by the edge: In the offloading path served by the MEC server, the task has three data processing phases. As shown in Fig. 4, the data required to be processed will be transmitted to the MEC server through the link first. Then, the MEC server will process the data. Finally, the result will be returned through the link. Each data processing phase can be modelled by the M/M/1 queueing model. Similar to the previous derivations, the QoS violation probability can be derived. Here, the difference between the service rate and the arrival rate is denoted by $u_x = \mu_x - \lambda_x$, where $x = UE, E, EU$. If we index the difference in each phase by i , $i = 1, 2, 3$, it may lead to different QoS violation probabilities based on the values of u_i . There are three possible cases and one can refer to [24] for details.

d) The task served by the cloud: This task has five data processing phases in the offloading path of the MCC server as shown in Fig. 5. The data to be processed will be transmitted to the MEC server through the link, then to the MCC server from the MEC server. After being processed by the MCC server,

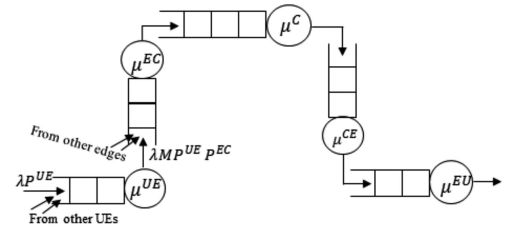


Fig. 5. Offloading path for the task served by the MCC server.

the result will be returned to the MEC server. Finally, the UE can obtain the result from the MEC server through the link. Likewise, each data processing phase can be modelled by the M/M/1 queueing model, where $u_x = \mu_x - \lambda_x$ is the difference between the service rate and the arrival rate with $x = UE, EC, C, CE, EU$. By indexing the difference in each phase via i , $i = 1, 2, 3, 4, 5$, different formulas of the QoS violation probability can be derived based on the values of u_i under seven cases as shown in [24].

After all the offloading paths have been analyzed to obtain the corresponding QoS violation probabilities, the QoS violation probability of the whole system can be expressed as follows:

$$P_X(\theta) = (1 - P^{UE} - P^{UF1} - P^{UF2}) \times P_X^U(\theta) + P^{UF1} \times P_X^{UF1}(\theta) + P^{UF2} \times P_X^{UF2}(\theta) + (1 - P^{EC}) \times P^{UE} \times P_X^E(\theta) + P^{EC} \times P^{UE} \times P_X^C(\theta). \quad (4)$$

3) Average Waiting Time in the System: The average waiting time for each offloading path can be calculated by the formula given in [25]. For example, the waiting time of the offloading path served by the intersection vehicular-fog can be expressed as follows:

$$W^{F1} = \frac{1}{\mu^{UF1} - \lambda^{UF1}} + \frac{1}{\mu^{F1}} + \frac{p_0 \cdot r^c}{c!(c\mu^{F1})(1-\rho)^2} + \frac{1}{\mu^{F1U} - \lambda^{F1U}}. \quad (5)$$

Then, the average waiting time of the whole system denoted by W^X can be obtained as follows:

$$W^X = (1 - P^{UE} - P^{UF1} - P^{UF2})W^U + P^{UF1}W^{F1} + P^{UF2}W^{F2} + (1 - P^{EC})P^{UE}W^E + P^{EC}P^{UE}W^C. \quad (6)$$

4) Problem Statement: The optimal offloading probabilities can be obtained in the sense that $P_X(\theta)$ is minimized with given M UEs connecting to the MEC and VFC servers, N MECs connecting to the MCC server, external arrival rate (λ), service rates of the servers ($\mu^U, \mu^{F1}, \mu^{F2}, \mu^E, \mu^C$), service rates of the links ($\mu^{UF1}, \mu^{F1U}, \mu^{UF2}, \mu^{F2U}, \mu^{UE}, \mu^{EU}, \mu^{EC}, \mu^{CE}$), and delay constraint θ . Hence, the corresponding problem is stated as:

$$\min_{\{P^{UE}, P^{UF1}, P^{UF2}, P^{EC}\}} P_X(\theta), \quad (7)$$

subject to

$$0 \leq P^{UE} + P^{UF1} + P^{UF2} \leq 1, 0 \leq P^{EC} \leq 1.$$

Algorithm 1: Main Probability Estimation (MPE) Algorithm.

Input: $M, N, c, \lambda, \mu^U, \mu^E, \mu^C, \mu^{F1}, \mu^{F2}, \mu^{UE}, \mu^{EU}, \mu^{EC}, \mu^{CE}, \mu^{UF1}, \mu^{F1U}, \mu^{UF2}, \mu^{F2U}$

Output: Optimal $P^{UE}, P^{UF1}, P^{UF2}, P^{EC}$

- 1: Construct set $C = \{c_1, c_2, c_3, c_4\}$. // c_i is a cutting point.
- 2: Initialize $c_i \in C$.
- 3: Calculate:
 $P^{UE} = c_1 - 0, P^{UF1} = c_2 - c_1, P^{UF2} = c_3 - c_2,$
 $P^{EC} = c_4;$
- 4: $newProbEst = P_X(\theta)$ with $P^{UE}, P^{UF1}, P^{UF2}, P^{EC}$ as in (4).
- 5: **repeat**
- 6: $oldProbEst = newProbEst;$
- 7: **for** each $i = 1, 2, 3, 4$ **do**
- 8: Search for new $c_i \in C$ by using SPE.
- 9: **end for**
- 10: Calculate: $P^{UE} = c_1 - 0, P^{UF1} = c_2 - c_1, P^{UF2} = c_3 - c_2, P^{EC} = c_4;$
- 11: $newProbEst = P_X(\theta)$ with $P^{UE}, P^{UF1}, P^{UF2}, P^{EC}$ as in (4).
- 12: **until** ($oldProbEst - newProbEst < \epsilon$)

IV. PROPOSED SOLUTIONS

By the definition of convex optimization stated in [26], it is evident that (7) is a convex optimization problem which can be solved efficiently by using the subgradient iterative method [27]. In this section, a subgradient searching algorithm is devised to determine the optimal probabilities. The algorithm consists of two parts: 1) the main probability estimation, i.e., MPE, algorithm is proposed to estimate the QoS violation probability of the whole system, and 2) the sub-probability estimation, i.e., SPE, algorithm is employed to determine the probability of each offloading path.

A. Main Probability Estimation Algorithm

The MPE algorithm works as follows. To determine the values of the offloading probabilities, such as P^{UE}, P^{UF1} , and P^{UF2} , one can consider a line segment with the unit and cut the line segment into four sections. The lengths of these sections represent the offloading probabilities of P^{UE}, P^{UF1}, P^{UF2} and the probability of the task not being offloaded, i.e., $1 - P^{UE} - P^{UF1} - P^{UF2}$, respectively. In line 3 of Algorithm 1, three cutting points c_1, c_2, c_3 with values, say, 0.25, 0.5, and 0.75, respectively, are posed. So, the offloading probabilities of P^{UE}, P^{UF1}, P^{UF2} , and P^U are all 0.25 initially and $c_4 = 0.5$ is initialized for P^{EC} . From lines 5 to 13, the adjustment to the cutting points in each iteration is performed by using the SPE algorithm, i.e., Algorithm 2, to be discussed in the next subsection. After revising the cutting points by the SPE algorithm, whether the QoS violation probability reaches the minimum value or not under this offloading condition is checked. This process is repeated until the difference between the old QoS

Algorithm 2: Sub-Probability Estimation (SPE) Algorithm.

Input: $M, N, c, \lambda, \mu^U, \mu^E, \mu^C, \mu^{F1}, \mu^{F2}, \mu^{UE}, \mu^{EU}, \mu^{EC}, \mu^{CE}, \mu^{UF1}, \mu^{F1U}, \mu^{UF2}, \mu^{F2U}, c_i$

Output: Optimal $P^X (X = UE, UF1, UF2, EC)$

- 1: Construct set $S = \{s_1, s_2, s_3\}$ and $P = \{p_1, p_2, p_3\}$.
- 2: Initialize: $cuttingPoint = c_i, step = 0.125$.
- 3: **repeat**
- 4: $s_1 = \max(cuttingPoint - step, 0);$
- 5: $s_2 = cuttingPoint;$
- 6: $s_3 = \min(cuttingPoint + step, 1);$
- 7: **for** $i = 1, 2, 3$ **do**
- 8: Adjust the P^{UE}, P^{UF1}, P^{UF2} , and P^{EC} values based on s_i and determine p_i as in (4).
- 9: **end for**
- 10: // Find the new $cuttingPoint$ with minimum p_i value.
- 11: **if** ($p_1 < p_2 \ \&\& \ p_1 < p_3 \ \&\& \ p_1 > 0$)
 $cuttingPoint = s_1;$
- 12: **else if** ($p_2 < p_1 \ \&\& \ p_2 < p_3 \ \&\& \ p_2 > 0$)
 $cuttingPoint = s_2;$
- 13: **else if** ($p_3 < p_1 \ \&\& \ p_3 < p_2 \ \&\& \ p_3 > 0$)
 $cuttingPoint = s_3;$
- 14: $step = step/2;$
- 15: **until** ($step < \zeta$)

violation probability and the new QoS violation probability is less than the given threshold, i.e., ϵ .

B. Sub-Probability Estimation Algorithm

The SPE algorithm is applied to search for the optimal offloading probabilities individually by changing the initial cutting point values. First, a small value, say $step$, is taken forward or backward from the chosen cutting point. Based on this, each cutting point has three values: $cuttingPoint$, $cuttingPoint - step$, and $cuttingPoint + step$. For example, $cuttingPoint = 0.5$ and $step = 0.125$ enable us to have three values for the new cutting points: 0.375 ($cuttingPoint - step$), 0.5 ($cuttingPoint$), and 0.625 ($cuttingPoint + step$). Based on the three possible cutting points, the offloading probabilities are readjusted and the QoS violation probabilities are determined. The cutting points with the minimal QoS violation probability will be chosen as new cutting points by comparing the results using different sets of offloading probabilities. Note that the values of these cutting points must not be less than 0 or greater than 1. The value of $step$ in this algorithm is cut in half before the next iteration. This process is repeated until its value is less than the given threshold, i.e., ζ .

C. Convergence Behavior and Complexity Analysis

Algorithms 1 and 2 are the outer and inner loops of our proposed subgradient method. The MPE algorithm is applied to solve the equivalent convex optimization problem, while the SPE algorithm is utilized to optimize each offloading path individually. It can be seen that ϵ , say, 0.000001, is specified

TABLE II
PARAMETER SETTING FOR MATLAB EXPERIMENTS

Variable	Description	Value
M	Number of UEs in each edge	5
N	Number of MEC servers	5
$c1$	Number of computation components in intersection vehicular-fog	6
$c2$	Number of computation components in parking lot vehicular-fog	10
θ	Delay constraint	1.2 ms
μ^U	Service rate of the UE	1.5 (tasks/ms)
μ^E	Service rate of the MEC server	8 (tasks/ms)
μ^{F1}, μ^{F2}	Service rate of the VFC servers	2 (tasks/ms)
μ^C	Service rate of the MCC server	25 (tasks/ms)
μ^{UE}, μ^{EU}	Service capacity of the link between the UE and the MEC server	12, 11 (tasks/ms)
$\mu^{UF1}, \mu^{F1U}, \mu^{UF2}, \mu^{F2U}$	Service capacity of the link between the UE and the VFC server	13, 12 (tasks/ms)
μ^{EC}, μ^{CE}	Service capacity of the link between the MEC server and the MCC server	22, 21 (tasks/ms)

to terminate the outer loop when the difference between the iterations on the QoS violation probability estimation is less than ϵ . The proposed SPE algorithm converges within twelve iterations for every time it was called from the MPE algorithm as we fix $step$ at 0.125 and ζ at 0.0001.

The running complexity of the algorithm is analyzed as follows. The total input into the system is $MN\lambda$. The MPE algorithm ensures a convergence rate of $O(\frac{1}{\epsilon})$ iterations to achieve an error lower than or equal to ϵ . Similarly, the SPE algorithm gives a convergence rate of $O(\frac{1}{\zeta})$ iterations to achieve an error lower than or equal to ζ . Hence, the worst-case running complexity of the algorithm is $O(\frac{MN\lambda}{\epsilon\zeta})$.

V. NUMERICAL RESULTS AND DISCUSSIONS

A. Parameter Setting

In this section, the analytical results of our proposed model are carried out by Matlab. Our proposed federated architecture is to be compared with three other federated architectures: federated system of MEC and MCC without fogs [24], federated system of MEC and MCC with the traffic intersection fog, and federated system of MEC and MCC with the parking lot fog. Theoretically, the service rate of the MCC system will be the highest among the computation resources. More components in the parking lot vehicular-fog than the intersection vehicular-fog will be observed normally. The parameters employed in our experiments are given in Table II.

B. Validation on Analytical Model

As illustrated in Fig. 6, the analytical results are compared with the simulation results shown by the 95% confidence interval (CI) calculation based on ten simulation runs by the t -distribution. As illustrated by Fig. 6, the analytical results and the simulation results match.

C. Performance Analysis

1) *Comparison of $P_X(\theta)$ in Different Architectures:* Fig. 7 shows the QoS violation probabilities of the four different federated architectures discussed in this paper. The result shows that the QoS violation probability in the federated architectures with

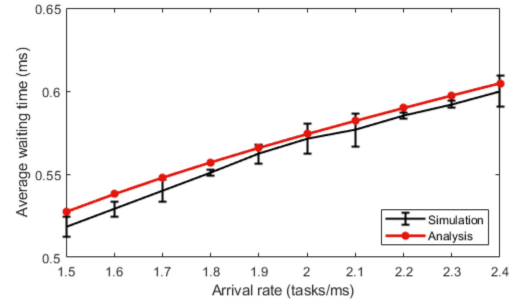


Fig. 6. Validation on the analytical model.

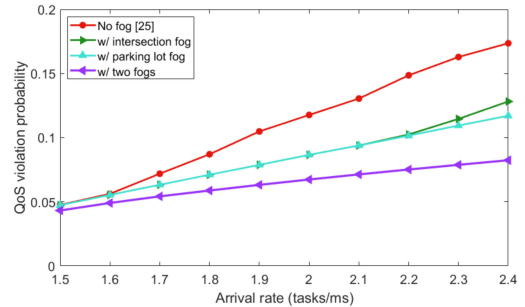


Fig. 7. Comparison of QoS violation probabilities in different architectures.

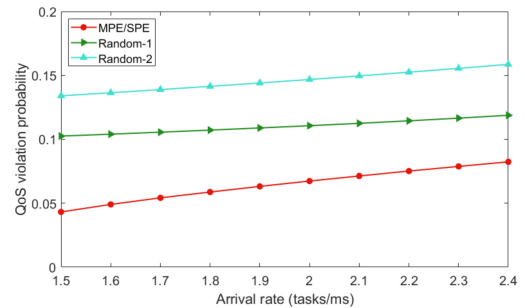


Fig. 8. Comparison of QoS violation probabilities in different algorithms.

vehicular-fogs is less than the architecture without fogs [24]. In particular, the architecture with two vehicular-fogs has nearly 10%-12% lower QoS violation probability than the federated systems without fogs [24]. This is because the systems with vehicular-fogs have more computation resources closer to UEs than the ones without fogs.

2) *$P_X(\theta)$ - MPE/SPE Vs. Random:* Fig. 8 shows the QoS violation probabilities comparison between the proposed MPE/SPE algorithm and the solutions where offloading is completely random in the cloud-edge federated architecture with two vehicular-fogs. We consider two sets of random offloading probabilities in terms of $(P^U, P^{UF1}, P^{UF2}, P^{UE}, P^{EC})$: Random-1, i.e., (0.2, 0.2, 0.2, 0.4, 0.5) and Random-2, i.e., (0.25, 0.25, 0.25, 0.25, 0.5). The result shows the QoS violation probability of our MPE/SPE algorithm is 5%-10% less than the two random algorithms. This is because the offloading probabilities in the random algorithms are fixed irrespective of available capacities of the computing servers, while the offloading probabilities are optimized in the MPE/SPE algorithm.

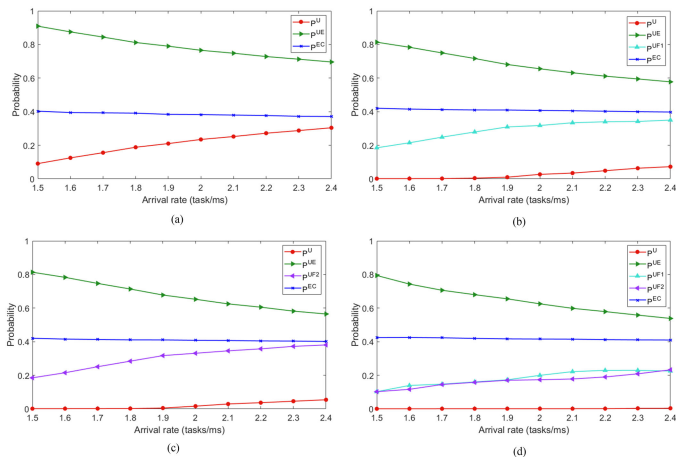


Fig. 9. Effect on the offloading probabilities in different architectures under various arrival rates: (a) federated system without vehicular-fogs [24], (b) federated system with the intersection vehicular-fog, (c) federated system with the parking lot vehicular-fog, and (d) federated system with two vehicular-fogs.

3) *Effect of Arrival Rates on Offloading Probabilities:* Here, we demonstrate the offloading probabilities for different federated architectures under various arrival rates. Since the goal is to minimize the QoS violation probability as shown in (7), guaranteeing task completion with the minimal QoS violation probability should have the highest priority. Thus, tasks will be offloaded to the computation resources with high data processing capacity but less communication delay.

Figs. 9(a)–(d) illustrate various offloading scenarios for the four different architectures. Fig. 9(a) shows the offloading scenario for the federated system with UEs, edge servers, and cloud servers without the vehicular-fog. Note that the external arrival rate is set lower than half of the rate to be processed by the MEC and MCC servers. When the external arrival rate goes up, the volume of tasks may exceed the capacity of the system (a task cannot be finished in time under such a case) even if it is possible to offload to other computational resources. UEs will therefore have more opportunities to process tasks by themselves under such a condition. Similarly, the offloading scenarios for UEs, edge, and cloud federated architectures with the traffic intersection fog, the parking lot fog, and both parking lot and traffic intersection fogs are shown in Figs. 9(b)–(d), respectively. As shown in Figs. 9(b)–(c), tasks are offloaded to MEC, MCC, and the fog when the arrival rate is low. On the contrary, more input tasks need to be processed by UEs when the arrival rate increases. As a result of the presence of the vehicular-fog in Figs. 9(b)–(c), the offloading probability to MEC decreases as compared to that in Fig. 9(a) because the tasks can be offloaded to the fog. As for our proposed architecture, i.e., UEs, edge, and cloud federated architecture with two types of fog, the results are shown in Fig. 9(d). The two fogs serve as the external computing resources for accepting the offloaded tasks. As a result, the computational load on the UEs is (relatively) much smaller as compared to the other architectures. One can observe that P^{EC} mostly keeps stable in all traffic conditions as seen in Figs. 9(a)–(d). When the cloud processes the task, it adds an extra communication delay. To avoid the delay caused by an

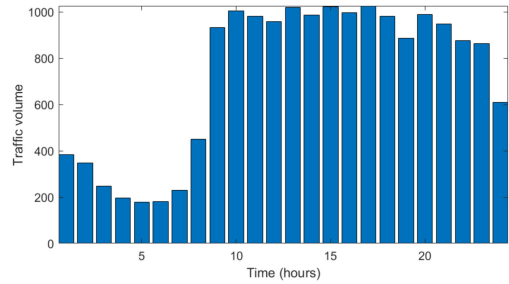


Fig. 10. Average daily traffic volume of Taichung City collected from [28].

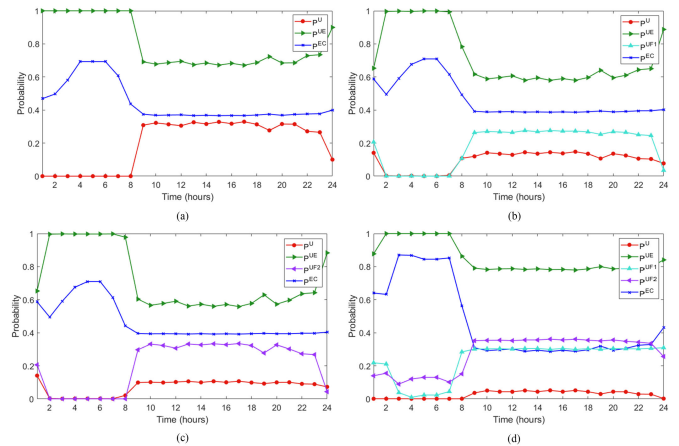


Fig. 11. Effect on the offloading probabilities in different architectures based on the realistic traffic: (a) architecture without fogs [24], (b) architecture with the intersection vehicular-fog, (c) architecture with the parking lot vehicular-fog, and (d) architecture with two types of vehicular-fog.

increase in traffic inputs, the tasks are processed by UE. Hence, P^U in Fig. 9(a) increases. Figs. 9(b)–(c) show that tasks are offloaded to the vehicular-fog, and UEs process some of the tasks. In Fig. 9(d), tasks are offloaded to the two fogs. As a result, the load on UEs is negligible.

4) *Offloading Probabilities With the Realistic Traffic:* To create a realistic traffic condition, we collected traffic data hourly over a week. We then calculated the average traffic volume hourly for a day. The datasets are actual traffic flow data from the Taichung City Transportation Bureau [28]. Fig. 10 shows the average traffic volume in a day, where heavy traffic flow occurs from 9:00 AM to 11:00 PM. Such heavy traffic flow results in more users who may make requests within that area. Undoubtedly, the external arrival rate becomes high during these hours. We can also determine the vehicle arrival rate for the intersection vehicular-fog and the parking lot vehicular-fog based on these data. With these observations, we adjusted the external arrival rate in proportion to the hourly traffic volume while running the simulation and the optimal offloading probabilities of a day for different federated architectures were obtained accordingly.

The corresponding offloading probabilities for a day with different federated architectures are shown in Figs. 11(a)–(d). Fig. 11(a) shows the offloading probabilities of the UEs, MEC, and cloud federated architecture without fogs. Like Fig. 9(a), tasks are offloaded to the edge when the traffic arrival rate is

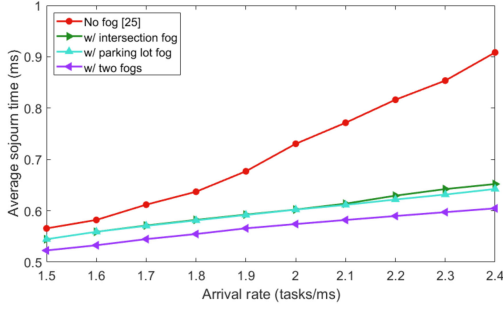


Fig. 12. Average waiting time under different federated architectures.

low, while some of the tasks are processed by UEs in peak traffic hours. Similarly, the offloading scenarios of UEs, edge, and cloud federated architectures with the traffic intersection fog, the parking lot fog, and both parking lot and traffic intersection fogs are shown in Figs. 11(b)–(d), respectively. Analogous to Figs. 9(b)–(c), tasks are offloaded to the MEC, MCC, and fog during low traffic-flow hours as shown in Figs. 11(b)–(c). During the peak traffic hours, some tasks are processed by UEs. Again, the presence of the vehicular-fog in Figs. 11(b)–(c) causes fewer tasks to be processed by UEs as compared to the case shown in Fig. 11(a). Likewise, the result of Fig. 11(d) is similar to Fig. 9(d). Here, more tasks are offloaded to the fogs because of the presence of two types of fog. As a result, the computational load on UEs is considerably less. In Fig. 11(a), P^U and P^{UE} are nearly 0.3 and 0.7, respectively, under the heavy traffic conditions. In Figs. 11(b)–(c), these probabilities decrease to 0.15 and 0.6 or so because of the inclusion of the vehicular-fog in the architecture. These values are reduced further to 0.05 and 0.35 in Fig. 11(d) because two vehicular-fogs are included. This implies that P^U and p^{UE} under heavy traffic in Fig. 11(d) are reduced nearly by 25% and 35%, respectively, as compared to Fig. 11(a) because vehicular-fogs are included.

5) *Comparison of Waiting Time in Different Architectures:* This is an essential factor in evaluating the system performance because it becomes challenging to handle highly time-sensitive applications when a high waiting time is incurred. The waiting time of tasks should therefore be kept as low as possible. In our architecture, we assume that it must release the resources once completed when the resources are assigned to a task. Fig. 12 shows the average sojourn time of the four different architectures under different traffic arrival rates. Fig. 12 reveals explicitly that a federated system with vehicular-fogs has the lowest average sojourn time as compared to the architecture without fogs. In particular, the architecture with two fogs can reduce the average sojourn time by 45% as compared to the federated architecture without fogs because the systems with vehicular-fogs have more computation resources than the one without fogs. Hence, it can process more tasks concurrently. In conclusion, the federated system without fogs has the longest average sojourn time and its average sojourn time increases rapidly as the arrival rate increases.

6) *Average Waiting Time - MPE/SPE Vs. Random:* Fig. 13 shows the average waiting time comparison between our proposed MPE/SPE algorithm and random solutions in our proposed federated architecture with two vehicular-fogs. We have

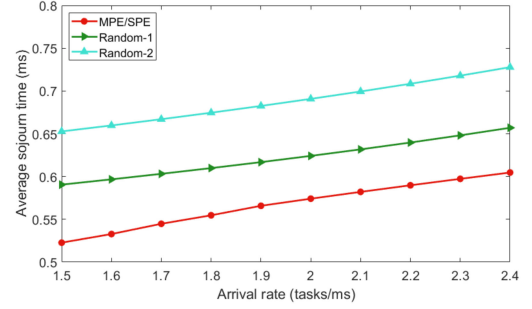


Fig. 13. Comparison of average waiting time in different algorithms.

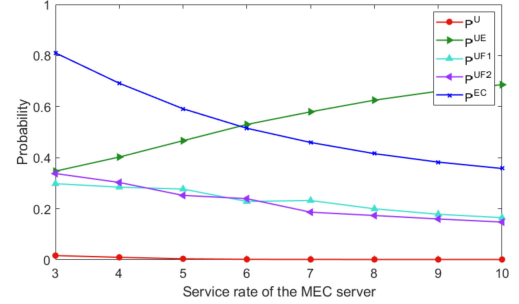


Fig. 14. Variation in the edge capacity.

considered the two random offloading probabilities employed in Fig. 13. The result shows the average sojourn time of our MPE/SPE algorithm is 7.5%-12.5% less than the two random algorithms. Since the offloading probabilities are fixed in the two random algorithms, fewer computing servers have limited and less capacity but are forced to accept more requests, increasing the average sojourn time accordingly. However, in our MPE/SPE algorithm, the tasks are offloaded in an optimized manner.

7) *Change in the Edge Capacity - P^{UE} Vs. P^{EC} :* When an edge server capacity is low in the federated system, the task offloaded to the edge may be further offloaded to the cloud, which needs additional communication delays. With the increase in the edge capacity, the possibility of the task offloaded to the edge getting served by the edge increases, while the possibility of offloading to the cloud decreases. Fig. 14 shows the effect caused by the change in the service rate of MEC servers in the federated system. The results show that more tasks are offloaded to the MEC servers and become processed accordingly when the service rate of the MEC server increases. As a result, P^{EC} decreases.

8) *Change in the Vehicular-Fog Capacity - P^{UF} Vs. P^{UE} :* In the previous experiments, we kept the number of vehicles in a vehicular-fog constant by considering the same arrival and departure rates of vehicles. But, they are not realistic. With a change in the arrival rate or the departure rate, the number of vehicles in a fog changes dynamically. Since these vehicles are considered as the computing components in the VFC system, the number of vehicles affects the service rate of VFC system. By changing the number of vehicles in a vehicular-fog, the corresponding computing capacity of the fog is changed accordingly. For example, the fog size at a road intersection during the peak hours will be large, increasing the capacity of the fog to process tasks. Fig. 15 shows that offloading to the

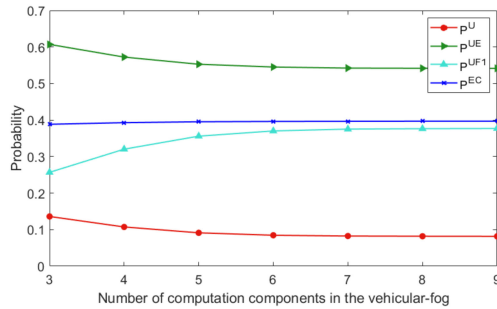


Fig. 15. Change in the no. of vehicles in the fog.

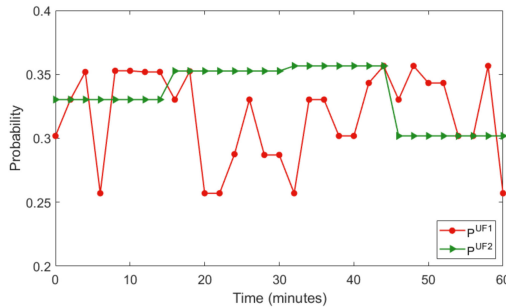


Fig. 16. Comparison of two vehicular-fogs.

intersection fog increases as it can process more tasks with the increase in the vehicular-fog size. While P^{UF1} increases, P^{UE} decreases as shown in Fig. 15 because the fogs are closer to the UE than the edge.

9) P^{UF1} Vs. P^{UF2} : The number of vehicles in the traffic intersection may change rapidly as a vehicle can remain for a maximum of 100-120 sec. However, a vehicle can stay longer in the parking lot, sometimes 10-15 minutes, or even hours or days. In the vehicular-fog, the size decreases when some vehicle departs and increases only when new vehicles arrive. Fig. 16 shows how these changes impact the offloading probabilities. In our experiment, we changed the number of computing components randomly in the intersection vehicular-fog every 2 minutes on average, and that in the parking lot vehicular-fog every 15 minutes. As shown in Fig. 16, P^{UF1} changes every 2 minutes and P^{UF2} changes every 15 minutes. In Fig. 16, the changes in P^{UF1} and P^{UF2} depend on the computing components of fog-1 and fog-2, respectively. The offloading probabilities increase when there is an increase in the computing components of the fog and decrease when there is a decrease in the corresponding computing components.

VI. CONCLUSION

The cloud, edges, and vehicular-fogs federated architecture as proposed in this paper addresses how UEs can offload tasks to the edge and vehicular-fogs, and how edges can further offload tasks to the cloud. In this federated architecture, we consider the probabilistic offloading strategy to investigate the delay constraint problem by determining the QoS violation probabilities with the aids of a probability estimation algorithm. Our numerical results show that our proposed architecture reduces the QoS

violation probability and the average waiting time by 10%-12% and 45%, respectively, as compared to the federated architecture without fogs [24]. During the peak traffic hours, P^{UE} in the architecture with two vehicular-fogs can be reduced by nearly 35% as compared to the architecture without fogs [24].

In the future work, we intend to extend our work in a number of ways. We plan to extend our results to a heterogeneous scenario. We plan to consider the vehicle arrival and departure rates in a more realistic scenario. The assumption related to the mobility in vehicular-fogs and its effect on different offloading scenarios in the federated systems can be further investigated.

REFERENCES

- [1] D. George and P. Demestichas, "Intelligent transportation systems," *IEEE Veh. Technol. Mag.*, vol. 5, no. 1, pp. 77–84, Mar. 2010.
- [2] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comput.*, vol. 13, no. 18, pp. 1587–1611, 2013.
- [3] J. Zheng, Y. Cai, Y. Wu, and X. Shen, "Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 771–786, Apr. 2019.
- [4] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surv. Tuts.*, vol. 19, no. 3, pp. 1628–1656, Jul.-Sep. 2017.
- [5] H. Guo, J. Liu, J. Zhang, W. Sun, and N. Kato, "Mobile-edge computation offloading for ultra-dense IoT networks," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4977–4988, Dec. 2018.
- [6] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.
- [7] Z. Zhou, P. Liu, J. Feng, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3113–3125, Apr. 2019.
- [8] Y. Zhang, C.-Y. Wang, and H.-Y. Wei, "Parking reservation auction for parked vehicle assistance in vehicular fog computing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3126–3139, Apr. 2019.
- [9] J. Wang, C. Jiang, K. Zhang, T. Q. S. Quek, Y. Ren, and L. Hanzo, "Vehicular sensing networks in a smart city: Principles, technologies and applications," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 122–132, Feb. 2018.
- [10] Z. Ning, J. Huang, and X. Wang, "Vehicular fog computing: Enabling real-time traffic management for smart cities," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 87–93, Feb. 2019.
- [11] T. K. Rodrigues, J. Liu, and N. Kato, "Application of cybertwin for offloading in mobile multi-access edge computing for 6G networks," *IEEE Internet Things J.*, vol. 8, no. 22, pp. 16231–16242, Nov. 2021.
- [12] B. Kar, Y.-D. Lin, and Y.-C. Lai, "OMNI: Omin-directional dual cost optimization of two-tier federated cloud-edge systems," in *Proc. IEEE Int. Conf. Commun.*, Dublin, Ireland, 2020, pp. 1–7.
- [13] G. Zhang, W. Zhang, Y. Cao, D. Li, and L. Wang, "Energy-delay trade-off for dynamic offloading in mobile-edge computing system with energy harvesting devices," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4642–4655, Oct. 2018.
- [14] L. Li, Q. Guan, L. Jin, and M. Guo, "Resource allocation and task offloading for heterogeneous real-time tasks with uncertain duration time in a fog queueing system," *IEEE Access*, vol. 7, no. 1, pp. 9912–9925, Jan. 2019.
- [15] M. Adhikari, M. Mukherjee, and S. N. Srirama, "DPTO: A deadline and priority-aware task offloading in fog computing framework leveraging multilevel feedback queueing," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5773–5782, Jul. 2019.
- [16] Y. Lin, J. Hu, B. Kar, and L. Yen, "Cost minimization with offloading to vehicles in two-tier federated edge and vehicular-fog systems," in *Proc. IEEE Veh. Technol. Conf.*, Hawaii, USA, 2019, pp. 1–6.
- [17] L. Yen, J. Hu, Y. Lin, and B. Kar, "Decentralized configuration protocols for low-cost offloading from multiple edges to multiple vehicular fogs," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 872–885, Jan. 2021.

- [18] T. K. Rodrigues, J. Liu, and N. Kato, "Offloading decision for mobile multi-access edge computing in a multi-tiered 6G networks," *IEEE Trans. Emerg. Topics Comput.*, to be published, doi: [10.1109/TETC2021.30900061](https://doi.org/10.1109/TETC2021.30900061).
- [19] S. Ghosh, A. Mukherjee, S. K. Ghosh, and R. Buyya, "Mobi-IoST: Mobility-aware cloud-fog-edge-IoT collaborative framework for time-critical applications," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2271–2285, Apr. 2019.
- [20] X. Wang, Z. Ning, and L. Wang, "Offloading in internet of vehicles: A fog enabled real-time traffic management system," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4568–4578, Oct. 2018.
- [21] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 283–294, Feb. 2018.
- [22] R.-H. Hwang, M. M. Islam, M. A. Tanvir, M. S. Hossain, and Y.-D. Lin, "Communication and computation offloading for 5G V2X: Modeling and optimization," in *Proc. IEEE Glob. Commun. Conf.*, 2020, pp. 1–6.
- [23] R. Fantacci and B. Picano, "Performance analysis of a delay constrained data offloading scheme in an integrated cloud-fog-edge computing system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12004–12014, Oct. 2020.
- [24] R.-H. Hwang, Y.-C. Lai, and Y.-D. Lin, "Offloading optimization with delay distribution in the 3-tier federated cloud, edge, and fog systems," 2021, arxiv.org/abs/2107.05015.
- [25] D. Gross and C. Harris, *Fundamentals of Queueing Theory*. Hoboken, NJ, USA: Wiley, 1985.
- [26] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. Philadelphia, PA, USA: SIAM, 2001, vol. 2.
- [27] S. Boyd, X. Lin, and A. Mutapcic, "Subgradient methods," *Lecture Notes of EE392o*. Stanford, CA, USA: Stanford University, Autumn Quarter 2003–2004.
- [28] "Taichung city realtime traffic information," [Online]. Available: <http://e-traffic.taichung.gov.tw/RoadGrid/Pages/VD/History2.html>.



optimaization, queueing theory, and network security.

Binayak Kar (Member, IEEE) received the Ph.D. degree in computer science and information engineering from National Central University, Taoyuan City, Taiwan, in 2018. He is currently an Assistant Professor of computer science and information engineering with the National Taiwan University of Science and Technology, Taipei, Taiwan. From 2018 to 2019, he was a Postdoctoral Research Fellow of computer science with National Chiao Tung University, Hsinchu, Taiwan. His research interests include network softwarization, cloud/edge/fog computing,



Kuan-Min Shieh received the B.S. degree from the Department of Information Management, National Dong Hwa University, Hualien, Taiwan, and the M.S. degree in computer science and information engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan.



Yuan-Cheng Lai received the Ph.D. degree from the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, in 1997. In August 2001, he joined the faculty of the Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan, and has been a Distinguished Professor since June 2012. His research interests include performance analysis, software-defined networking, wireless networks, and IoT security.



Ying-Dar Lin (Fellow, IEEE) received the Ph.D. degree in computer science from the University of California, Los Angeles, CA, USA, in 1993. He is currently a Chair Professor of computer science with National Yang Ming Chiao Tung University, Taiwan. He was a Visiting Scholar with Cisco Systems, San Jose, CA, USA, during 2007–2008, the CEO with Telecom Technology Center, Taiwan, during 2010–2011, and the Vice President of National Applied Research Labs, Taiwan, during 2017–2018. Since 2002, he has been the Founder and the Director of Network Benchmarking Lab, which reviews network products with real traffic and automated tools, and has been an approved test lab of the Open Networking Foundation since July 2014. He has also Co-Founded L7 Networks Inc. in 2002, later acquired by D-Link Corp., and O'Prueba Inc. in 2018. He is the author or coauthor of textbook, *Computer Networks: An Open Source Approach*, with Ren-Hung Hwang and Fred Baker (McGraw-Hill, 2011). His research interests include network security, wireless communications, and network softwarization. His work on multi-hop cellular was the first along this line, and has been cited more than 1000 times and standardized into IEEE 802.11 s, IEEE 802.15.5, IEEE 802.16j, and 3GPP LTE-Advanced. He is an IEEE Distinguished Lecturer (2014–2017), ONF Research Associate, and was the recipient of the 2017 Research Excellence Award and K. T. Li Breakthrough Award. He has served or is serving on the editorial boards of several IEEE journals and magazines, and was the Editor-in-Chief of IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, during 2016–2020.



Hwei-Wen Ferng (Senior Member, IEEE) received the B.S. degree in electrical engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 1993 and the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2000. In 2001, he joined the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, as an Assistant Professor, where he was an Associate Professor from 2005 to 2011, has been a Professor and a Distinguished Professor since 2011 and 2012, respectively, and was the Department Head from 2016 to 2019. In 2003, he visited the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA, funded by the Pan Wen-Yuan Foundation, Taiwan. His research interests include wireless networks, mobile computing, high-speed networks, protocol design, teletraffic modeling, queueing theory, and performance analysis. He was the recipient of the Research Award for Young Researchers from the Pan Wen-Yuan Foundation, Taiwan, in 2003 and the Outstanding Young Electrical Engineer Award from the Chinese Institute of Electrical Engineering, Taiwan, in 2008.