IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 1 September 2024, accepted 3 October 2024, date of publication 9 October 2024, date of current version 26 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3476679

RESEARCH ARTICLE

What If VEC Is Moving: Probabilistic Model of Task Execution Through Offloading in Vehicular Computing Environments

ASMAA IBRAHIM AND BASSEM MOKHTAR, (Senior Member, IEEE)
Computer and Network Engineering Department, College of Information Technology, United Arab Emirates University, Al Ain, United Arab Emirates
Corresponding author: Bassem Mokhtar (bassem.mokhtar@uaeu.ac.ae)

This work was supported by the Start-Up Research Program—United Arab Emirates University, under Grant G00004585/12T050.

ABSTRACT Various computing approaches within vehicular networks, such as vehicular edge computing (VEC) and cloud computing, have been suggested to facilitate task offloading, aiming to improve user satisfaction. The features of vehicular networks, including the rapid movement of vehicles and the fluctuating distribution of vehicle densities, present challenges to task offloading with in the VEC. Numerous algorithms have been suggested to address these challenges and provide an effective task-offloading framework. This paper introduces a probabilistic model that analyzes task offloading across different computing tiers, alongside proposing a mobile computing paradigm tailored to the dynamic nature of vehicular networks (VN). This paradigm aims to maintain persistent connectivity and enhanced connection reliability despite mobility facilitating sustainable end-to-end service delivery. Building upon this premise, we propose a three-tier computing paradigm comprising Vehicle Edge of Things (VEoTC), VEC, and Cloud Computing (CC). Within the VEoTC tier, Service Vehicles (SV) equipped with computational resources serve as the mobile computing layer. The proposed model ensures continuous connectivity by extending the dwell time between the service requester and the vehicular computational resource. The model ensures that the relative speed between the service vehicle (representing computational resources) and the service requester remains constant while within the communication range. We proposed a probabilistic model for the end-to-end serving time of the proposed computing paradigm. Then, we computed the dwell time between the SV and the served vehicle based on real data published by Didi Chuxing GAIA Initiative for Chengdu city, China. Utilizing a simulated model, we illustrated the additional penalty incurred by the road side unit (RSU) handovers.

INDEX TERMS Probabilistic model, task offloading, vehicular communication network, vehicular cloud computing, vehicular edge computing, vehicle edge of things.

I. INTRODUCTION

Internet of Vehicles (IoV) has sparked considerable research attention as a pivotal component of Intelligent Transportation Systems (ITS). In earlier times, Vehicle-to-Vehicle (V2V) communication via an onboard unit (OBU) was introduced to establish Vehicular Ad Hoc Networks (VANETs). Furthermore, the Internet of Things (IoT), which gathers diverse data from sensors and the surrounding environment, is revolutionizing and amplifying the demand for vehicular networks. By blending the principles of the IoT with VANET networks, the IoV emerges, facilitating Vehicles-to-Everything (V2X) communication. Recent studies advocate for two primary communication approaches facilitating V2X communications: Direct Short Range Communication (DSRC) and Cellular-based Vehicular Communication (C-V2X). These approaches have been standardized by the IEEE and 3GPP, respectively. However, both encounter limitations and challenges. Consequently, heterogeneous vehicular networks have been proposed to address these limitations and enhance the user experience.

The associate editor coordinating the review of this manuscript and approving it for publication was Tariq Umer.

Extensive research efforts have put forward various network typologies to characterize the IoV network. In [1] and [2], the author delineates six network elements to describe the IoV network, encompassing the OBU, roadside units (RSUs), and the edges computing units deployed with cloud computing network architecture. While the conventional cloud computing architecture boosts the computational abilities of connected devices, it lacks the capability to meet the low-latency and real-time requirements of applications [3] because of the distance from remote cloud servers, heavy reliance on the internet, and congestion in the back-haul network. The paradigms of Multi-Access Edge Computing (MEC) [4], [5], [6] and fog computing [7] have been suggested for executing cloud computing tasks at the network's edge, in proximity to connected devices, while also deploying CC. Vehicular edge computing (VEC) has been suggested as a solution to address the challenge posed by the limited resources of vehicular sensor networks and the low latency requirements of applications [8]. VEC extends computing services to vehicles to amplify their computational resources through the implementation of VEC in conjunction with RSUs.

Implementing VEC alongside the RSU addresses the computational demands and time-critical requirements posed by new applications. Conversely, in certain scenarios, the VEC configuration can lead to longer processing times due to excessive network congestion, the rejection of some requests, rapid vehicle movement, and the restricted coverage area of the RSU. This motivates the research for various types of clouding, as dynamic vehicle computing [9], [10], and volunteer computing based VANET (VCBV) [11], [12], [13]. The author of [9] presents a dynamic cloud computing framework utilizing vehicles, which receives requests from the closest RSU, conducts computations, and then forwards them back to the nearest RSU for dissemination throughout the network. While in [12], the author introduced VCBV, a system designed to fulfill user requests using available vehicles resources through multi-hop communication in high-traffic areas, thereby decreasing the request rate on the infrastructure. The author expanded their research in [11] by introducing two distinct types of VCBV: ad hoc VCBV, where tasks are scheduled and executed using multiple vehicles, and RSU-based VCBV, where the RSU is tasked with scheduling and distributing requests to the vehicles.

In literature, the utilization of resources through vehicles has been presented in [10] and [14], where the author suggests a multi-hop communication model. In this model, a Road Side Unit (RSU) first receives an offloading request, then delegates it to a service vehicle for computation, and ultimately uploads it to the nearest RSU. However, this network setup experiences prolonged end-to-end service times, which fail to meet the low-latency requirements. In [15], the authors presented public transportation vehicles as mobile routers responsible for distributing user requests throughout the network by identifying routes. In this configuration, the vehicles are equipped solely with LAN modules and lack computational capabilities. Introducing computational resources within specific vehicles has been previously suggested in [16], where the authors proposed a network architecture consisting of three layers. This model implements Vehicle edge of things computing (VEoTC), enabling vehicles to possess computational resources and execute computational tasks requested by other devices. Recently different V2V communication schemes have been proposed to enhance the VANET reliability and offers secured communications. In [17], the author introduces a novel model for Emergency Message Dissemination called Trust Cascading-based Emergency Message Dissemination (TCEMD). This model integrates entity-oriented trust values into data-oriented trust assessment efficiently. When an emergency arises, the model facilitates the spread of emergency messages among nearby vehicles using a trust cascading approach, where entity-oriented trust values serve as crucial weights. These trust values are assessed and updated using trust certificates and are included in the messages. While in [18], the author proposes a novel Privacy Preserving Trust Management (PPTM) scheme for the emergency message dissemination in SAGIVNs. As he provides formal verification of the quantitative relationship between the false positive rate—caused by the fuzzification of reputation scores and trust thresholds—and the number of alternative reputation and threshold levels. This work considers the Space-Air-Ground Integrated Network (SAGIN) architecture, which can significantly enhance the performance of vehicular networks by utilizing the unique benefits of space, air, and ground segments. This integration improves coverage, flexibility, reliability by developing the Space-Air-Ground Integrated Vehicular Networks (SAGIVNs). Moreover, the privacy preserving reputation has been extended to the cloud assisted networks. In [19] the author introduces a novel Privacy-Preserving Reputation Updating (PPRU) scheme for cloud-assisted vehicular networks based on the Elliptic Curve Cryptography (ECC) and Paillier algorithms. In the proposed model the reputation feedbacks are collected and preprocessed by the honest-but-curious Cloud Service Provider (CSP) in a privacy preserving manner. In [20] a well-balanced approach between trust evaluation and privacy preservation, with minimal overhead, to support distributed data fusion in cooperative vehicular safety applications. As the author proposes a novel LPPTE scheme that effectively balances trust evaluation and privacy preservation while minimizing computation, communication, and storage overheads. Another Lightweight Trustworthy Message Exchange (LTME) scheme for UAV networks has been proposed in [21]. LTME efficiently aggregates the cryptography and trust management technologies using centralized Ground Control Station (GCS), that periodically updates the reputation levels of registered UAVs and securely distributes secret values to the UAVs. Using these secret values, each trustworthy broadcasting UAV creates encrypted messages that can be decrypted solely by trustworthy receiving UAVs. Furthermore, each trustworthy receiving UAV can evaluate, efficiently and with minimal overhead, the reliability of both the received messages and the broadcasting UAVs. The aforementioned VANET communication schemes each utilize different algorithms and methodologies to define trusted vehicles for secure communications. In contrast, the proposed service vehicle model establishes a secure connection through a single dedicated trusted vehicle designated as the service vehicle. This approach allows the model to ensure secure communication across all connected vehicles with minimal authentication and trust-related overhead. The VANET communication schemes, mentioned above, each utilize different algorithms and methodologies to define trusted vehicles for secure communications. In contrast, the proposed service vehicle model establishes a secure connection through a single dedicated trusted vehicle designated as the service vehicle. This approach allows the model to ensure secure communication across all connected vehicles with minimal authentication and trust-related overhead.

This paper introduces the probabilistic approach to the End-to-End serving time through task offloading to a 3-tier computing model, where service vehicles equipped with computational resources handle computational requests from other connected devices. The previously proposed VEoTC in [16], operates under the condition that continuous connectivity is maintained between mobile service requesters and resource providers, as long as the requesters remain within the communication range of the providers. This setup meets the low-latency requirements of certain real-time applications and services. These performance improvements come with the trade-off of higher costs for deploying edge computing capabilities in each service vehicle. We investigate the impact of deploying computation-capable vehicles on the end-to-end delay of the network. The remainder of the paper is organized as follows: Section II proposes the system model. Section III introduces the problem formulation and the adopted mathematical model. Section IV provides and discusses the simulation scenarios and the obtained results. Finally, the paper is concluded in Section V with outlining for future work.

II. SYSTEM MODEL

In this paper, we present a model of computing architecture featuring three tiers of computing capabilities: vehicle edge of things computing (VEoTC), vehicle edge computing (VEC), and vehicle cloud computing (VCC), as shown in Fig. 1. The three tiers are integrated and operate collaboratively to meet quality of service requirements and enhance user experience. The proposed VEoTC is introduced to address the gap and mitigate the limitations of VEC and VCC in meeting the low-latency requirements of specific real-time applications. In the VEoTC system, particular vehicles, service vehicles (SV), are equipped with small embedded computing units, that share their computational resources with all other network elements. The vehicular service provider deploys these units based on a reward system for participating cars. The VEoTC layer functions as a vehicular ad-hoc network (VANET), facilitating V2X communication by enabling vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. It provides similar shared resources as the VEC for its connected vehicles. Each SV can communicate with and serve a specific number of service requesters simultaneously. The task assignment process involves a service-based assignment problem, which determines the necessary level of cloud computing based on the required service. While the VEoTC can be allocated to any service based on its utilization, low latency takes precedence as the highest priority for being served by the VEoTC. The IEEE 802.1 family of standards provides the foundation for Dedicated, short-range communication between vehicles (V2V) and (V2I).

Determining the appropriate computing level to meet service requirements involves considering various factors, including the utilization of the computing layer, the end-to-end processing time required for the task, and others. The proposed three-tier cloud computing model within the VEoTC architecture presents a challenge for the network to determine the appropriate computing hierarchy required for various tasks. When a vehicle requests task offloading, nearby service vehicles will receive the request and decide whether to accept it based on factors such as their utilization and required quality of service (QoS). The SV will subsequently broadcast the response to all connected computing elements. If no acceptance message is received from any SV within a specific time window, the RSU will handle this request. Additionally, we propose deploying a Central unit in the network, tasked with monitoring the assignment of tasks and developing cooperative learning algorithms based on the collected data. This assignment process at each network element is reliant on artificial intelligence algorithms that utilize network features, request details, and SV information. Fig. 2 shows the computing resources of each computing tier, each tier has several servers C and waiting room capacity K. The problem is described as a birth-eath process with request arrival rate λ and μ serving rate.

The end-to-end delay of the service is calculated from the moment the service requester initiates a request until the completion time. The multi-tier cloud computing system encompasses various operational scenarios. Initially, when the requester initiates a request and seeks offloading, the first tier of computing, namely the VEoTC option, comes into play. Depending on the requested service, the VEoTC allocates priority to each task according to latency constraints. Priority within this computing tier is primarily accorded to low-latency real-time services. Upon acceptance of offloading by the VEoTC, two sub-scenarios, that highly affects the end to end delay of the requested service, are taken into account. The two sub-scenarios depend on the dwelling time with respect to the end-to-end service time. Firstly, if the SV can accomplish the necessary computing task within the connectivity window with the requester and

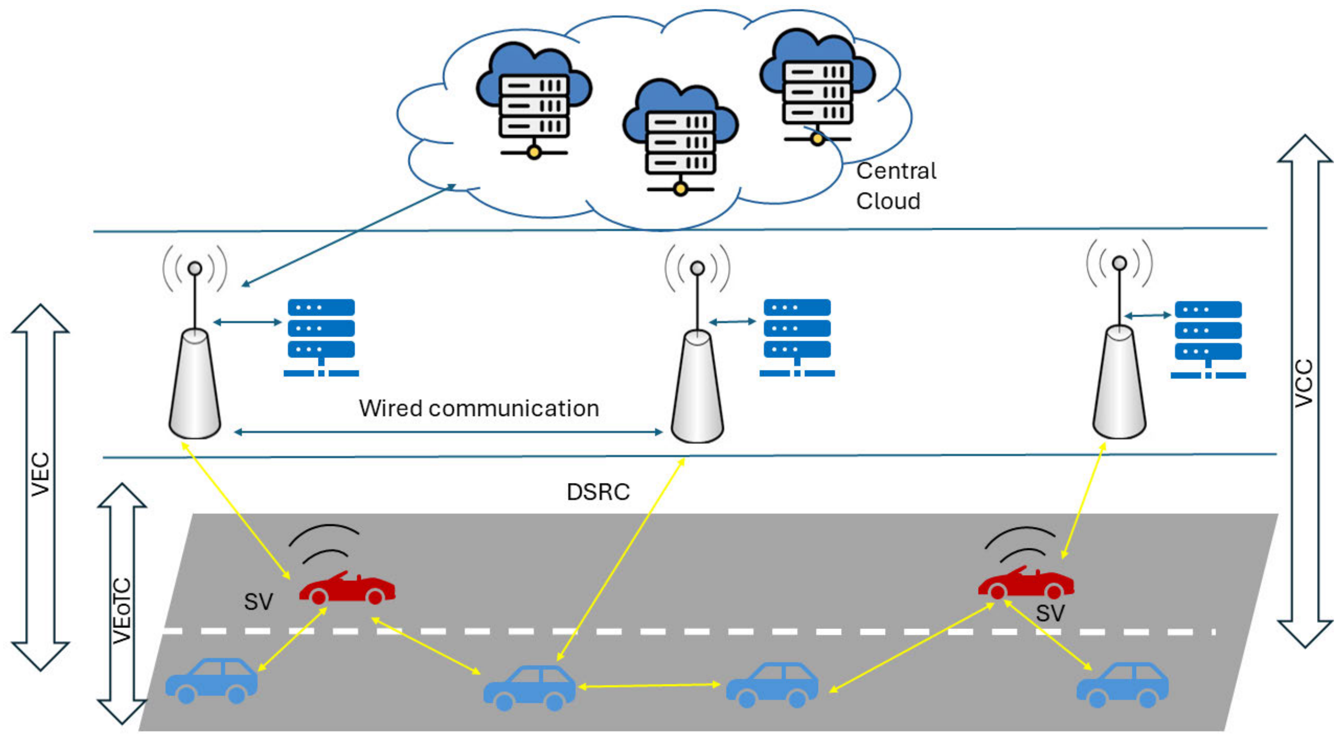

FIGURE 1. 3-Tier computing network architecture with VEoTC, VEC and CC with service vehicles (SV).

return it promptly, the end-to-end delay is characterized by three factors: the service uploading time, computing duration, and downloading time. Otherwise, if the connection between the SV and the requester is lost during the end-to-end service time, the SV must transmit the task upon completion to the nearest RSU, which then forwards it to the requester through the backhaul network. If the request is rejected by the VEoTC tier, the second computing level using the VEC will be considered. In this scenario, the end-to-end service time comprises the same three components as in the case of VEoTC. The primary enhancement offered by the proposed VEoTC layer is overcoming the limited coverage area of the RSU and addressing the high mobility of the requester. In the VEoTC setup, the SV is considered relatively stable concerning the requester since they are in the same velocity frame. This resolves the problem of mobility disparity between the computing unit and the service requester. Additionally, ensuring stable coverage areas for both vehicles takes higher priority during the service time, especially for services requiring short computing time. If the computing assignment fails at both computing levels, the task will ultimately be offloaded to the central cloud computing. The computing tier assignment procedure is described by the offloading process flow chart in Fig. 3.

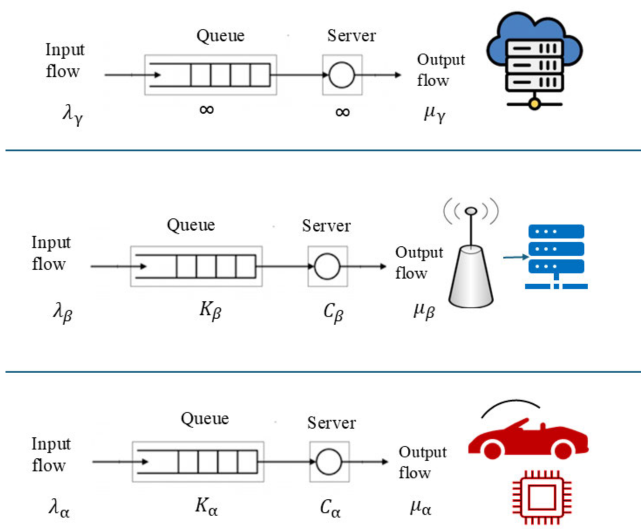

FIGURE 2. Computing resources in service vehicle (SV), RSU, and the cloud.

III. PROBLEM FORMULATION

In this section, we present a probability model for analyzing the performance of task offloading and distributing tasks among VEoTC, VEC, and CC. Subsequently, we outline the End-to-End processing time at each level. Based on this information, we demonstrate how each level impacts the anticipated system latency. We describe the problem as a discrete probability problem with three distinct, non-overlapping, and independent events. Offloading tasks to each computing tier is considered as an event, where

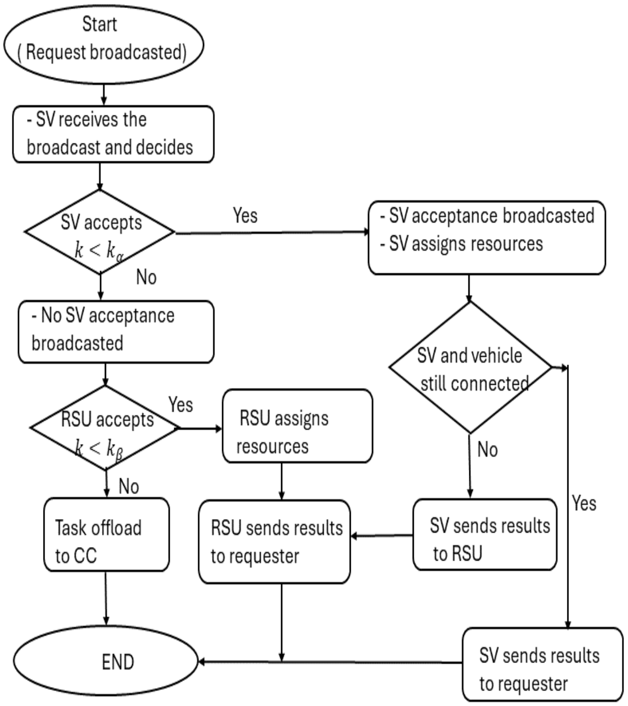

FIGURE 3. 3-Tier offloading process flow chart.

offloading to VEoTC, VEC, and CC represent events α, β, γ, respectively. The computing layers are structured hierarchically, with ascending levels offering increased processing capabilities, faster processing speeds, higher clock frequencies, and greater numbers of available processors.

In VEoTC layer each service vehicle is assumed to have compact embedded processing unit with multiple core that can serve up to N requester, simultaneously. All the communications are performed with DSRC standards, over wireless channels. The offloading process is described a Poisson process, as VEoTC implements $M/M/C_\alpha/K_\alpha$ queue model with C_α servers and clock frequency (exponential service rate) F_α, and arrival rate λ_α. Upper tiers of computing are characterized by queuing models featuring larger capacities and additional faster processing units. The VEC layer is deployed with C_β processing units, λ_β arrival rate, and service rate (clock frequency) F_β. The VEC is simulated $M/M/C_\beta/K_\beta$ queuing model. Finally, the center cloud represents infinite computing capabilities, λ_γ arrival rate, F_γ processing clock frequency, and simulated by $M/M/\infty$ queuing model. The VEoTC and the VEC computing are represented as finite queue, to limit the task waiting time. Each computing tier can have up to K_α and K_β serving and waiting tasks respectively.

Upon a user's request for task offloading, a broadcast message is received by both the Roadside Unit (RSU) and the connected SV. Depending on the task's priority and the SV's current utilization, the SV replies to the request with either acceptance or refusal. If no SV accepts the task within a predetermined time interval, the RSU undertakes

TABLE 1. Abbreviations.

	VEoTC(α)	VEC(β)	CC(γ)
Arrival rate	λ_α	λ_β	λ_γ
Service rate	F_α	F_β	F_γ
Queue capacity	K_α	K_β	∞
Number of servers	C_α	C_β	C_γ
Task rejection probability	$P_{r\alpha}$	$P_{r\beta}$	-
Serving probability (Queue)	$P_{s\alpha}$	$P_{s\beta}$	-
Task offloading probability	P_α	P_β	P_γ
End-to-End time	F_α	F_β	F_γ
Number of cycles	N_α	N_β	N_γ

the task. We investigate the end-to-end service time under two scenarios, determined by the dwelling time. Dwelling time is defined as the duration of the connection between the SV and the user. If the end-to-end service time is shorter than the dwelling time, the SV can execute the task and return it to the user. Conversely, if the end-to-end service time exceeds the dwelling time, the SV must complete the task and forward it to the connected RSU for delivery to the user.

A. END-TO-END SERVICE TIME

In this section, we provide a mathematical model for the end-to-end service time in each computing tier. The expected end-to-end service time across three possibilities of computing is given by (1)

$$E(T) = P_\alpha * T_\alpha + P_\beta * T_\beta + P_\gamma * T_\gamma \tag{1}$$

where P_α, T_α are the probability of vehicle computing and end-to-end service time of VEoTC. Similarly, for VEC and CC probabilities and end-to-end service time.

As mentioned in [22], the primary factors affecting the end-to-end service duration include the front-haul transmission delay, computing delays (VEoTC, VEC, CC), the back-haul transmission delay, and the queuing delays. The front-haul transmission delay depends on the transmission rate (R_F), determined according to Shannon rule as in (3)

$$R_n^f = B_n^f * \log_2(1 + SINR^f) \tag{2}$$

where R_n^f, B_n^f are the transmission rate of user n in the front-haul connection with DSRC standards, and the bandwidth assigned to user n, respectively. and $SINR^f$ is the signal to interference plus noise ratio in the front-haul network. Accordingly the uploading time of sending D bits from user n to the VEC is given by

$$T_n^{uf} = \frac{D_n}{R_n^f} = \frac{D_n}{B_n^f * \log_2(1 + SINR^f)} \tag{3}$$

While the back-haul transmission delay is assumed to fixed over wired communication network and denoted by T^{ub}.

The computing time of each cloud is defined by the processing time and the queuing time, as in the VEoTC layer the average waiting time of $M/M/C$ queue model according to the queuing theory

$$T_w = \frac{P_o(\frac{\lambda}{\mu})^c \rho}{c!(1-\rho)^2} \tag{4}$$

where P_o, ρ are the probability that there are 0 customers in the system, and the utilization of the server, respectively, and are given by

$$P_o = \frac{1}{\sum_{m=0}^{c-1} \frac{(c\rho)^m}{m!} + \frac{(c\rho)^c}{c!(1-\rho)}} \tag{5}$$

$$\rho = \frac{\lambda}{c\mu} \tag{6}$$

While the average computing time of offloading task is

$$T_{proc} = \frac{1}{\mu} \tag{7}$$

Hence, the end-to-end task of the VEoTC T_α is given by

$$T_\alpha = \frac{N_\alpha}{F_\alpha} + \frac{P_{o\alpha}(\frac{\lambda_\alpha}{F_\alpha})^{C_\alpha}\rho_\alpha}{C_\alpha!(1-\rho_\alpha)^2} + 2 * \frac{D_n}{B_n^f * \log_2(1+SINR^f)} \tag{8}$$

While the end-to-end time for the VEC offloading has two cases first if the assigned RSU can complete the task, the T_β is given by

$$T_\beta = \frac{N_\beta}{F_\beta} + \frac{P_{o\beta}(\frac{\lambda_\beta}{F_\beta})^{C_\beta}\rho_\beta}{C_\beta!(1-\rho_\beta)^2} + 2 * \frac{D_n}{B_n^f * \log_2(1+SINR^f)} \tag{9}$$

otherwise, if the vehicle is not in the coverage area of the assigned RSU, then the RSU will forward the task after completion to the back-haul to forward it to the nearest RSU. Deploying RSUs within the VCN presents challenges, with a high cost associated with deploying a large number of fully connected RSUs. Conversely, opting for a smaller quantity of RSUs leads to an increase in the End-to-End time during handover, as the RSU must relay data to the CC, which then forwards it to the new RSU [23]. This paper explores the deployment of a reduced number of RSUs to minimize system costs. In such scenarios, the handover overload is equivalent to transmitting the data to the CC twice.

$$T_\beta = \frac{N_\beta}{F_\beta} + \frac{P_{o\beta}(\frac{\lambda_\beta}{F_\beta})^{C_\beta}\rho_\beta}{C_\beta!(1-\rho_\beta)^2} + 2 * \frac{D_n}{B_n^f \times \log_2(1+SINR^f)} + H \times T^{ub} \tag{10}$$

where H represents the number of RSU handovers during the task completion time.

Finally, if the offloading to the VEoTC and VEC failed the task will be offloaded to the CC, and the time T_γ is given by the processing and the front-haul and the back-haul transmission times as no waiting delays for infinite number of processors

$$T_\gamma = \frac{N_\gamma}{F_\gamma} + 2 * \frac{D_n}{B_n^f * \log_2(1+SINR^f)} + 2 * T^{ub} \tag{11}$$

B. TASK ARRIVAL RATES

Similarly, the model characterizes each computing tier is a queuing model with distinct arrival rates, service rates, and server quantities. The arrival rate at the VEC, which is determined by the number of vehicles rejected by the VEoTC tier and decided to offload tasks to the VEC, is given by.

$$\lambda_\beta = \lambda_\alpha \times (1 - P_\alpha) \tag{12}$$

where $P_{r\alpha}$ represents the probability of rejecting task offloading in VEoTC tier Similarly, The arrival rate at the CC, which is determined by the number of vehicles rejected by the VEoTC and VEC tier and decided to offload tasks to the CC, is given by.

$$\lambda_\gamma = \lambda_\beta \times (1 - P_{s\beta}) \tag{13}$$

where $P_{s\beta}$ is the probability of serving the offloaded task by the VEC.

C. OFFLOADING PROBABILITY

In this section, we provide the offloading probability analysis in VEoTC, VEC, and CC. The probability of offloading in each tier, depends on the rejection rate from the previous tier and the computation capability of the tier itself.

$$\begin{aligned} P_\alpha &= P(offload_{SV}) \times P_{s\alpha} \\ &= P_{SV} \times P_{s\alpha} \end{aligned} \tag{14}$$

The probability of encountering a service vehicle, denoted as P_{SV}, is established by the service provider. While, $P_{s\alpha}$ represents the likelihood of discovering an available spot in the VEoTC queue to perform the offloaded task, and it is given by

$$\begin{aligned} P_{s\alpha} &= P(N < K_\alpha) \\ &= \sum_{k=0}^{C_\alpha - 1} P_{o\alpha}\frac{(C_\alpha\rho_\alpha)^k}{k!} + \sum_{k=C_\alpha}^{K_\alpha - 1} P_{o\alpha}\frac{C_\alpha^{C_\alpha}}{C_\alpha!}\rho_\alpha^k \end{aligned} \tag{15}$$

When the queue is at maximum capacity, the system denies access to all users. For a task to offload to the VEoTC tier, the total number of tasks in the VEoTC queue at time "t" must be below K_α.

The probability P_β governs the likelihood of offloading to VEC in two scenarios: either when there is no available SV or when there are insufficient resources within the assigned SV.

$$\begin{aligned} P_\beta &= P(offload_{VEC}) \times P_{s\beta} \\ &= \big((1 - P_{SV}) + P_{SV}(1 - P_{s\alpha})\big)P_{s\beta} \\ &= (1 - P_{SV}P_{s\alpha})P_{s\beta} \end{aligned} \tag{16}$$

Similarly, $P_{s\beta}$ indicates the probability of encountering an empty position in the VEC queue to perform the offloaded task, and its value is

$$\begin{aligned} P_{s\beta} &= P(N < K_\beta) \\ &= \sum_{k=0}^{C_\beta - 1} P_{o\beta}\frac{(C_\beta\rho_\beta)^k}{k!} + \sum_{k=C_\beta}^{K_\beta - 1} P_{o\beta}\frac{C_\beta^{C_\beta}}{C_\beta!}\rho_\beta^k \end{aligned} \tag{17}$$

Finally, the probability of offloading tasks to the CC tier, is the probability of rejecting the task from offloading to the VEoTC and VEC, so it can be written as

$$P_{\gamma} = 1 - P_{\alpha} - P_{\beta} \tag{18}$$

IV. PERFORMANCE EVALUATION

In this section, we provide the simulations results of the service time analysis detailed in section II. The simulations are divided into two parts: firstly, an examination of the dwell time of the RSU and the service vehicle utilizing actual data; and secondly, an evaluation of the service time based on the computed dwell time derived from real-world data. The authentic vehicular trajectories are sourced from the Didi Chuxing GAIA Initiative for Chengdu city, China, and were extracted on November 16th, 2018, serving as traffic inputs [24].

The GAIA data, processed by [25], offers the latitude and longitude of 360 vehicles across 24 hours divided into 300 time samples (sample at each 5 minutes). Initially, we designate a specific ratio of vehicles as Service Vehicles (SVs) equipped with minimal intelligent processing capabilities. Subsequently, we create a cluster around each SV, with the SV serving as the cluster center and including all nearby vehicles. At every time step, these clusters update in response to the movements of the vehicles. Ultimately, we calculate the average dwell time for each SV and all connected vehicles. The overall average dwell time for all SVs is computed by averaging the assigned SVs' dwell times individually. The average dwell time for 36 Service Vehicles (SVs), representing 10 percent of the total available vehicles, is computed using Jupyter. The average dwell time for deploying 36 SVs is computed, alongside the average dwell time for the RSU. This comparison underscores the improvement achieved by deploying SVs as mobile computing units in maintaining connectivity with service requesters. Since the mobile SVs are relatively stationary compared to other vehicles within the same speed frame, this enhances connectivity continuity. The algorithm randomly selects several SV (from GAIA data set) to form multiple clusters, as a cluster center, then assigns all nearby vehicles within the SV communication range to this cluster. This process is depicted through spatial clustering, which relies on calculating the Euclidean distance between the SV and all neighboring vehicles. This spatial clustering is repeated at every time stamp. When a vehicle moves beyond the communication range of the cluster center, the dwell time is calculated throughout the connectivity period.

Alternatively, the average dwell time for both the vehicle and the RSU is calculated as the duration it takes for each vehicle to traverse a distance of 1 kilometer, corresponding to the RSU's coverage area. The dwell time is heavily influenced by the average speed of the vehicle. In this context,, Table 2 provides recorded data on the average vehicle speeds at various times, including both rush and off-peak hours. Accordingly, the average dwell time with the RSU is 2.5 Mins, alternatively, the average dwell time of the moving SV is 50 Mins. This strategy ensures reliable connectivity over a broader time frame, facilitating the delivery of real-time services.

TABLE 2. Real time data.

Time	Velocity	Arrival rate
8:00	18 Km/Hr	31.7 (Veh/ min)
12:00	21 Km/Hr	27.2 (Veh/ min)
14:00	18 Km/Hr	25.1 (Veh/min)
22:00	24 Km/Hr	12.7 (Veh/min)
Average	20 Km/Hr	25 (Veh/min)

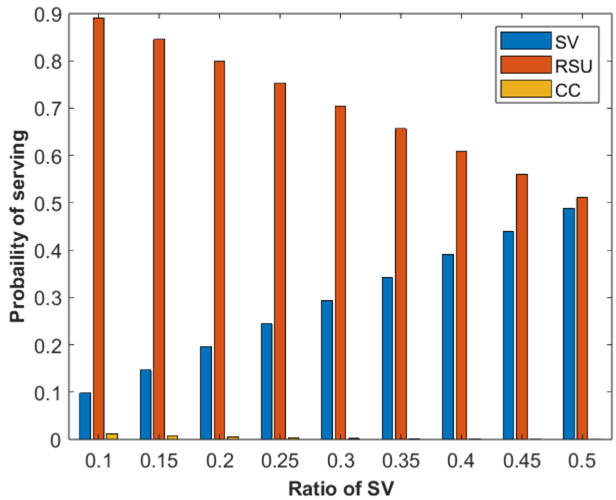

FIGURE 4. Probability of serving for each tier SV,RSU,CC with increasing the SV number.

In the subsequent phase of our investigation, we utilized Matlab simulations to examine the end-to-end service duration. The scenario of vehicle request arrival is conceptualized as a Birth-death process, where requests arrive at a rate of λ_{α}, and are assigned to the nearest SV contingent upon its queue capacity K_{α}. Requests that are declined due to SV unavailability or full utilization are redirected to the RSU at a rate of λ_{β}. Subsequently, all rejected requests from the RSU are forwarded to the Central Controller (CC) for processing. The services required are standardized and necessitate N_{α} time slots of Giga CPU cycles for execution. These tasks are deemed real-time services, demanding high connection reliability and consistent connectivity throughout the service period.

All pertinent simulation parameters are outlined in Table 3. It is assumed that both the RSU and the SV possess identical processing capabilities in terms of CPU clock frequency and the number of cores. Each device operates with a task processing rate of μ_{α} and μ_{β} of 1.5 $tasks/min$, respectively, with 8 cores facilitating parallel processing. The CC, on the other hand, is presumed to have an unlimited number of processors and a processing rate of $\mu_{\gamma} = 2.5\ tasks/min$. The transmission rate of tasks over the wireless network between

TABLE 3. Simulation parameter.

	VEoTC(α)	VEC(β)	CC(γ)
Arrival rate λ	5	2.55	0.001
Queue capacity k	4	4	∞
Number of servers C	8	8	∞
Task serving rate μ	1.5	1.5	2.5

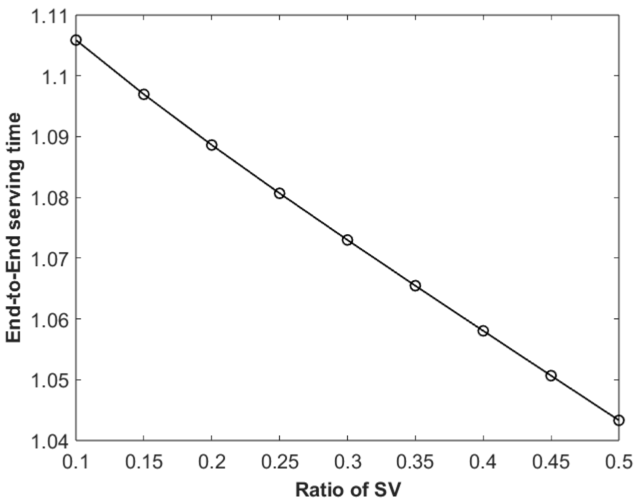

FIGURE 5. End-to-end serving time with increasing the SV number.

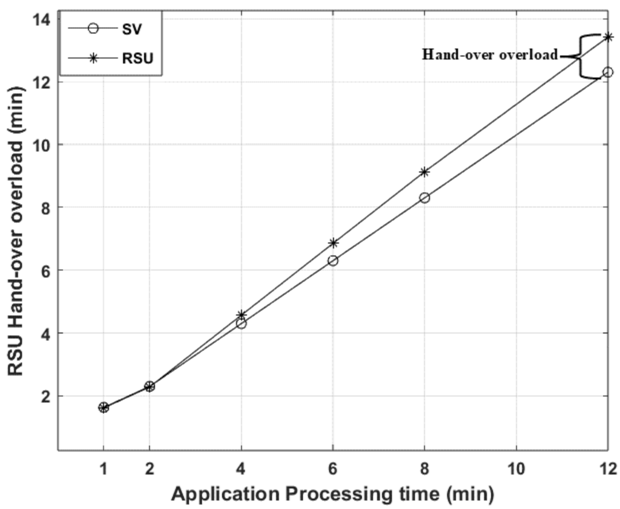

FIGURE 6. End-to-end serving time using SV, RSU offloading.

vehicles and the RSU is 7 $tasks/min$, while the migration rate to the CC and inter-RSU communication is 12 $tasks/min$.

Fig. 4 displays the probability of distributing requests across different tiers. As indicated, boosting the proportion of SVs equipped with computational capabilities reduces the back-haul overhead by prioritizing task offloading to available SVs.

Fig. 5 depicts the end-to-end service time under varying ratios of assigned SVs relative to the total number of vehicles. It is evident that augmenting the number of SVs enhances the

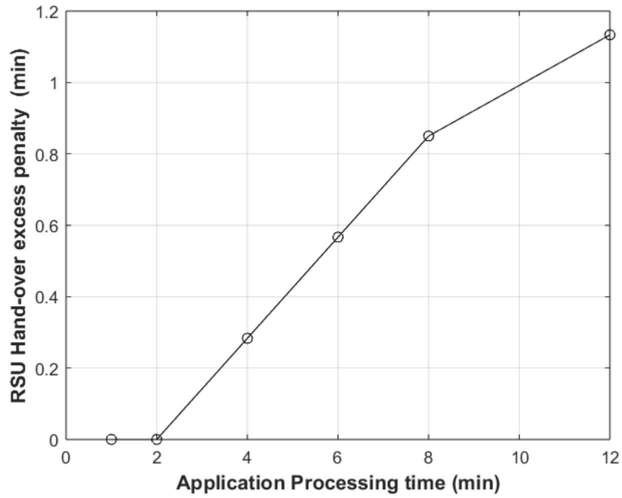

FIGURE 7. Excess penalty of offloading tasks to stationary RSU (Handover overload in RSU).

likelihood of task offloading to the SV tier, thus economizing back-haul processing time and reducing the overall end-to-end service duration.

Fig. 6 shows the End-to-End serving time with RSU and SV offloading cases, with different services. Light, moderate, and heavy processing applications are considered. Increasing the processing time of applications causes multiple RSU handovers, shown in Fig. 7, due to the mobility of the vehicles, while the SV consistent connection enhances the end-to-end time avoiding the handover overload.

V. CONCLUSION

In this study, a three-tier offloading system for VCN is explored, comprising VEoTC, VEC, and the CC. Probabilistic model analysis for the End-to-End serving time of 3-tier computing paradigm is introduced with a mobile computing capability, by deploying SV with computational resources to serve vehicles within its communication range. The previously proposed computing paradigm addresses the mobility aspect of VCN by ensuring a consistent relative speed between the requester and the moving edge computing, represented by the SV. We introduce a probabilistic model for the end-to-end serving time for task offloading to the VEoTC, VEC, and CC. In this model each tier is depicted by a queue with capacity K_α, K_β, and infinite queue length for the VEoTC, VEC, and CC, respectively. Similarly C_α, C_β, and infinite servers for the three computing levels. The task arrival and execution at the VEoTC are described as a Birth-Death process.

Using real data from [24], we calculated the dwell time between the randomly selected SV and the assigned requesters. The analysis revealed that the SV's dwell time significantly exceeds that of the RSU, attributed to the sustained relative speed between the SV and its connected vehicles. Next, we conducted simulations of the Birth-Death process for receiving requests. We examined the probability of service by each computing tier, the End-to-End serving time using both the RSU and the SV, and the additional penalty incurred due to handover using the stationary RSU computing tier. We demonstrated that offloading computationally intensive tasks with extended processing times to RSU substantially prolongs the End-to-End serving time due to frequent handovers. Conversely, offloading to SV, which provides a stable connection, effectively addresses the high mobility inherent in VCN, eliminating the need for handovers associated with stationary RSUs. Further intelligence should be integrated into the proposed SV to improve its connectivity. This entails considering additional requester features such as speed, destination, and service priority. Furthermore, the intelligent unit must analyze system variations using spatio-temporal real data to anticipate requests and allocate resources at specific times and locations.

REFERENCES

[1] Y.-J. Ku, S. Baidya, and S. Dey, "Adaptive computation partitioning and offloading in real-time sustainable vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 13221–13237, Dec. 2021.

[2] Y.-J. Ku, P.-H. Chiang, and S. Dey, "Real-time QoS optimization for vehicular edge computing with off-grid roadside units," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11975–11991, Oct. 2020.

[3] A. Waheed, M. A. Shah, S. M. Mohsin, A. Khan, C. Maple, S. Aslam, and S. Shamshirband, "A comprehensive review of computing paradigms, enabling computation offloading and task execution in vehicular networks," *IEEE Access*, vol. 10, pp. 3580–3600, 2022.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[5] L. U. Khan, I. Yaqoob, N. H. Tran, S. M. A. Kazmi, T. N. Dang, and C. S. Hong, "Edge-computing-enabled smart cities: A comprehensive survey," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10200–10232, Oct. 2020.

[6] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, and A. Neal, "Mobile-edge computing introductory technical white paper," *Mobile-Edge Comput.*, vol. 29, pp. 854–864, May 2014.

[7] C. F. C. Solutions. (2015). *Unleash the Power of the Internet of Tings*.

[8] F. Dai, G. Liu, Q. Mo, W. Xu, and B. Huang, "Task offloading for vehicular edge computing with edge-cloud cooperation," *World Wide Web*, vol. 25, no. 5, pp. 1999–2017, Sep. 2022.

[9] I. Jang, S. Choo, M. Kim, S. Pack, and G. Dan, "The software-defined vehicular cloud: A new level of sharing the road," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 78–88, Jun. 2017.

[10] Z. Jiang, S. Zhou, X. Guo, and Z. Niu, "Task replication for deadline-constrained vehicular cloud computing: Optimal policy, performance analysis, and implications on road traffic," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 93–107, Feb. 2018.

[11] A. Waheed, M. A. Shah, A. Khan, and G. Jeon, "An infrastructure-assisted job scheduling and task coordination in volunteer computing-based VANET," *Complex Intell. Syst.*, vol. 9, no. 4, pp. 3613–3633, Aug. 2023.

[12] A. Waheed, M. A. Shah, A. Khan, C. Maple, and I. Ullah, "Hybrid task coordination using multi-hop communication in volunteer computing-based VANETs," *Sensors*, vol. 21, no. 8, p. 2718, Apr. 2021.

[13] A. Amjid, A. Khan, and M. A. Shah, "VANET-based volunteer computing (VBVC): A computational paradigm for future autonomous vehicles," *IEEE Access*, vol. 8, pp. 71763–71774, 2020.

[14] K. Mershad and H. Artail, "Finding a STAR in a vehicular cloud," *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 2, pp. 55–68, Jun. 2013.

[15] H. Alshaer, T. Ernst, and A. de La Fortelle, "A QoS architecture for provisioning high quality in intelligent transportation services," in *Proc. IEEE Netw. Operations Manage. Symp.*, Apr. 2012, pp. 595–598.

[16] C. N. Tadros, M. Gad, B. M. Mokhtar, and M. E. Shimy, "Vehicular edge of thing computing for sustainable smart IoT services," in *Proc. IEEE 7th World Forum Internet Things (WF-IoT)*, Jun. 2021, pp. 55–59.

[17] Z. Liu, J. Weng, J. Ma, J. Guo, B. Feng, Z. Jiang, and K. Wei, "TCEMD: A trust cascading-based emergency message dissemination model in VANETs," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4028–4048, May 2020.

[18] Z. Liu, J. Weng, J. Guo, J. Ma, F. Huang, H. Sun, and Y. Cheng, "PPTM: A privacy-preserving trust management scheme for emergency message dissemination in space–air–ground-Integrated vehicular networks," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5943–5956, Apr. 2022.

[19] Z. Liu, L. Wan, J. Guo, F. Huang, X. Feng, L. Wang, and J. Ma, "PPRU: A privacy-preserving reputation updating scheme for cloud-assisted vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 2, no. 3, pp. 1–16, May 2024.

[20] Z. Liu, J. Ma, J. Weng, F. Huang, Y. Wu, L. Wei, and Y. Li, "LPPTE: A lightweight privacy-preserving trust evaluation scheme for facilitating distributed data fusion in cooperative vehicular safety applications," *Inf. Fusion*, vol. 73, pp. 144–156, Sep. 2021.

[21] Z. Liu, J. Guo, F. Huang, D. Cai, Y. Wu, X. Chen, and K. K. Igorevich, "Lightweight trustworthy message exchange in unmanned aerial vehicle networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2144–2157, Feb. 2023.

[22] J. Medhi, *Stochastic Models in Queueing Theory*. Amsterdam, The Netherlands: Elsevier, 2002.

[23] B. Lv, C. Yang, X. Chen, Z. Yao, and J. Yang, "Task offloading and serving handover of vehicular edge computing networks based on trajectory prediction," *IEEE Access*, vol. 9, pp. 130793–130804, 2021.

[24] *Data Source: Didi Chuxing Gaia Open Dataset Initiative*. [Online]. Available: https://outreach.didichuxing.com/research/opendata/en/

[25] X. Xu, K. Liu, P. Dai, F. Jin, H. Ren, C. Zhan, and S. Guo, "Joint task offloading and resource optimization in NOMA-based vehicular edge computing: A game-theoretic DRL approach," *J. Syst. Archit.*, vol. 134, Jan. 2023, Art. no. 102780.

ASMAA IBRAHIM received the bachelor's and master's degrees (Hons.) from Cairo University, Egypt, and the Ph.D. degree from UPC, Catalonia, Spain. She was an ESR with CTTC, Barcelona, Spain, in the framework of the Marie-Curie ITN 5G STEP-FWD Project; and a Teaching Assistant with AUC, Zewail, and the International Academy for Engineering and Media Science (IAEMS), Egypt. She is currently a Postdoctoral Fellow with UAEU. Her M.Sc. thesis focused on optimized resource allocation and interference management techniques in indoor optical communications. She has published several papers during her M.Sc. studies at international journals and conferences. Her research interests include 5G wireless communication networks, wireless optical communications, and visible light communications.

BASSEM MOKHTAR (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Alexandria University, Egypt, in 2004 and 2008, respectively, and the Ph.D. degree in computer engineering from Virginia Tech, USA. He is currently an Assistant Professor of computer engineering with the College of Information Technology, United Arab Emirates University, Al Ain, United Arab Emirates. He has experience in teaching undergraduate and graduate courses related to computer and electrical engineering and computer science. His research interests include embedding intelligence in networking operations, autonomic resilient networking, network semantics reasoning, and machine learning applications in computer networks. He has been involved in a set of funded multidisciplinary research projects integrating advances in fields of machine learning-based applications in sensor networks, nanotechnology, and flexible electronics. Also, he has participated as the PI, the Co-PI, and a Researcher in a set of funded research projects that cross multi-disciplinary research fields, including the Internet of Things, the Internet of Vehicles, software-defined networking, and flexible electronics. Additionally, he has been involved as a Reviewer for various conferences and journals, including IEEE INTERNET OF THINGS JOURNAL, *IEEE Vehicular Technology Magazine*, and *Journal of Network and Computer Applications* (Elsevier).