

GaMiCO: Game-slicing based Multi-interface Computation Offloading in 5G Vehicular Networks

Suhwan Jung, Hyoil Kim, Xinyu Zhang, and Sujit Dey

Abstract—Smart vehicles require constantly running heavy vehicular computations with their limited computation/energy resources. 5G vehicular networks have potential to resolve the issue, by letting the vehicular tasks offloaded to 5G mobile edge computing (MEC) servers. To better support vehicular computation offloading, this paper proposes a road-side 5G infrastructure consisting of multiple millimeter-wave (mmWave) small-cell base stations (BSs) and a cellular mid-band based macro-cell BS where each BS is equipped with an MEC server. Then, the vehicles with mmWave/mid-band dual interfaces can decide which BS to choose for offloading. We propose a decentralized offloading decision mechanism where each vehicle tries to minimize the time-energy joint cost with three choices: local computing, offloading to a small-cell MEC, offloading to a macro-cell MEC. In particular, we model the problem as an ordinal potential game, derive its potential function to ensure the existence of and finite-time convergence to a Nash equilibrium (NE), analyze its Price-of-Anarchy, and develop an iterative offloading decision update algorithm. In doing so, we also consider slicing the global game into multiple non-overlapping smaller games and running them in parallel, to investigate the best slicing strategy. Our extensive simulations show the game’s real-time convergence to an NE, reveal the NE’s near-optimal performance, and present the efficacy of the proposed game slicing.

Index Terms—5G, computation offloading, game slicing, millimeter wave, mobile edge computing, potential game, vehicular network

I. INTRODUCTION

Smart vehicles have emerged as an important trend in transportation, thanks to the recent advances in autonomous driving, artificial intelligence (AI), and electric vehicles. While vehicles’ computing power is consistently growing, it is not fast enough to handle diversified and heavy tasks such as processing exploding sensory data, driver monitoring, platooning, infotainment, etc. Energy consumption in computation is another critical factor due to the slow evolution of the battery technology. Therefore, efficiently processing computation-intensive vehicular tasks has become one of the most important problems [1], [2].

Manuscript received June 14, 2023; approved for publication, June 18, 2023. This paper is specially handled by EIC and Division Editor with the help of three anonymous reviewers in a fast manner.

This work was supported in part by the National Research Foundation of Korea (NRF) through the Korean Government (MSIT) under Grant 2020R1F1A1071726.

S. Jung and H. Kim are with the Department of Electrical Engineering, the Ulsan National Institute of Science and Technology (UNIST), Ulsan 44919, Korea, email: {fydnrltlzl, hkim}@unist.ac.kr.

Xinyu Zhang and Sujit Dey are with the Department of Electrical and Computer Engineering, University of California, San Diego, CA 92093, USA, email: xyzhang@ucsd.edu, dey@ece.ucsd.edu.

H. Kim is the corresponding author.

Digital Object Identifier: 10.23919/JCN.2023.000027

Computation offloading is a promising solution to the computation-intensive vehicular applications. By offloading local tasks to the resource-rich cloud servers via wireless channels, smart vehicles can process heavy tasks remotely while reducing their energy consumption. In case the cloud servers are located at public clouds (e.g., Amazon’s AWS, Google’s GCP, Microsoft Azure), however, offloading should suffer from high latency in accessing the servers. Hence, alleviating communication latency in offloading has been identified as a critical issue [3], [4].

Nowadays, 5G mobile edge computing (MEC) is expected to mitigate such latency effectively, by placing them at the mobile edge (thus closer to the vehicles). The cloud resources (henceforth called the MEC server) are usually installed at a base station (BS), where the BS can be either a macro cell BS (mBS) or a small cell BS (sBS). There is a tradeoff between mBS’s MEC server (mMEC) and sBS’s MEC (sMEC) server: the mMEC has more computing power but can be accessed through bandwidth-limited cellular bands, while the sMEC, possibly installed at roadside units (RSUs), has limited computing power (mainly due to its small form factor) but can be accessed via millimeter wave (mmWave) links with extensive bandwidth. As a result, 5G MEC based computation offloading introduces a unique challenge to smart vehicles: *whether to offload or not*, and *where to offload (mMEC vs. sMEC)*.

Existing work on vehicular task offloading mostly considered utilizing sub-6GHz mid-bands for the vehicle-to-sBS wireless link when offloading a task to the sMEC [5]–[8]. Moreover, offloading to the mMEC is assumed to be performed by transmitting the task to the sBS first and then forwarding it to the mMEC via an optical wired backhaul between the sBS and the mBS [5], [6], [9], [10]. Such backhaul-based forwarding, however, would entail a high implementation cost to deploy a backhaul per each small cell [11]–[13], and should endure additional non-trivial delays to transmit over the backhaul, including the conversion delay between electrical and optical signals, compared to the direct wireless access to sBSs and mBSs [14], [15].

Unlike the aforementioned approaches, this paper proposes a multi-interface computation offloading mechanism for 5G vehicular networks, where each vehicle is equipped with two wireless interfaces, mid-band and mmWave, and has three choices for a given task: local computing, offloading to an mMEC (henceforth called ‘mBS offloading’), and offloading to a sMEC (henceforth called ‘sBS offloading’). By offloading tasks wirelessly to both sMEC (via mmWave) and mMEC (via mid-band), we can solve the issue of implementation cost

Creative Commons Attribution-NonCommercial (CC BY-NC).

This is an Open Access article distributed under the terms of Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided that the original work is properly cited.

and delay incurred by the wired backhaul. In fact, leveraging mid-band and mmWave together has been also utilized by 5G NR to improve network coverage and data rate for mobile devices [16], [17]. Considering that nowadays 5G cellphones can support both midband and mmWave band with manageable chipset prices, we believe our proposal of dual-interface vehicular architecture is realistic.

Among the three offloading choices, local computing has the weakest computing power but does not incur any communication overhead. If a task is offloaded, however, sBS offloading incurs less communication overhead than mBS offloading thanks to larger channel bandwidth, but offers inferior computing power. Therefore, each vehicle should carefully decide what offloading option to choose so as to jointly minimize the overall latency and energy consumption of the offloaded computation.

In particular, we formulate the problem as an ordinal potential game, to realize distributed offloading decision by individual vehicles. Some existing work in the offloading literature have proposed centralized decision making methods considering multiple MEC server locations at small and macro cells [18], [19]. In the offloading context, however, it is known that centralized decision does not scale well with the number of vehicles, thus leading to an NP-hard problem [20]. On the other hand, distributed decision-making can ease the burden of centralized control, for which game theory is a powerful tool. Hence, we hereby adopt a game theoretic approach to model the strategic interactions between the vehicles.

Most game theoretic approaches for computation offloading in vehicular networks assumed a quasi-static scenario, where the set of vehicles and their channel conditions remain unchanged during a game period [20]–[23]. Such approaches, however, are not suitable for high-speed vehicle environments where the channel's coherence time reaches only about tens of milliseconds [24], [25]. Therefore, it is crucial to ensure the quasi-static assumption in any game theoretic formulation of vehicular scenarios by designing the game to converge within the coherence time. To do so, this paper proposes *slicing* the global game covering the whole road *into* smaller games (henceforth called 'game slices') each covering a non-overlapping road block, so as to run those games in parallel. In addition, we develop a method of determining the optimal number of game slices that minimizes the system-wise cost while satisfying the quasi-static condition.

The contributions of the paper are four-fold as follows.

- This paper proposes a novel multi-interface offloading architecture with mmWave-based small cells and a cellular mid-band-based macro cell, where a vehicle has three computation venues: local, sMEC, and mMEC. To the best of the authors' knowledge, this is the first attempt to consider dual MEC server locations with small cells operated by mmWave bands.
- The paper presents how to model inter-vehicle competition on offloading resources with a game theoretic framework. Specifically, the paper casts the problem into a potential game and proves the existence of Nash equilibrium (NE) by deriving the corresponding potential

function. In addition, we derive the lower/upper-bounds of the price of anarchy (PoA) for the game.

- The paper proposes game-slicing based multi-interface computation offloading (GaMiCO), a practical algorithm with which vehicles can participate in the offloading game in a distributed way. This paper analytically shows how fast the proposed iterative algorithm can converge to a NE. In addition, the dynamics of the game is shown via extensive simulations, e.g., how fast a NE is reached in practice, how good the NE is compared to the optimum, and how effective the game slicing is for minimizing the system-wide cost considering various vehicle arrival rates. Then, we compare the performance of the proposed algorithm with state-of-the-art game-theoretic offloading schemes.
- The paper evaluates the impact of game slicing on GaMiCO's game convergence time, and also shows that our optimal game slicing technique can satisfy the quasi-static assumption in practical vehicular offloading scenarios. To the best of the authors' knowledge, our work is the first paper that slices the global game into parallel ones to guarantee the quasi-static assumption in vehicular offloading environments.

The rest of the paper is organized as follows. Section II introduces related work, and then Section III describes our system model. Section IV formulates the multi-interface computation offloading potential game, and derive its potential function. Section V elaborates on the proposed offloading game algorithm, and analyzes the convergence of the algorithm. Section VI derives the lower and upper bounds of the PoA. Section VII evaluates the performance and dynamics of the proposed game, and the paper concludes with Section VIII.

II. RELATED WORK

A 5G network is built upon a multi-tier architecture composed of relays, small cells, and macro cells, designed to enhance spectral efficiency, coverage, and interference management [26]. Among the work dealing with MEC offloading in the 5G network [13], [15], [27]–[29], some studies treated small cells just as relays between mobile users and the macro cell's MEC server [13], [15], [27]. While [13] assumed a wireless backhaul between sBS and mBS, [15], [27] considered a wired backhaul between them. Others assumed that only small cells are equipped with MEC servers [28], [29], e.g., [28] proposing small cell cloud computing (SCC) and [29] considering the offloading decision problem to minimize per-user energy consumption. As opposed to the aforementioned works, we assume that MEC servers are placed at both sBS and mBS.

Regarding the work on dual MEC locations at small and macro cells (also referred to as the three-tier/layer architecture), [18], [19] have dealt with centralized optimal decision by a mBS for minimizing per-user cost or maximizing per-operator revenue. As mentioned in [20]–[22], however, decentralized offloading decision has several advantages than centralized decision, and thus we formulate the multi-interface offloading problem with game-theoretic approaches. In particular, the potential game has been considered as a technique

for decentralized computation offloading. In this vein, [20] formulated decentralized computation offloading in a multi-user environment as an ordinal potential game. [21] introduced a decentralized multi-user multi-channel offloading algorithm. [22] proposed a distributed multi-channel offloading algorithm that releases the channel resources of transmission-completed tasks for transmission-incomplete tasks. Aforementioned studies, however, considered macro cell's MEC server as the sole offloading destination.

Computation offloading has been treated as a powerful tool for the vehicular network. [9] analyzed the characteristics and the performance of vehicular Wi-Fi offloading scenarios. [10] proposed a theoretical model to characterize the performance of mmWave in a highway vehicular network with blockages. [30] formulated a mmWave beam design problem with position prediction based beam switching for vehicle-to-infrastructure communications. Aforementioned works, however, treated Wi-Fi access points or mmWave BSs just as relays with no computing resources. Since vehicular applications usually have stringent latency requirements, it is imperative to consider small cells as an offloading target.

In terms of dual-MEC computation offloading in vehicular networks, [5], [6] jointly dealt with offloading decision and MEC resource allocation considering the MEC servers at RSUs and at the remote cloud. While they adopted different types of game theoretic approaches for multi-vehicle offloading decisions, both assumed that offloaded tasks to the remote cloud should go through the wired backhaul between an RSU and the remote cloud, which causes non-negligible delay as discussed earlier in Section I. On the contrary, our dual-MEC architecture has an advantage in the sense that the mMEC (as an equivalent to the remote cloud) can be directly accessed by a vehicle via its own wireless interface.

Only few works [7], [8] have considered offloading to both of sMEC and mMEC via wireless links. [7] considered densely-deployed distributed small cells focusing on the impact of interference and limited resources of sBSs, and proposed a hierarchical genetic algorithm and a particle swarm optimization method to minimize the total energy consumption of all mobile devices. On the other hand, [8] studied the channel and computation resource allocation problem from the content provider's perspective, where the potential game is utilized to maximize the number of allocated mobile devices while minimizing the system cost.

To the best of the authors' knowledge, this paper is the first game theoretic study that tries to solve the issue of game convergence within channel coherence time in the vehicular offloading environment, by splitting the global game into smaller game slices.

III. SYSTEM MODEL

We consider a set $\mathcal{N} = \{1, 2, \dots, N\}$ of smart vehicles with computationally intensive tasks, which are geometrically covered by two types of BS, a single mBS and multiple sBSs. Each BS is equipped with its own MEC server, where the mBS's server is superior to the sBS's server in computation. We also assume the mBS utilizes 5G cellular spectrum under

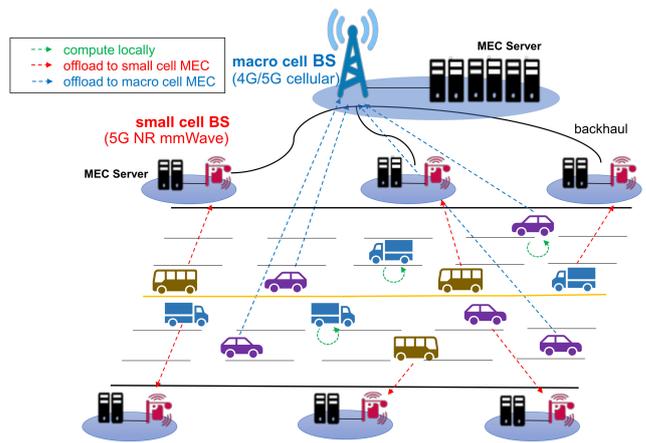


Fig. 1. Multi-interface computation offloading in 5G vehicular networks.

6 GHz whereas each sBS is a 5G mmWave BS equipped with a set of phased-arrays [16], [17], [31], [32]. In addition, each smart vehicle (which is a user equipment) is assumed to be equipped with two types of network interfaces (mid-band and mmWave) through which two simultaneous wireless connections with the mBS and a sBS are maintained. Our system model is depicted in Fig. 1.

The mmWave spectrum, ranging from 24 GHz to 300 GHz [33], suffers from higher attenuation loss than the mid-band spectrum. To compensate the loss, beamforming using high-gain phased array antennas has been considered for mmWave transmission. Furthermore, the array of phased arrays (APA) has been proposed to further extend the field-of-view of mmWave BS's coverage [34]. Assuming the hybrid beamforming architecture (i.e., multiple RF chains exist where each of them is connected to one phased array), we can also make a mmWave sBS maintain multiple beams simultaneously, each associated with a separate vehicle.

We assume periodic offloading where the mBS periodically announces the start of a new offloading epoch to all the vehicles within its coverage.¹ Then, the vehicles participate in the announced offloading game concurrently. In addition, the offloading epoch consists of four sequential steps: offloading decision (local vs. sMEC vs. mMEC), (if offloaded) uplink transmission of the task to sMEC or mMEC, offloaded computation at the MEC, and downlink transmission of the computation result to the vehicle. Note that we assume the downlink transmission time is negligible since the computation result is usually quite small [20], [21].

We assume \mathcal{N} does not change during an offloading period (i.e., a quasi-static scenario), with little loss of generality as justified below. To guarantee the quasi-static scenario, two conditions must be met — the channel state should not change until the tasks are uplink transmitted, and handover should not occur until the computed results are transmitted through the downlink — which can be satisfied as follows. First, Section VII-C will show that with our game slicing, the tasks

¹Considering that various delay-sensitive tasks generate and collect information periodically [35], [36], we assume that the game also occurs periodically. In the meantime, we reserve it as our future work to determine the optimal game period by considering practical task-related factors such as queuing, task arrival rate, etc.

can be uplink-transmitted within the channel coherence time of both mBS and sBS. For instance, for a vehicle moving at 80 km/h, the channel coherence time is obtained as 34.38 ms for mBS [37] and 40.95 ms for sBS [25].² In the meantime, our slicing-based game consumes at most 32.89 ms until the completion of uplink transmission, thus less than both coherence times, while the existing game-theoretic approaches may exceed the coherence times. Second, Section VII-F will also show that only up to 1.82% of the vehicles would cross a small or macro cell's edge until the downlink transmission completes, suggesting that the quasi-static assumption becomes valid if such cell-edge located vehicles are guided not to participate in the game but to compute locally.

We consider that vehicle n has a task to offload which runs with input data of b_n bits and requires c_n CPU cycles for computation. Then, we denote by $a_n \in \{0, 1, 2\}$ the offloading decision of vehicle $n \in \mathcal{N}$, where $a_n = 0$ means local computing, $a_n = 1$ means sBS offloading, and $a_n = 2$ means mBS offloading. We also denote by $\mathbf{a} = (a_1, a_2, \dots, a_N)$ the decision profile of all the vehicles.

We also assume that the offloaded task's code already exists at MEC servers, which is achievable since the set of offloadable vehicular tasks should be known in advance and thus their codes can be pre-installed at MEC servers by the offloading service provider. Regarding the communication latency between a vehicle and a BS, we consider uplink transmission time as a major factor due to potentially heavy offloading data, while assuming that downlink transmission time is negligible.

A. Game Slicing Model

As mentioned in Section I, we propose slicing one global offloading game (that covers the entire road) into parallel non-overlapping games, each called a 'game slice'. Each slice belongs to the set $\mathbb{G} = \{1, 2, \dots, N_{GS}\}$ with N_{GS} game slices. In addition, each vehicle belongs to the slice covering its current location on the road, and thus slice g includes a set of vehicles \mathcal{N}_g and $\sum_{g=1}^{N_{GS}} |\mathcal{N}_g| = N$. We also assume that the mBS's channel bandwidth, denoted by B_m ,³ is distributed to N_{GS} game slices such as $\sum_{g=1}^{N_{GS}} B_m^g = B_m$, where B_m^g is the macro cell channel bandwidth allocated to game slice $g \in \mathbb{G}$. Note that Section III-C will explain how B_m^g is allocated to the vehicles in the slice, while Section VII will evaluate the impact of game slicing on minimizing the system-wide cost.

Fig. 2 shows how the global game can be sliced into N_{GS} game slices in a real road environment, where each red box represents the area a single game slice covers. For instance, Fig. 2(a) addresses a scenario where $N_{GS} = 1$, indicating that all vehicles on the road participate in the single global game. Note that we tried to evenly distribute the traffic load into the multiple slices, by making per-slice area the same. Although

²Note that the vehicle moving at 80 km/h (equivalently 22.22 m/s) would associate with its sBS via an allocated beam for 63 ms. Specifically, assuming the beamwidth is 10° for the maximum beam gain [38] and the 8 m high sBS is right above the road, the length of the road section covered by the beam becomes $8 \times \tan(5^\circ) \times 2 = 1.4$ m, and thus $1.4/22.22 \times 10^3 = 63$ ms.

³ B_m can be also thought as the reserved portion of the cellular bandwidth for offloading services.

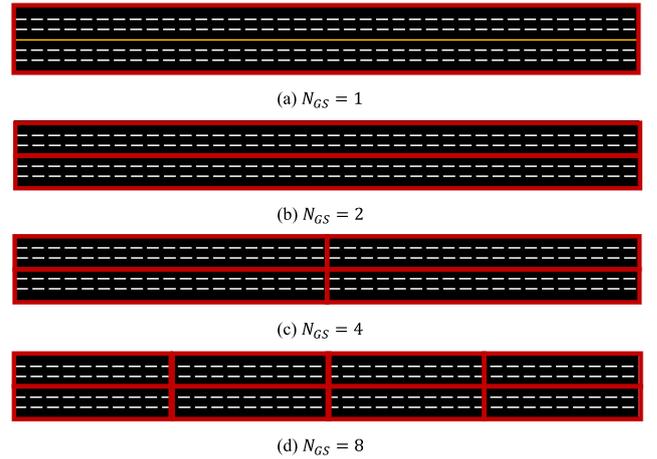


Fig. 2. Game slicing model.

this paper assumes N_{GS} is an exponential power of 2 for the ease of visualization, how it is sliced can be determined differently by the offloading service providers.

B. Communication Model: MmWave sBS

We assume that each vehicle can be assigned its own separate line-of-sight (LoS) beam from its associated sBS by considering the following deployment scenario: 1) sBS are installed at a high position and at the both sides of the road, 2) sBS are densely deployed (by nature) and there exist a large number of phased arrays per sBS. We believe such a scenario reflects the reality to some extent since sBS can be installed at tall streetlamps (following the RSU deployment scenario in the 3GPP TR 37.885 which considers the antenna height of 5 meters), and the number of phased arrays per mmWave BS is up to eight nowadays [39] and tends to increase. According to the experimental evidence in [40], the probability of a link suffering from LoS blockage when assuming the 5 meter sBS height is only 2% in the highway scenario. The paper also showed that more than 99% of blockages disappear by increasing the BS height to 10 meters, regardless of deployment scenarios. Based on this evidence, our orthogonal beam assignment through the LOS path seems reasonable.

By the assumption, a vehicle's uplink transmission to its sBS is not affected by other vehicles' transmissions, and hence the uplink data rate r_n^s of vehicle n (with $a_n = 1$) is given as

$$r_n^s = B_s \cdot \log_2(1 + SNR_n^s), \quad (1)$$

where B_s is the mmWave channel bandwidth and SNR_n^s is the signal to noise ratio (SNR) measured at the sBS of vehicle n . To obtain SNR_n^s , we introduce the beamforming antenna gains [38] to the omnidirectional large-scale path loss model for a 28 GHz LOS channel in [41], such as

$$PL_{28\text{GHz}}^{\text{LOS}}(d)[\text{dB}] = (61.4 - (G_{TX} + G_{RX})) + 21 \log(d) + \chi_\sigma, \quad (2)$$

where d (m) is the distance between vehicle n and the sBS, χ_σ is normally distributed with zero mean and standard deviation of $\sigma = 3.6$ (dB), and G_{TX} and G_{RX} (both in dB) are

beamforming antenna gains at the transmitter and the receiver, respectively.⁴ Note that G_{TX} and G_{RX} may depend on the mmWave beam pattern (e.g., beam width).

C. Communication Model: Cellular mBS

We assume each vehicle desires to finish their uplink transmission to the mBS within t_d ms, where t_d ms is a design parameter according to the target quality of service (QoS) or quality of experience (QoE) level of the network operator [43]–[45]. Furthermore, we assume ‘ t_d ms’ is an elastic deadline, i.e., it is a preference but not a requirement for vehicles. Then, we can calculate the necessary amount of bandwidth W_n for vehicle n to complete the uplink transmission in time, such as

$$W_n = \frac{10^3 \cdot b_n}{t_d \cdot \log_2(1 + SNR_n^m)}, \quad (3)$$

where b_n is the offloading data size and SNR_n^m is the SNR observed at the mBS.⁵ To obtain SNR_n^m , we can refer to the LTE suburban/rural macro channel model [46].

Since B_m^g , mBS’s channel bandwidth assigned to slice g , should be shared by the vehicles belonging to the slice, we consider proportional resource allocation where the allocated bandwidth of vehicle $n \in \mathcal{N}_g$ is given as

$$B_m^g \cdot \frac{W_n}{W_n + \sum_{i \neq n, i \in \mathcal{N}_g} W_i \cdot I_{(a_i=2)}}, \quad (4)$$

where $I_{(x)}$ is an indicator function that returns 1 if the event x is true and 0 otherwise. The intuition of proportional bandwidth allocation is that offloading users are sensitive to latency and thus it is important to be fair in providing the same level of uplink transmission delay to the macro cell users.

Finally, we calculate the uplink data rate of vehicle n as

$$r_n^m(\mathbf{a}^g) = \frac{B_m^g \cdot W_n}{W_n + \sum_{i \neq n, i \in \mathcal{N}_g} W_i \cdot I_{(a_i=2)}} \cdot \log_2(1 + SNR_n^m), \quad (5)$$

where \mathbf{a}^g is the decision profile of the vehicles in slice g .

D. Computation Model: Local Computing

When a vehicle chooses local computing, there are two kinds of cost involved, computational latency and energy consumption in computation. Denoting by f_n^l (cycles/second) the computing power of vehicle n , the computational latency t_n^l is derived as

$$t_n^l = \frac{c_n}{f_n^l}. \quad (6)$$

Next, denoting by ϵ_n (J/cycle) the energy consumption per CPU cycle, the energy consumption e_n^l is given as

$$e_n^l = \epsilon_n \cdot c_n. \quad (7)$$

⁴The Doppler effect has been assumed to be mitigated by utilizing Doppler shift compensation techniques as proposed in [10], [42].

⁵In reality, the offloading data size also includes extra bits for constructing a 4G/5G frame, e.g., header size. Since we consider relatively large b_n (e.g., uniformly distributed in [10, 30] KB) than such overhead, we ignore the extra bits in this formulation. Including them in the analysis, however, is straightforward.

Using (6) and (7), the local computing cost Z_n^l is given as

$$Z_n^l = e_n^l + \lambda \cdot t_n^l = c_n \cdot \left(\epsilon_n + \frac{\lambda}{f_n^l} \right), \quad (8)$$

where $\lambda \in [0, 1]$ is the weighting factor between time and energy, to capture a vehicle’s preference between them.⁶ As λ gets closer to 1, vehicles care more about time than energy. Note that λ needs to be well-tuned not only to make a good balance between energy and delay, but also to compensate their difference in magnitude. In such a vein, Section VII will evaluate the influence of λ by varying its value.

E. Computation Model: Offloading to sBS’s MEC (sMEC)

Unlike local computing, offloading to sBS or mBS should deal with additional overhead in uplink transmission to derive the overall cost. Using (1), we can calculate the time overhead $t_n^{s,o}$ incurred by uplink transmission to a sBS as

$$t_n^{s,o} = b_n / r_n^s. \quad (9)$$

Denoting by P_n^s (W) the uplink transmit power of vehicle n to its sBS, the energy consumption $e_n^{s,o}$ due to transmission is derived as

$$e_n^{s,o} = P_n^s \cdot b_n / r_n^s. \quad (10)$$

Next, we need to calculate the computing time at the sMEC. denoting by f_s the computing power of the sMEC, the computational latency $t_n^{s,c}$ of the offloaded task is given as

$$t_n^{s,c} = c_n / f_s. \quad (11)$$

Finally, using (9), (10), and (11), the sBS offloading cost Z_n^s is derived as

$$Z_n^s = e_n^{s,o} + \lambda \cdot (t_n^{s,o} + t_n^{s,c}) = \frac{b_n}{r_n^s} \cdot (P_n^s + \lambda) + \lambda \cdot \frac{c_n}{f_s}. \quad (12)$$

As previously mentioned, we have ignored the downlink transmission time for receiving the processing result of an offloaded task.

F. Computation Model: Offloading to mBS’s MEC (mMEC)

The derivation of the mBS offloading cost is very similar to that of sBS offloading, except that mBS’s communication and computation parameters should be used. First, let us denote by f_m the computational capability of the mMEC, and by P_n^m the uplink transmit power of vehicle n to the mBS. Then, using (3) and (5), we can derive the overheads of mBS offloading similarly to the sBS case, such as

$$t_n^{m,o} = \frac{b_n}{r_n^m(\mathbf{a}^g)}, \quad e_n^{m,o} = P_n^m \cdot \frac{b_n}{r_n^m(\mathbf{a}^g)}, \quad t_n^{m,c} = \frac{c_n}{f_m}, \quad (13)$$

where $t_n^{m,o}$ denotes the time overhead incurred by uplink transmission to the mBS, $e_n^{m,o}$ denotes the energy consumption due to transmission, and $t_n^{m,c}$ denotes the computational latency of

⁶Before applying the weighted sum of energy and time, they can be individually normalized for unitless combining. Such an approach, however is fundamentally the same as ours in (8), as the former is only a scaled version of the latter.

the offloaded task. Finally, we can derive the mBS offloading cost Z_n^m using (13), such as

$$Z_n^m = e_n^{m,o} + \lambda \cdot (t_n^{m,o} + t_n^{m,c}) = \frac{b_n(P_n^m + \lambda)}{r_n^m(\mathbf{a}^g)} + \frac{c_n \lambda}{f_m}. \quad (14)$$

IV. POTENTIAL GAME FORMULATION

In this section, we present how to formulate the distributed offloading decision making problem with game theoretic analysis. We first formulate it as a strategic game, and then show that the game is a potential game by deriving its potential function. Note that potential games are known to guarantee the existence of an NE and finite-time convergence to the NE.

By the communication model in Section III, a vehicle's local computing cost and sBS offloading cost are not affected by other vehicles' decisions. Therefore, each vehicle can initially compare its local computing cost with its sBS offloading cost, to decide which one of them is better (i.e., smaller). Then, the vehicle compares the chosen option (i.e., local or sBS offloading) with mBS offloading to make the final decision. As discussed earlier, the mBS offloading cost is influenced by other vehicles in the same slice due to the proportional resource allocation mechanism in (4); as more vehicles offload to mBS, allocated bandwidth per vehicle decreases and in turn per-vehicle data rate is impaired, leading to an aggravated offloading cost in (14). Hence, the vehicles choosing mBS offloading should participate in the mBS offloading game, until when no more vehicles want to deviate from their decisions. In this vein, Section V will propose an iterative decision update algorithm to guarantee distributed decision making and convergence.

A. Game Formulation as a Strategic Game

The strategic game is a non-cooperative decentralized game [47], for which we need to define the set of players, per-player strategy, and per-player utility (or cost) function. In game slice g , \mathcal{N}_g is the set of players, and the decision of player $n \in \mathcal{N}_g$ is given as $a_n \in \mathcal{A}_n^g = \{0, 1, 2\}$ where \mathcal{A}_n^g represents the player's set of strategies. Then, the strategy space \mathbb{S}^g is given as $\mathbb{S}^g = \mathcal{A}_{\mathcal{N}_g(1)}^g \times \mathcal{A}_{\mathcal{N}_g(2)}^g \times \cdots \times \mathcal{A}_{\mathcal{N}_g(|\mathcal{N}_g|)}^g$, where each element of \mathbb{S}^g is called a strategy profile [48]. Note that $\mathcal{N}_g(k)$ denotes k th element of \mathcal{N}_g . In the sequel, we will omit the slice index g unless necessary otherwise.

Now, per-player cost function Z_n in slice g is determined as

$$Z_n(a_n, a_{-n}) = \begin{cases} Z_n^l, & \text{if } a_n = 0 \\ Z_n^s, & \text{if } a_n = 1 \\ Z_n^m(\mathbf{a}^g), & \text{if } a_n = 2 \end{cases} \quad (15)$$

where $a_{-n} = (a_1, a_2, \dots, a_{(n-1)}, a_{(n+1)}, \dots, a_{|\mathcal{N}_g|})$. Note that the slice-wide cost is then obtained as $\sum_{n \in \mathcal{N}_g} Z_n(\mathbf{a}^g)$, while the system-wide cost is obtained as $\sum_{g=1}^{N_{GS}} \sum_{n \in \mathcal{N}_g} Z_n(\mathbf{a}^g)$.

Next, we can formulate our game as a strategic game such as $\mathcal{G}^g = [\mathcal{N}_g, \mathbb{S}^g, \{Z_n\}_{n \in \mathcal{N}_g}]$, where each vehicle (i.e., a player) wants to minimize its cost given others' decision profiles, i.e.,

$$\min_{a_n \in \mathcal{A}_n^g = \{0, 1, 2\}} Z_n(a_n, a_{-n}). \quad (16)$$

Then, the NE of the game is defined as a strategy profile from which no vehicle intends to deviate as long as others' strategies remain unchanged [47], and the strategy profile $\mathbf{a}^* = (a_1^*, a_2^*, \dots, a_n^*)$ is called a pure-strategy NE if and only if

$$Z_n(a_n^*, a_{-n}^*) \leq Z_n(a_n, a_{-n}^*), \quad \forall a_n \in \mathcal{A}_n^g, \forall n \in \mathcal{N}_g. \quad (17)$$

B. Verifying the Game to Be a Potential Game

A game is considered as a potential game if it has a potential function that maps each player's strategy update to the change of the real number the function produces. The potential game is very useful since it always ensures the existence of NE. Moreover, any asynchronous update (i.e., only one player updates at each given decision time) always leads to convergence to an NE in finite time.

Our game \mathcal{G}^g becomes an ordinal potential game if and only if a potential function $\phi(\mathbf{a}^g) : \mathbb{S}^g \mapsto \mathbb{R}$ exists, such that $\forall n \in \mathcal{N}_g$,

$$\begin{aligned} Z_n(a_n, a_{-n}) - Z_n(a'_n, a_{-n}) &> 0 \\ \Leftrightarrow \phi(a_n, a_{-n}) - \phi(a'_n, a_{-n}) &> 0, \quad \forall a_n, a'_n \in \mathcal{A}_n^g. \end{aligned} \quad (18)$$

To derive our game's potential function, we first state the dynamics of the game by considering three cases: local computing vs. sBS offloading, local computing vs. mBS offloading, and sBS offloading vs. mBS offloading.

local computing ($a_n = 0$) vs. **mBS offloading** ($a_n = 2$): Using (8) and (14), $Z_n^l < Z_n^m$ leads to

$$T_n^{(l,m)} < W_n + \sum_{i \neq n, i \in \mathcal{N}_g} W_i \cdot I_{\{a_i=2\}}, \quad (19)$$

where

$$T_n^{(l,m)} = \frac{B_m^g \cdot W_n \cdot c_n \cdot (\epsilon_n + \frac{\lambda}{f_n^l} - \frac{\lambda}{f_m})}{b_n \cdot (P_n^m + \lambda)} \cdot \log_2(1 + SNR_n^m). \quad (20)$$

Equation (19) states that if the threshold $T_n^{(l,m)}$ is smaller than the sum of required bandwidth by the vehicles choosing mBS offloading, vehicle n should prefer local computing against mBS offloading.

sBS offloading ($a_n = 1$) vs. **mBS offloading** ($a_n = 2$): Using (12) and (14), $Z_n^s < Z_n^m$ leads to

$$T_n^{(s,m)} < W_n + \sum_{i \neq n, i \in \mathcal{N}_g} W_i \cdot I_{\{a_i=2\}}, \quad (21)$$

where

$$\begin{aligned} T_n^{(s,m)} &= \frac{B_m^g \cdot W_n}{b_n \cdot (P_n^m + \lambda)} \cdot \log_2(1 + SNR_n^m) \\ &\times \left(\frac{b_n \cdot (P_n^s + \lambda)}{B_s \cdot \log_2(1 + SNR_n^s)} + \lambda \cdot c_n \cdot \left(\frac{1}{f_s} - \frac{1}{f_m} \right) \right). \end{aligned} \quad (22)$$

local computing ($a_n = 0$) vs. **sBS offloading** ($a_n = 1$): Using (8) and (12), $Z_n^l < Z_n^s$ leads to

$$c_n \cdot \left(\epsilon_n + \frac{\lambda}{f_n^l} \right) < \frac{b_n(P_n^s + \lambda)}{B_s \cdot \log_2(1 + SNR_n^s)} + \lambda \cdot \frac{c_n}{f_s}, \quad (23)$$

which then becomes

$$T_k^{(s,m)} > T_k^{(l,m)}, \quad (24)$$

by applying (20) and (22).

Now, for vehicle k in slice g , there exist six possible decision changes, from a_k to a'_k . Since a vehicle updates its decision only when it will reduce its cost, (19), (21), and (24) instantly give us the following necessary conditions:

$$a_k = 0 \rightarrow a'_k = 1 : T_k^{(l,m)} - T_k^{(s,m)} > 0, \quad (25)$$

$$a_k = 1 \rightarrow a'_k = 0 : T_k^{(s,m)} - T_k^{(l,m)} > 0, \quad (26)$$

$$a_k = 0 \rightarrow a'_k = 2 : T_k^{(l,m)} > W_k + \sum_{i \neq n, i \in \mathcal{N}_g} W_i \cdot I_{\{a_i=2\}}, \quad (27)$$

$$a_k = 2 \rightarrow a'_k = 0 : T_k^{(l,m)} < W_k + \sum_{i \neq n, i \in \mathcal{N}_g} W_i \cdot I_{\{a_i=2\}}, \quad (28)$$

$$a_k = 1 \rightarrow a'_k = 2 : T_k^{(s,m)} > W_k + \sum_{i \neq n, i \in \mathcal{N}_g} W_i \cdot I_{\{a_i=2\}}, \quad (29)$$

$$a_k = 2 \rightarrow a'_k = 1 : T_k^{(s,m)} < W_k + \sum_{i \neq n, i \in \mathcal{N}_g} W_i \cdot I_{\{a_i=2\}}. \quad (30)$$

Using these relationships, we can show that the multi-interface computation offloading decision game is a potential game by constructing its potential function ϕ with given decision profile \mathbf{a} , as stated in the following theorem.

Theorem 1: The proposed multi-interface computation offloading decision game is an ordinal potential game with a potential function $\phi(\mathbf{a}^g)$ given as

$$\begin{aligned} \phi(\mathbf{a}^g) &= \sum_{i \in \mathcal{N}_g} W_i \cdot T_i^{(s,m)} \cdot I_{\{a_i=1\}} + \sum_{i \in \mathcal{N}_g} W_i \cdot T_i^{(l,m)} \cdot I_{\{a_i=0\}} \\ &+ \frac{1}{2} \sum_{i \in \mathcal{N}_g} \left(2W_i + \sum_{j \neq i, j \in \mathcal{N}_g} W_j \cdot I_{\{a_j=2\}} \right) \cdot W_i \cdot I_{\{a_i=2\}}. \end{aligned} \quad (31)$$

Proof: We need to show that (18) holds for all possible six combinations of (a_n, a'_k) . To do so, we rewrite $\phi(\mathbf{a}^g)$ as $\phi(a_k, a_{-k})$, and for $i, j \in \mathcal{N}_g$ we apply three possible values

of a_k to (31) such as

$$\begin{aligned} \phi(a_k = 0, a_{-k}) &= \sum_{i \neq k} W_i \cdot T_i^{(s,m)} \cdot I_{\{a_i=1\}} \\ &+ W_k \cdot T_k^{(l,m)} + \sum_{i \neq k} W_i \cdot T_i^{(l,m)} \cdot I_{\{a_i=0\}} \\ &+ \sum_{i \neq k} W_i^2 \cdot I_{\{a_i=2\}} \\ &+ \frac{1}{2} \sum_{i \neq k} \left(\sum_{\substack{j \neq i \\ j \neq k}} W_j \cdot I_{\{a_j=2\}} \right) W_i \cdot I_{\{a_i=2\}}, \end{aligned} \quad (32)$$

$$\begin{aligned} \phi(a_k = 1, a_{-k}) &= W_k \cdot T_k^{(s,m)} + \sum_{i \neq k} W_i \cdot T_i^{(s,m)} \cdot I_{\{a_i=1\}} \\ &+ \sum_{i \neq k} W_i \cdot T_i^{(l,m)} \cdot I_{\{a_i=0\}} \\ &+ \sum_{i \neq k} W_i^2 \cdot I_{\{a_i=2\}} \\ &+ \frac{1}{2} \sum_{i \neq k} \left(\sum_{\substack{j \neq i \\ j \neq k}} W_j \cdot I_{\{a_j=2\}} \right) W_i \cdot I_{\{a_i=2\}}, \end{aligned} \quad (33)$$

$$\begin{aligned} \phi(a_k = 2, a_{-k}) &= \sum_{i \neq k} W_i \cdot T_i^{(s,m)} \cdot I_{\{a_i=1\}} \\ &+ \sum_{i \neq k} W_i \cdot T_i^{(l,m)} \cdot I_{\{a_i=0\}} \\ &+ W_k^2 + \sum_{i \neq k} (W_i^2 + W_k W_i) \cdot I_{\{a_i=2\}} \\ &+ \frac{1}{2} \sum_{i \neq k} \left(\sum_{\substack{j \neq i \\ j \neq k}} W_j \cdot I_{\{a_j=2\}} \right) W_i \cdot I_{\{a_i=2\}}. \end{aligned} \quad (34)$$

Then, when $a_k = 0 \rightarrow a'_k = 1$ happens, it means vehicle k 's sBS offloading cost is smaller than its local computing cost, i.e., $Z_k(a_k, a_{-k}) - Z_k(a'_k, a_{-k}) > 0$. In addition, by (32) and (33), we have

$$\phi(a_k, a_{-k}) - \phi(a'_k, a_{-k}) = W_k \cdot (T_k^{(l,m)} - T_k^{(s,m)}) > 0, \quad (35)$$

where the last inequality comes from (25). As a result, it satisfies the definition of ordinal potential games in (18). Similarly, the decision update $a_k = 1 \rightarrow a'_k = 0$ means $Z_k(a_k, a_{-k}) - Z_k(a'_k, a_{-k}) > 0$, which also satisfies (18) such that

$$\phi(a_k, a_{-k}) - \phi(a'_k, a_{-k}) = W_k \cdot (T_k^{(s,m)} - T_k^{(l,m)}) > 0, \quad (36)$$

where the last inequality comes from (26).

Next, when $a_k = 0 \rightarrow a'_k = 2$ happens, vehicle k 's mBS offloading cost is smaller than its local computing cost, i.e., $Z_k(a_k, a_{-k}) - Z_k(a'_k, a_{-k}) > 0$. By (32) and (34), we have

$$\begin{aligned} & \phi(a_k, a_{-k}) - \phi(a'_k, a_{-k}) \\ &= W_k \cdot \left(T_k^{(l,m)} - W_k - \sum_{i \neq k, i \in \mathcal{N}_g} W_i \cdot I_{(a_i=2)} \right) > 0, \end{aligned} \quad (37)$$

where the last inequality comes from (27), thus satisfying (18). On the other hand, when $a_k = 2 \rightarrow a'_k = 0$, (18) is also satisfied since

$$\begin{aligned} & \phi(a_k, a_{-k}) - \phi(a'_k, a_{-k}) \\ &= W_k \cdot \left(W_k + \sum_{i \neq k, i \in \mathcal{N}_g} W_i \cdot I_{(a_i=2)} - T_k^{(l,m)} \right) > 0, \end{aligned} \quad (38)$$

where the last inequality comes from (28).

Finally, when $a_k = 1 \rightarrow a'_k = 2$, (33) and (34) provide

$$\begin{aligned} & \phi(a_k, a_{-k}) - \phi(a'_k, a_{-k}) \\ &= W_k \cdot \left(T_k^{(s,m)} - W_k - \sum_{i \neq k, i \in \mathcal{N}_g} W_i \cdot I_{(a_i=2)} \right) > 0, \end{aligned} \quad (39)$$

where the last inequality comes from (29), leading to (18). Similarly, according to (30), $a_k = 2 \rightarrow a'_k = 1$ leads to

$$\begin{aligned} & \phi(a_k, a_{-k}) - \phi(a'_k, a_{-k}) \\ &= W_k \cdot \left(W_k + \sum_{i \neq k, i \in \mathcal{N}_g} W_i \cdot I_{(a_i=2)} - T_k^{(s,m)} \right) > 0, \end{aligned} \quad (40)$$

thus satisfying (18).

As a result, all six cases satisfy (18), which completes the proof. \blacksquare

By the property of ordinal potential games, Theorem 1 also confirms that our game should have a Nash equilibrium and it can be reached in finite time by developing a proper iterative decision update algorithm, where the latter will be discussed in the next section.

V. GAME-SLICING BASED MULTI-INTERFACE COMPUTATION OFFLOADING ALGORITHM

One key feature of the ordinal potential game is that it should be an *asynchronous improvement* process to ensure finite time convergence to a NE [47], [48]. It means that only one vehicle should be allowed to update its decision at a time. To fulfill the requirement, this section proposes the iterative offloading decision update algorithm called *game-slicing based multi-interface computation offloading (GaMiCO)* that satisfies the asynchronous improvement property. The proposed algorithm also addresses another important issue: how each vehicle can obtain the information necessary for its distributed decision update. This arises since the mBS communication model in (4) states that per-vehicle uplink channel resources depend on the aggregate demand of the vehicles offloading to the mBS. Therefore, each vehicle needs to know the sum demand while not knowing which vehicles are choosing mBS offloading.

Algorithm 1 provides the pseudo-code of the proposed algorithm. At first, every vehicle assumes that all the vehicles in the same slice initially choose local computing,

i.e., $a_n = 0, \forall n$ (line 4), and then determines its initial decision (line 5).⁷ When the derived decision is either $a_n = 0$ or $a_n = 1$, the vehicle exits the algorithm while performing corresponding operation, i.e., local computing or offloading to sBS (lines 7 and 8). Otherwise, the vehicle will participate in the mBS offloading game by sending the 'Offloading Request' message to the mBS, with W_n enclosed (line 11). We assume this uplink transmission occurs simultaneously among the vehicles with $a_n = 2$ in the same uplink subframe of 5G, where each vehicle is using its own assigned physical resource block (PRB). After collecting all the requests, the mBS broadcasts the calculated sum demand to the vehicles via the 'Offloading Response' message (lines 12–13). The vehicles involved in the game will go through a repeated process with multiple iterations until convergence to a NE is reached (lines 10–26). Specifically, each vehicle re-calculates its best decision among three options (line 14), and if the result is to deviate from $a_n = 2$ then the vehicle will send the 'Change Decision' message to the mBS (line 16). Again, such transmissions from multiple vehicles occur in the same uplink subframe. Then, the mBS selects one vehicle among them randomly, and broadcasts the 'Change Approval' message with the chosen vehicle's id and the updated sum demand to exclude W_n of the chosen one (line 17). The selected vehicle changes its decision (line 18) while other vehicles keep their decisions as before (line 21). When the mBS no longer receives any Change Decision message, it broadcasts the 'Convergence' message to announce that a NE is reached and then the game is finished (line 26).

A. Convergence Analysis of the Algorithm

Despite the fact that a potential game ensures finite-time convergence, it is practically important to understand how fast it would converge to a NE. To address this, we first investigate the time complexity of the algorithm using the big \mathcal{O} notation and then analyze the upperbound of the time required for the algorithm to converge. Because the time complexity is dominated by the algorithm's most time-consuming parts, we focus on the offloading decision process corresponding to lines 14–24. Denoting by M_g the upperbound of the number of iterations until convergence, \mathcal{N}_g vehicles perform argmin until M_g iterations in the worst case. Since the complexity of each iteration is $\mathcal{O}(\mathcal{N}_g)$, the time complexity of the algorithm becomes $\mathcal{O}(M_g \times \mathcal{N}_g)$.

Next, we analyze the upperbound of the convergence time of the algorithm. Algorithm 1 would require at most $(3+2M_g)$ TTIs (transmission time intervals), considering that each uplink/downlink transmission takes one subframe and a subframe corresponds to one TTI. More specifically, the initial exchange of *Offloading Request* and *Offloading Response* messages costs two TTIs, each iteration also needs two TTIs for *Change Decision* and *Change Approval* messages, and the final *Con-*

⁷Assuming $a_n = 0, \forall n$ at initialization is reasonable since it will give the least mBS offloading cost (the cost grows as more vehicles join mBS offloading). That is, if vehicle n wouldn't choose mBS offloading even when $a_n = 0, \forall n$, it will never choose it in any circumstances (thus not necessitating joining the mBS offloading game).

Algorithm 1 Game-slicing based multi-interface computation offloading (GaMiCO)

```

1: define  $a_n(t)$  as  $a_n$  at decision iteration  $t$ 
2: set  $t = 0$ 
3: initialization (for each vehicle  $n \in \mathcal{N}_g$ ):
4:   assume  $a_{n'}(0) = 0, \forall n' \neq n, n' \in \mathcal{N}_g$ 
5:   compute  $a_n(0) = \operatorname{argmin}_{a_n \in \{0,1,2\}} Z_n(a_n, a_{-n})$ 
6: end initialization
7: if  $a_n(0) = 0$  then run the task locally and exit
8: else if  $a_n(0) = 1$  then offload the task to sBS and exit
9: end if
10: repeat for each vehicle  $n$  and each iteration  $t$ :
11:   send Offloading Request to mBS with  $W_n$  enclosed
12:   receive Offloading Response from mBS
13:     with  $\sum_{i \in \mathcal{N}_g} W_i \cdot I_{\{a_i=2\}}$  enclosed
14:   compute  $a'_n(t) = \operatorname{argmin}_{a_n \in \{0,1,2\}} Z_n(a_n, a_{-n})$ 
15:   if  $a_n(t) \neq a'_n(t)$  then
16:     send Change Decision to mBS
17:     if receive Change Approval from mBS then
18:       update  $a_n(t+1) = a'_n(t)$ 
19:       if  $a_n(t+1) \neq 2$  then go to line 7
20:     end if
21:   else keep the original decision:  $a_n(t+1) = a_n(t)$ 
22:   end if
23:   else keep the original decision:  $a_n(t+1) = a_n(t)$ 
24:   end if
25:   update the iteration:  $t \leftarrow t + 1$ 
26: until Convergence is received from mBS

```

vergence announcement takes another TTI. Note that in 5G NR, one TTI equals to 0.125 ms [49].

To derive M_g , we consider the upperbound of the potential function $\phi(\mathbf{a}^g)$ (denoted by P_{\max}) and the minimum amount of decrease in $\phi(\mathbf{a}^g)$ when a vehicle changes its decision (denoted by P_{\min}). Then, M_g can be determined as

$$M_g = \lceil P_{\max}/P_{\min} \rceil, \quad (41)$$

following the approach introduced in [21].

Let us define D_n^{\max} for vehicle $n \in \mathcal{N}_g$ as

$$D_n^{\max} = W_n \cdot \max \left\{ T_n^{(s,m)}, T_n^{(l,m)}, W_n + \frac{1}{2} \sum_{j \neq n, j \in \mathcal{N}_g} W_j \right\}. \quad (42)$$

Then, by the form of $\phi(\mathbf{a})$ in (31), P_{\max} is obtained as

$$P_{\max} = \sum_{i \in \mathcal{N}_g} D_i^{\max}. \quad (43)$$

Next, the definition of P_{\min} implies that $\phi(a_k, a_{-k}) - \phi(a'_k, a_{-k}) \geq P_{\min}$. To obtain P_{\min} , we define D_n^{\min} as

$$D_n^{\min} = \min \left\{ W_n \cdot \left| T_n^{(s,m)} - T_n^{(l,m)} \right|, \right. \\ \left. W_n \cdot \left| T_n^{(l,m)} - \sum_{i \in \mathcal{N}_g} W_i \right|, W_n \cdot \left| T_n^{(s,m)} - \sum_{i \in \mathcal{N}_g} W_i \right| \right\}. \quad (44)$$

Note that each term inside the min operator states the difference in the potential function due to the three types of decision changes, i.e., between local computing and sBS offloading, local computing and mBS offloading, and sBS offloading and mBS offloading, each corresponding to (35) and (36), (37) and (38), and (39) and (40), respectively. Then, P_{\min} is obtained as

$$P_{\min} = \min_n D_n^{\min}. \quad (45)$$

In Section VII, we will numerically evaluate the number of iterations required for convergence in practical scenarios, to show that the algorithm actually converges much earlier than the derived M_g in most cases.

VI. PRICE OF ANARCHY ANALYSIS

The price of anarchy (PoA) measures how much ‘game theoretic competition’ can approximate the performance of ‘cooperation between game players’ [50]. Following the definition, this section analyzes the efficiency of our multi-interface computation offloading game slicing by quantifying the ratio of the worst-case NE performance to the optimal performance (achieved by the centralized solution), in a term of the system-wide cost $\sum_{g=1}^{N_{GS}} \sum_{n \in \mathcal{N}_g} Z_n(\mathbf{a}^g)$.

Let $\bar{\mathbf{a}}^g = (\bar{a}_{\mathcal{N}_g(1)}, \bar{a}_{\mathcal{N}_g(2)}, \dots, \bar{a}_{\mathcal{N}_g(|\mathcal{N}_g|)})$ be the optimal decision profile minimizing the system-wide cost in slice g , and $\hat{\mathbf{a}}^g = (\hat{a}_{\mathcal{N}_g(1)}, \hat{a}_{\mathcal{N}_g(2)}, \dots, \hat{a}_{\mathcal{N}_g(|\mathcal{N}_g|)})$ be the worst-case NE such that $\hat{\mathbf{a}}^g = \operatorname{argmax}_{\mathbf{a}^g \in \mathbb{S}^g} \sum_{g=1}^{N_{GS}} \sum_{n \in \mathcal{N}_g} Z_n(\mathbf{a}^g)$. Then, the PoA is defined as

$$PoA = \frac{\sum_{g=1}^{N_{GS}} \sum_{n \in \mathcal{N}_g} Z_n(\hat{\mathbf{a}}^g)}{\sum_{g=1}^{N_{GS}} \sum_{n \in \mathcal{N}_g} Z_n(\bar{\mathbf{a}}^g)}, \quad (46)$$

where a smaller PoA implies that the game shows better performance.

Unfortunately, it is mathematically intractable to derive an exact decision profile that can achieve the worst-case NE. Moreover, it is infeasible to derive the optimal centralized solution due to the exponentially-growing search space with the number of vehicles. Therefore, following the approach presented in [20], [21], we investigate the upper and lower bounds of the PoA instead, to analyze how far the performance of the proposed game can deviate from the optimal performance. To do so, we define two terms, $Z_{n,\min}^m$ and $Z_{n,\max}^m$, which are the minimum and the maximum mBS offloading cost of vehicle n , respectively, provided that n chooses mBS offloading. $Z_{n,\min}^m$ is achieved when only vehicle n selects mBS offloading, so that it can fully utilize the mBS’s channel bandwidth. In such a case, by (4) and (5), the uplink data rate of vehicle n in slice g is maximized as

$$r_{n,\max}^m = B_m^g \cdot \log_2(1 + SNR_n^m), \quad (47)$$

and thus $Z_{n,\min}^m$ is derived as

$$Z_{n,\min}^m = \frac{b_n \cdot (P_n^m + \lambda)}{B_m^g \cdot \log_2(1 + SNR_n^m)} + \frac{c_n \lambda}{f_m}, \quad (48)$$

by applying $r_{n,\max}^m$ to $r_n^m(\mathbf{a}^g)$ in (14).

$Z_{n,\max}^m$ is obtained when all the users in the same slice select mBS offloading, where vehicle n 's uplink data rate is minimized as

$$r_{n,\min}^m = \frac{B_m^g \cdot W_n}{\sum_{i \in \mathcal{N}_g} W_i} \cdot \log_2(1 + SNR_n^m), \quad (49)$$

and hence $Z_{n,\max}^m$ becomes

$$Z_{n,\max}^m = \frac{b_n \cdot (P_n^m + \lambda)}{B_m^g \cdot \log_2(1 + SNR_n^m)} \cdot \frac{\sum_{i \in \mathcal{N}_g} W_i}{W_n} + \frac{c_n \lambda}{f_m}. \quad (50)$$

Now, we can show the two bounds of the PoA in the following theorem.

Theorem 2: The PoA of the proposed multi-interface computation offloading game is bounded as

$$1 \leq PoA \leq \frac{\sum_{g=1}^{N_{GS}} \sum_{n \in \mathcal{N}_g} \max\{Z_n^l, Z_n^s, Z_{n,\max}^m\}}{\sum_{g=1}^{N_{GS}} \sum_{n \in \mathcal{N}_g} \min\{Z_n^l, Z_n^s, Z_{n,\min}^m\}}. \quad (51)$$

Proof: Since $\bar{\mathbf{a}}^g$ minimizes the system-wide cost, the PoA should be lower-bounded by 1. For the upper bound of PoA, we need to find the lower bound of $\sum_{n \in \mathcal{N}_g} Z_n(\bar{\mathbf{a}}^g)$ and the upper bound of $\sum_{n \in \mathcal{N}_g} Z_n(\hat{\mathbf{a}}^g)$.

If vehicle n performs mBS offloading in the optimal solution $\bar{\mathbf{a}}^g$, i.e., $\bar{a}_n = 2$, we have

$$Z_n(\bar{\mathbf{a}}^g) = Z_n^m(\bar{\mathbf{a}}^g) \geq Z_{n,\min}^m \geq \min\{Z_n^l, Z_n^s, Z_{n,\min}^m\}.$$

Otherwise (i.e., $\bar{a}_n = 0$ or 1), we have

$$Z_n(\bar{\mathbf{a}}^g) = \min\{Z_n^l, Z_n^s\} \geq \min\{Z_n^l, Z_n^s, Z_{n,\min}^m\}.$$

So, we can conclude that

$$Z_n(\bar{\mathbf{a}}^g) \geq \min\{Z_n^l, Z_n^s, Z_{n,\min}^m\}. \quad (52)$$

If vehicle n performs mBS offloading in the NE solution $\hat{\mathbf{a}}^g$, i.e., $\hat{a}_n = 2$, we have

$$Z_n(\hat{\mathbf{a}}^g) = Z_n^m(\hat{\mathbf{a}}^g) \leq Z_{n,\max}^m \leq \max\{\min\{Z_n^l, Z_n^s\}, Z_{n,\max}^m\}.$$

Otherwise (i.e., $\hat{a}_n = 0$ or 1), we have

$$Z_n(\hat{\mathbf{a}}^g) = \min\{Z_n^l, Z_n^s\} \leq \max\{\min\{Z_n^l, Z_n^s\}, Z_{n,\max}^m\}.$$

So, we can conclude that

$$Z_n(\hat{\mathbf{a}}^g) \leq \max\{\min\{Z_n^l, Z_n^s\}, Z_{n,\max}^m\}. \quad (53)$$

Finally, we can obtain the upper bound of the PoA by applying (52) and (53) to (46). ■

In Section VII, we will numerically evaluate the system-wide cost achieved by the optimal decision set, to show how the proposed game theoretic approach performs well compared to the centralized optimal method.

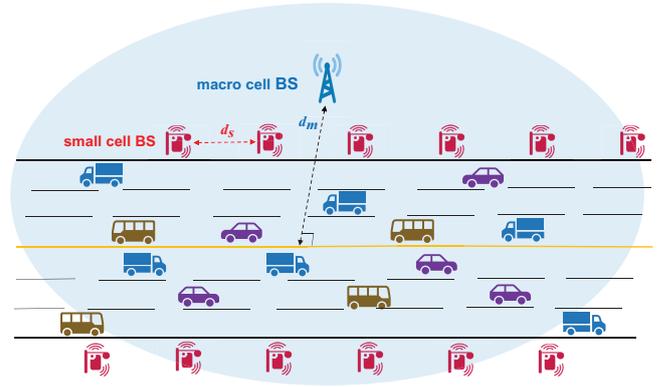


Fig. 3. The simulation scenario and topology.

VII. PERFORMANCE EVALUATION

This section evaluates the performance of the proposed game theoretic mechanism via numerical simulations. Fig. 3 illustrates our simulation scenario, where a mBS with 1 km coverage radius [51] coexists with multiple sBSs deployed on both sides of a six-lane highway.⁸ The mBS is d_m meters away from the center of the road and the inter-sBS distance is d_s meters. We set $d_m = 500$ m and $d_s = 100$ m [52] making 32 sBSs deployed in the whole network (16 sBSs on each side of the road).

For each sBS, we set the channel bandwidth as $B_s = 100$ MHz and the transmit power as $P_n^s = 23$ dBm [53].⁹ We also set $G_{TX} = G_{RX} = 24.5$ dBi for the mmWave path loss model, utilizing a configuration of 10° beamwidth given by 12×12 antenna elements on a 66 mm \times 66 mm plate [38], [54].¹⁰ For the mBS, we set the transmit power as $P_n^m = 23$ dBm.¹¹ Then, we set the mBS channel bandwidth to $B_m = 100$ MHz in VII-A through VII-G (to observe the impact of the arrival rate on game slicing),¹² while in VII-H we vary the mBS bandwidth in finding the optimal number of game slices that minimizes the system-wide cost at various vehicle arrival rates, to consider practical scenarios.

For the tasks to offload, the input data size b_n is uniformly distributed in $[10, 30]$ KB and the required number of CPU cycles is $c_n = 2500$ (cycles/bit). For local computing, we set $\epsilon_n = 1.2$ (nJ/cycle) according to [55], [56], and set the computing power of a vehicle as $f_n^l = 2.5$ GHz. For computing at MEC servers, we set the computational capability of the mMEC and the sMEC as $f_m = 10$ GHz [1] and $f_s = 5$ GHz, respectively. In addition, considering the system-level requirements of various vehicular network applications [43]–[45], we set the elastic deadline for uplink transmission to the mBS as $t_d = 100$ ms.

⁸We have chosen a highway scenario for the smart vehicles, since it is considered as an early-stage deployment site of autonomous driving due to less unpredictability (e.g., no pedestrians, no traffic lights).

⁹23 dBm is the maximum total radiated power (TRP) for power class 3.

¹⁰Considering the form factor of a vehicle, the assumed antenna size seems practical.

¹¹23 dBm is the maximum uplink transmit power of power class 3 for non high power UE (HPUE).

¹²We assume the mBS bandwidth is fully utilized, which is possible in suburban/rural highway scenarios considered in this paper.

The arrival and mobility patterns of vehicles are simulated by the widely-used SUMO (Simulation of Urban MObility) simulator [57]. Specifically, we assume the vehicles have an average speed of 80 km/h while arriving at the road section in Fig. 3 with varying arrival rates as $\{0.2, 0.6, 1.0, 1.4, 1.8\}$ (vehicles/sec), corresponding to an average of $\{21, 54, 84, 110, 137\}$ (vehicles) on the road. For a given arrival rate, we run one hundred simulations each with different data size realizations, and in each simulation run we wait until the road is fully occupied by vehicles and then take the snapshot of vehicle locations at ten independent time moments. In a snapshot, each vehicle's mBS and sBS uplink channels are generated by the 3GPP channel model, where a vehicle chooses the closest sBS as its sBS. Then, we run Algorithm 1 in each snapshot, and observe the average performance over one thousand snapshots (since there are ten runs, each with one hundred snapshots).

We set λ , the weighting factor between time and energy, to 0.5, while it varies in Section VII-E to investigate its impact on the system dynamics.

A. Impact of Game Slicing on the System-wide Cost

Fig. 4 illustrates the evolution of the system-wide cost with Algorithm 1's iteration, for various arrival rates. According to the arrival rate, Fig. 5 shows that some values of N_{GS} could make the game convergence plus uplink transmission time grow beyond the channel coherence time,¹³ thus violating the quasi-static assumption in Section III (i.e., game convergence and uplink transmission must be completed within the channel coherence time). Therefore, we have excluded such cases in each of Fig. 4 (like in (d) and (e)). Then, we analyze the effect of game slicing on the system-wide cost. When the arrival rate is 0.2, no slicing (i.e., $N_{GS} = 1$) minimizes the system-wide cost. As the arrival rate gradually increases, however, the cost-minimizing number of game slices (in terms of N_{GS}) varies. For the arrival rates $\{0.6, 1.0, 1.4, 1.8\}$ (corresponding to a total of $\{54, 84, 110, 137\}$ vehicles respectively), the optimal N_{GS} is found as $\{1, 8, 16, 32\}$. In conclusion, unlike the existing game theoretic work without game slicing, our approach of slicing the game according to the traffic volume is more beneficial in minimizing the system-wide cost. In Section VII-D, we will compare the performance of our algorithm with other state-of-the-art game-theoretic offloading schemes. Then in Section VII-G, we will further investigate the impacts of game slicing on the cost and PoA.

B. Convergence Time to a NE with Varying Arrival Rate

In every case of Fig. 4, the system-wide cost tends to decrease with iterations, eventually converging to a minimal value once an NE is reached. For each arrival rate in Fig. 4, we consider the optimal number of game slices minimizing the system-wide cost. Then, thus-obtained case's NE is reached after $\{2, 13, 14, 16, 12\}$ iterations, corresponding to the five arrival rate values. Since the algorithm consumes

¹³Since mBS's channel coherence time (34.38 ms) is tighter than sBS's (40.95 ms), we have considered mBS's channel coherence time.

$(3 + 2 \cdot (\text{the number of iterations until convergence}))$ TTIs as derived in Section V-A, the observed iterations correspond to $\{0.88, 3.63, 3.88, 4.38, 3.38\}$ ms respectively. On the contrary, in the legacy game theoretic approaches without game slicing (i.e., $N_{GS} = 1$), the convergence time becomes worse as $\{0.88, 3.63, 12.38, 23.63, 32.63\}$ ms for those arrival rates.

We then compare the observed number of iterations until convergence with M , the upperbound of the number of decision iterations required for convergence in (41). For the five arrival rates considered, M is calculated as $M = \{17, 50, 74, 101, 246\}$ iterations respectively, and hence the proposed algorithm converges to an NE much faster than the maximal possible iterations predicted by M .

Moreover, Fig. 6 presents the total number of vehicles on the road, the number of vehicles offloading to mBS at NE, and the number of vehicles offloading to sBS at NE, each obtained by averaging over ten snapshots. Naturally, all three numbers tend to increase with arrival rate, while the number of 'mBS offloading' vehicles grows more slowly than that of 'sBS offloading' vehicles, gradually saturating due to the bandwidth competition in mBS's uplink channel.

C. Impact of Game Slicing on the Quasi-static Assumption

In this simulation, we show the impact of game slicing on ensuring the quasi-static assumption. Figs. 7 and 8 show the combined time taken for the game to converge and the tasks to be uplink-transmitted after that, with and without game slicing, respectively. Fig. 7 shows that with game slicing, it takes at most 32.89 ms even in the worst case, which is less than the channel coherence time. On the contrary, Fig. 8 shows that without game slicing, it may exceed the channel coherence time as the arrival rate increases. In addition, the combined time shows a large variance among the vehicles in Fig. 7 whereas the variance becomes negligible in Fig. 8. This is because the vehicles belonging to the same game slice have the same game convergence time while having a similar uplink transmission time given by (4),¹⁴ while the vehicles belonging to different slices would have different game convergence and uplink transmission times. In conclusion, with varying arrival rate, the quasi-static assumption can be guaranteed only by applying our game slicing method.

D. Comparison with Other Baseline Offloading Schemes

We evaluate our GaMiCO in comparison with the following game-theoretic offloading schemes, in terms of the system-wide cost and the game convergence time.

- **Local computing only (LCO):** An algorithm where vehicles always compute their tasks locally.
- **Distributed computation offloading algorithm (DCO):** The algorithm in [21] that addresses the problem of making decisions between local computing and mBS offloading, considering how multiple mobile users can efficiently achieve wireless access coordination.
- **Distributed multi-channel computation offloading algorithm (DMCO):** The algorithm in [22] that proposes offloading decision-making between local and mBS

¹⁴Note that in Fig. 8, there is only one slice – the single global game.

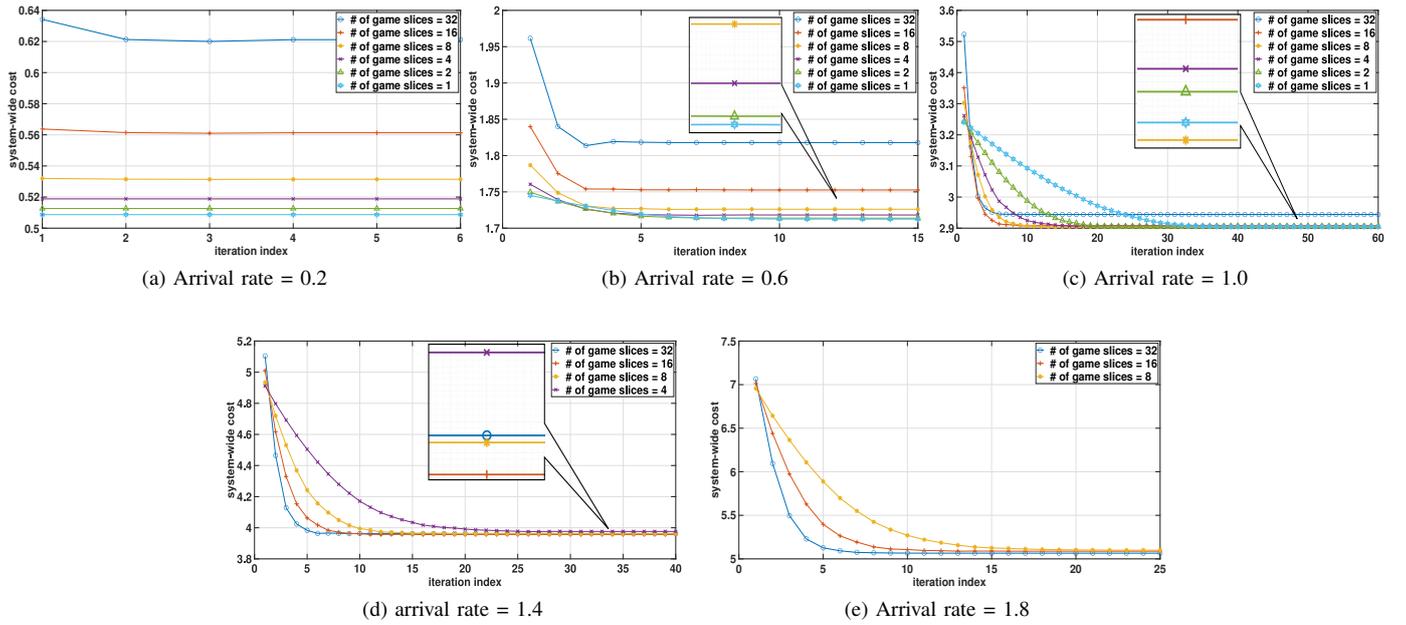


Fig. 4. The evolution of system-wide cost with varying arrival rate.

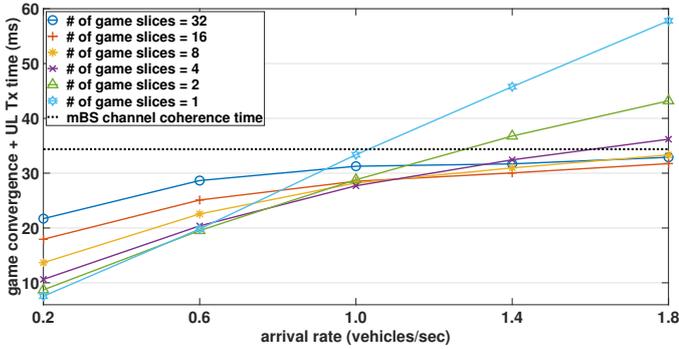


Fig. 5. The time for tasks to be uplink transmitted after the game converged for different N_{GS} with varying arrival rate.

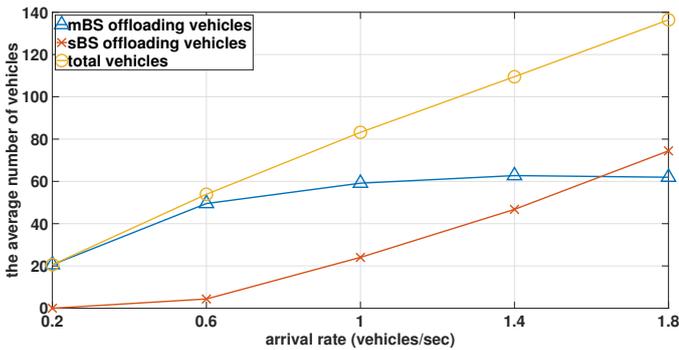


Fig. 6. The average number of vehicles with varying arrival rate.

while releasing the channel resources of transmission-completed tasks for transmission-incomplete tasks.

Regarding the system-wide cost, both GaMiCO and DMCO achieve the best performance where GaMiCO performs better than DMCO at high arrival rates while DMCO is better at low rates, as shown in Fig. 9. In terms of the game convergence time, however, GaMiCO performs the best among all the algorithms, as shown in Fig. 10. More importantly, the game

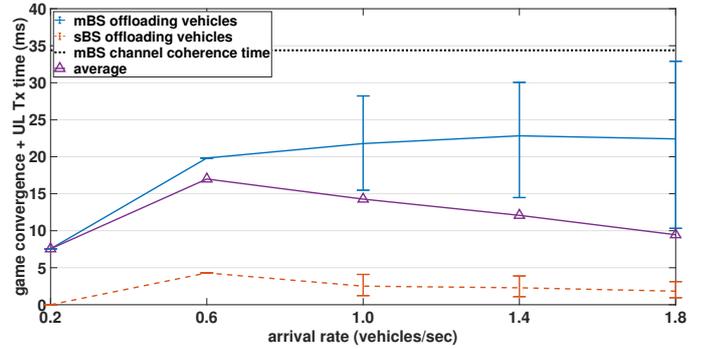


Fig. 7. The combined time taken for the game to converge and the tasks to be uplink-transmitted, with game slicing.

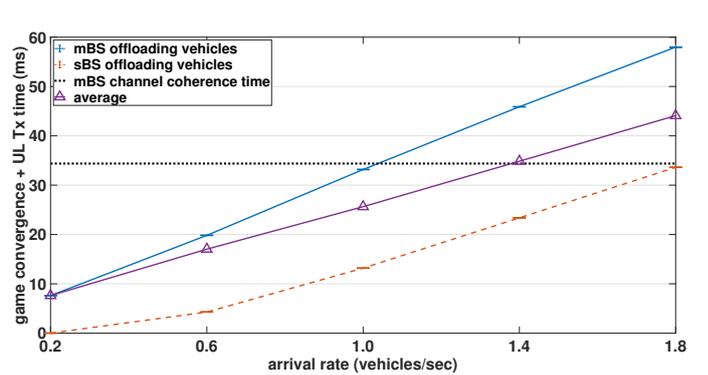


Fig. 8. The combined time taken for the game to converge and the tasks to be uplink-transmitted, without game slicing.

convergence time remains very small and steady in GaMiCO, whereas it increases fast with arrival rate in other algorithms. In conclusion, the proposed GaMiCO algorithm not only shows the best performance in minimizing the system-wide cost but also successfully suppresses the game convergence time thus ensuring the quasi-static assumption.

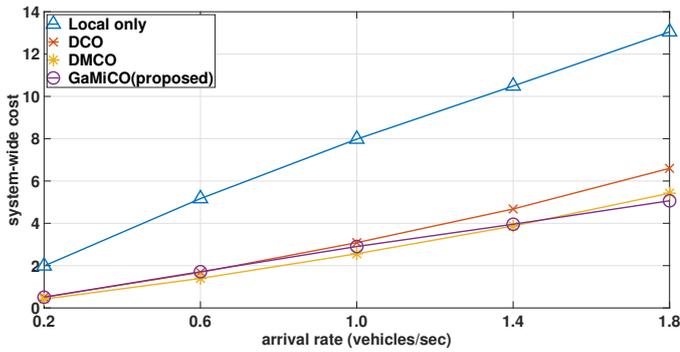


Fig. 9. The evolution of system-wide cost with varying arrival rate compared to other game-theoretic approaches.

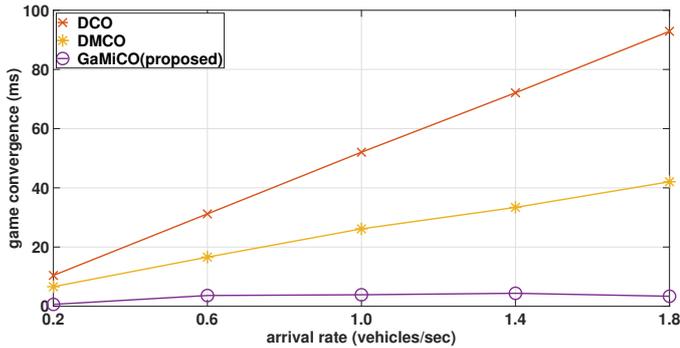


Fig. 10. The evolution of game convergence time with varying arrival rate compared to other game-theoretic approaches.

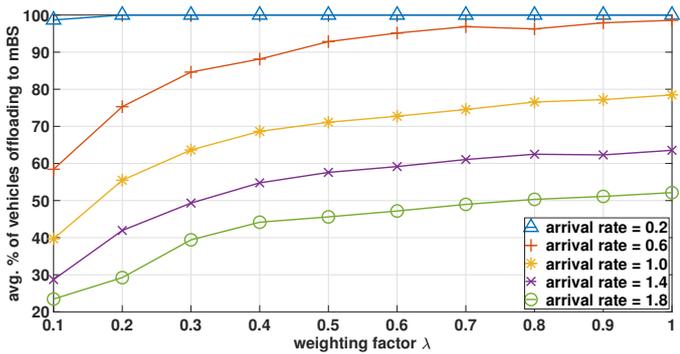


Fig. 11. The percentage of vehicles offloading mBS with varying λ .

E. Per-vehicle Decision with Varying λ

In this test, we present the impact of varying λ on the achieved NE. Fig. 11 shows that as λ grows, more vehicles try to offload to the mBS, regardless of the arrival rate. This is because the considered task requires heavy computation, and hence the computing latency at a MEC server affects vehicles' offloading decision significantly. That is, the larger λ gets, time cost reduction is more crucial, thus making the vehicles prefer mBS's superior MEC server against sBS's MEC server.

F. Impact of Mobility During a Game Period

In this simulation, we show the impact of vehicle mobility on the quasi-static assumption made in Section III. It is possible that a cell-edge located vehicle should leave its game due to the change of its associated sBS or the mBS. We measured the number of such vehicles, with the vehicle arrival rate varying as $\{0.2, 0.6, 1.0, 1.4, 1.8\}$. On average,

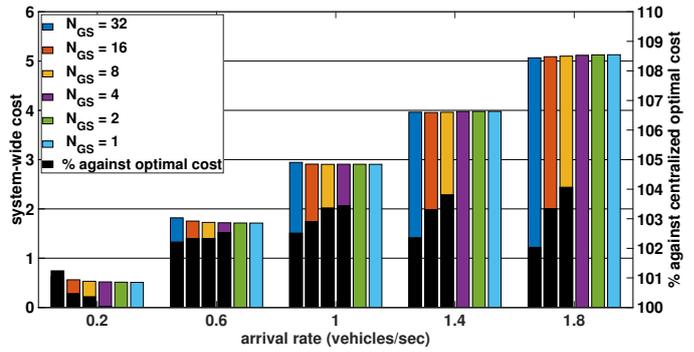


Fig. 12. Optimal vs. System-wide cost.

only $\{0, 1.82, 1.13, 1.63, 1.25\}$ % of the vehicles (respectively) are changing their cell association during a game period, thus having little impact on the proposed game framework.

G. Goodness of NE Compared to the Optimum

In this simulation, we evaluate the goodness of the achieved NE by comparing it to the optimal solution, in terms of the average system-wide cost (over one thousand snapshots), and also analyze the impact of game slicing on minimizing the system-wide cost. The optimal profile is obtained by a brute-force search method that enumerates all possible decision profiles and measures the cost of each profile.¹⁵ Then, we measure the ratio of the average system-wide cost to the centralized optimal cost (based on the obtained optimal profile), denoted by '% against centralized optimal cost', to show how much more cost our scheme incurs than the optimum.

Fig. 12 suggests that a proper number of game slices (i.e., N_{GS}) can effectively minimize the system-wide cost, where the best N_{GS} varies with the arrival rate. It also shows that optimizing the system-wide cost does not necessarily achieve the best performance compared to the centralized optimal cost, i.e., '% against centralized optimal cost'. Nevertheless, the performance loss (i.e., extra cost incurred) at the NE is measured as at most 3.59%, showing that our algorithm achieves a near-optimal performance. Considering that finding the optimal profile requires perfect information on all the vehicles and should deal with enormous search space, the observed performance loss (by applying game theory) seems quite reasonable.

H. Optimal Number of Game Slices for the Minimal Cost

In this simulation, we observe the tendency of the optimal number of game slices which minimizes the system-wide cost when various mBS bandwidths and arrival rates are considered (over one thousand snapshots). Fig. 13 shows that the optimal number of game slices varies according to a given pair of mBS bandwidth and arrival rate, presenting two different trends. First, when the mBS bandwidth is fixed, the optimal number of game slices tends to increase as the arrival rate grows. Similarly, when the arrival rate is fixed, the optimal number

¹⁵When the brute-search method is used, the time required to find the optimal overhead increases exponentially with the number of vehicles per game. Hence, Fig. 12 omits some of the optimal cost values which incurred prohibitively large computational time.

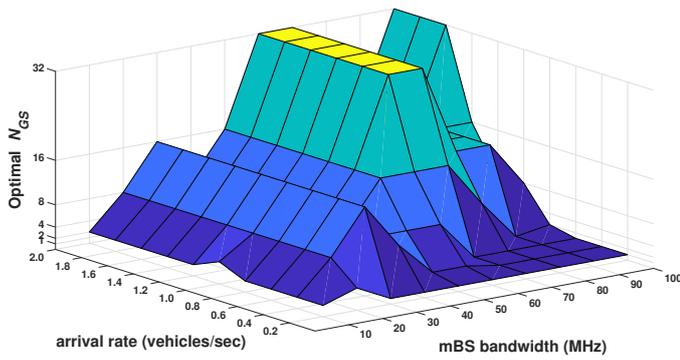


Fig. 13. Optimal N_{GS} minimizing the system-wide cost under heterogeneous input data distributions.

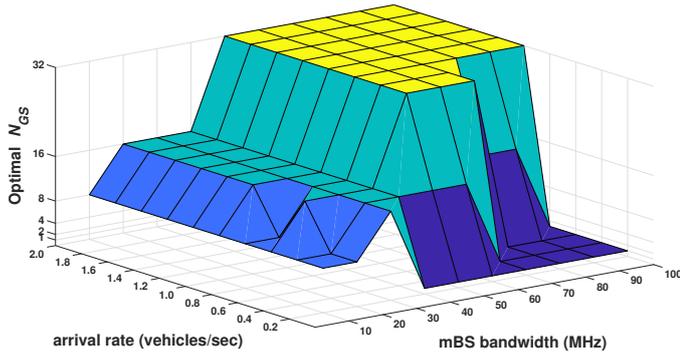


Fig. 14. Optimal N_{GS} minimizing the system-wide cost under homogeneous input data distributions.

of game slices increases as more mBS bandwidth is given while experiencing sudden drops at 40 MHz (for the arrival rates from 0.4 to 2.0) and around 80–90 MHz (for the arrival rates from 1.0 to 2.0). In fact, such drops are rooted at the randomness of the system-wide cost due to the heterogeneous input data distributions. More specifically, the system-wide cost at 40 MHz turns out almost indistinguishable between $N_{GS} = 8$ and $N_{GS} = 16$ (only up to 0.83% difference) thus making the optimal N_{GS} to be 8 while at 30 and 50 MHz it is 16. The sudden drop around 80–90 MHz occurs due to the same reason. With homogeneous distribution, however, no such dropping happens as shown in Fig. 14, since there exists no randomness in input data distributions.

Note that the results in Figs. 13 and 14 can be used to establish optimal offloading service provisioning with minimal system-wide cost.

VIII. CONCLUSION

This paper considered multi-interface computation offloading in 5G vehicular networks, and proposed how to model it as an ordinal potential game. We further considered game slicing and derived the potential function to justify the potential game formulation, and proposed an offloading decision game algorithm with finite-time convergence to the NE. We have analyzed the upperbound of the decision iterations towards convergence and the price of anarchy of the proposed game. Then, our extensive simulations have shown that our algorithm converges to an NE in finite time, which is as good as the centralized optimal solution. It has also been shown that the

system-wide cost can be minimized by finding the optimal number of game slices according to various traffic conditions.

REFERENCES

- [1] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," in *Proc. IEEE ISCC*, 2012.
- [2] P. Sroka and A. Kliks, "Towards edge intelligence in the automotive scenario: A discourse on architecture for database-supported autonomous platooning," *J. Commun. Netw.*, vol. 24, no. 2, pp. 192–208, 2022.
- [3] L. Yang, J. Cao, H. Cheng, and Y. Ji, "Multi-user computation partitioning for latency sensitive mobile cloud applications," *IEEE Trans. Comput.*, vol. 64, no. 8, pp. 2253–2266, 2014.
- [4] E. Meskar, T. D. Todd, D. Zhao, and G. Karakostas, "Energy aware offloading for competing users on a shared communication channel," *IEEE Trans. Mobile Comput.*, vol. 16, no. 1, pp. 87–96, 2016.
- [5] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7944–7956, 2019.
- [6] Z. Wang, S. Zheng, Q. Ge, and K. Li, "Online offloading scheduling and resource allocation algorithms for vehicular edge computing system," *IEEE Access*, vol. 8, pp. 52 428–52 442, 2020.
- [7] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. Leung, "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2651–2664, 2018.
- [8] G. Cui *et al.*, "Interference-aware game-theoretic device allocation for mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 21, no. 11, pp. 4001–4012, 2022.
- [9] N. Cheng, N. Lu, N. Zhang, X. S. Shen, and J. W. Mark, "Vehicular WiFi offloading: Challenges and solutions," *Veh. Commun.*, vol. 1, no. 1, pp. 13–21, 2014.
- [10] A. Tassi, M. Egan, R. J. Piechocki, and A. Nix, "Modeling and design of millimeter-wave networks for highway vehicular communication," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10 676–10 691, 2017.
- [11] X. Ge, H. Cheng, M. Guizani, and T. Han, "5G wireless backhaul networks: Challenges and research advances," *IEEE Netw.*, vol. 28, no. 6, pp. 6–11, 2014.
- [12] U. Siddique, H. Tabassum, E. Hossain, and D. I. Kim, "Wireless backhauling of 5G small cells: Challenges and solution approaches," *IEEE Wireless Commun.*, vol. 22, no. 5, pp. 22–31, 2015.
- [13] Q.-V. Pham, L. B. Le, S.-H. Chung, and W.-J. Hwang, "Mobile edge computing with wireless backhaul: Joint task offloading and resource allocation," *IEEE Access*, vol. 7, pp. 16 444–16 459, 2019.
- [14] V. Bobrovs, S. Spolitis, and G. Ivanovs, "Latency causes and reduction in optical metro networks," in *Optical Metro Netw. Short-Haul Syst. VI*, vol. 9008, pp. 91–101, 2014.
- [15] K. Zhang *et al.*, "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.
- [16] O. N. C. Yilmaz, O. Teyeb, and A. Orsino, "Overview of LTE-NR dual connectivity," *IEEE Commun. Mag.*, vol. 57, no. 6, pp. 138–144, 2019.
- [17] M. Agiwal, H. Kwon, S. Park, and H. Jin, "A survey on 4G-5G dual connectivity: Road to 5G implementation," *IEEE Access*, vol. 9, pp. 16 193–16 210, 2021.
- [18] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, 2018.
- [19] H. Guo, J. Liu, and J. Zhang, "Computation offloading for multi-access mobile edge computing in ultra-dense networks," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 14–19, 2018.
- [20] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, 2014.
- [21] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, 2016.
- [22] S. Chu *et al.*, "Efficient multi-channel computation offloading for mobile edge computing: A game-theoretic approach," *IEEE Trans. Cloud Comput.*, vol. 10, no. 3, pp. 1738–1750, 2022.
- [23] Q. Luo, C. Li, T. H. Luan, W. Shi, and W. Wu, "Self-learning based computation offloading for internet of vehicles: Model and algorithm," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 5913–5925, 2021.

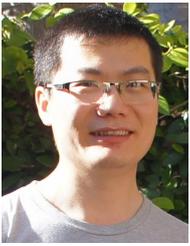
- [24] S. Wang *et al.*, “Doppler shift and coherence time of 5G vehicular channels at 3.5 GHz,” in *Proc. IEEE AP-S/URSI*, 2018.
- [25] V. Va, J. Choi, and R. W. Heath, “The impact of beamwidth on temporal channel variation in vehicular channels and its implications,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5014–5029, 2017.
- [26] E. Hossain, M. Rasti, H. Tabassum, and A. Abdelnasser, “Evolution towards 5G multi-tier cellular wireless networks: An interference management perspective,” *arXiv preprint arXiv:1401.5530*, 2014.
- [27] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, “Computation offloading and resource allocation in wireless cellular networks with mobile edge computing,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924–4938, 2017.
- [28] P. Mach and Z. Becvar, “Mobile edge computing: A survey on architecture and computation offloading,” *IEEE Commun. Surveys Tuts*, vol. 19, no. 3, pp. 1628–1656, 2017.
- [29] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. Leung, “Energy efficient computation offloading for multi-access MEC enabled small cell networks,” in *Proc. IEEE ICC Workshops*, 2018.
- [30] V. Va, T. Shimizu, G. Bansal, and R. W. Heath, “Beam design for beam switching based millimeter wave vehicle-to-infrastructure communications,” in *Proc. IEEE ICC*, 2016.
- [31] M. G. Kibria, K. Nguyen, G. P. Villardi, K. Ishizu, and F. Kojima, “Next generation new radio small cell enhancement: Architectural options, functionality and performance aspects,” *IEEE Wireless Commun.*, vol. 25, no. 4, pp. 120–128, 2018.
- [32] H. Peng, K. Moriwaki, and Y. Suegara, “Macro-controlled beam database-based beamforming protocol for LTE-WiGig aggregation in millimeter-wave heterogeneous networks,” in *Proc. IEEE VTC-Spring*, 2016.
- [33] Y. Niu, Y. Li, D. Jin, L. Su, and A. V. Vasilakos, “A survey of millimeter wave communications (mmWave) for 5G: Opportunities and challenges,” *Wireless Netw.*, vol. 21, no. 8, pp. 2657–2676, 2015.
- [34] S. Wang, J. Huang, X. Zhang, H. Kim, and S. Dey, “X-array: Approximating omnidirectional millimeter-wave coverage using an array of phased arrays,” in *Proc. ACM MobiCom*, 2020.
- [35] J. Huang, H. Gao, S. Wan, and Y. Chen, “AoI-aware energy control and computation offloading for industrial IoT,” *Future Generation Comput. Syst.*, vol. 139, pp. 29–37, 2023.
- [36] S. Josilo and G. Dán, “Computation offloading scheduling for periodic tasks in mobile edge computing,” *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 667–680, 2020.
- [37] T. S. Rappaport, *Wireless Communications: Principles and Practice (Second Edition)*. Prentice Hall, 2002.
- [38] J. Zhang, A. Beletchi, Y. Yi, and H. Zhuang, “Capacity performance of millimeter wave heterogeneous networks at 28GHz/73GHz,” in *Proc. IEEE GLOBECOM Workshops*, 2014.
- [39] Airfide Inc., 2019. [Online]. Available: <http://airfidenet.com>
- [40] S. Wang, J. Huang, and X. Zhang, “Demystifying millimeter-wave V2X: Towards robust and efficient directional connectivity under high mobility,” in *Proc. ACM MobiCom*, 2020.
- [41] T. S. Rappaport, R. W. Heath Jr, R. C. Daniels, and J. N. Murdock, *Millimeter wave wireless communications*. Pearson Education, 2014.
- [42] Z. Pi and F. Khan, “An introduction to millimeter-wave mobile broadband systems,” *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 101–107, 2011.
- [43] ETSI, “Intelligent transport systems (ITS); V2X applications; Part 1: Road hazard signalling (RHS) application requirements specification,” *TS 101 539-1 V1.1.1*, 2013.
- [44] 3GPP, “5G; Service requirements for enhanced V2X scenarios,” *TS 22.186 V16.2.0*, 2019.
- [45] 5GAA, “C-V2X use cases volume II: Examples and service level requirements,” *White Paper*, 2020.
- [46] ITU-R, “Guidelines for evaluation of radio interface technologies for IMT-Advanced,” *Report ITU-R M.2135-1*, 2009.
- [47] D. Monderer and L. S. Shapley, “Potential games,” *Games Economic Behavior*, vol. 14, no. 1, pp. 124–143, 1996.
- [48] Y. H. Chew *et al.*, *Potential Game Theory: Applications in Radio Resource Allocation*. Springer Publishing Company, Inc., 2016.
- [49] G. Pocovi, H. Shariatmadari, G. Berardinelli, K. Pedersen, J. Steiner, and Z. Li, “Achieving ultra-reliable low-latency communications: Challenges and envisioned system enhancements,” *IEEE Netw.*, vol. 32, no. 2, pp. 8–15, 2018.
- [50] T. Roughgarden, *Selfish routing and the price of anarchy*. MIT press Cambridge, 2005.
- [51] F. Richter, A. J. Fehske, and G. P. Fettweis, “Energy efficiency aspects of base station deployment strategies for cellular networks,” in *Proc. IEEE VTC-Fall*, 2009.
- [52] T. Bai and R. W. Heath, “Coverage and rate analysis for millimeter-wave cellular networks,” *IEEE Trans. Wireless Commun.*, vol. 14, no. 2, pp. 1100–1114, 2014.
- [53] ETSI, “5G; NR; User equipment (UE) radio transmission and reception; Part 2: Range 2 Standalone,” *ETSI TS 138 101-2 V15.2.0*, 2018.
- [54] T. S. Rappaport *et al.*, “Millimeter wave mobile communications for 5G cellular: It will work!” *IEEE Access*, vol. 1, pp. 335–349, 2013.
- [55] Y. Wen, W. Zhang, and H. Luo, “Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones,” in *Proc. IEEE INFOCOM*, 2012.
- [56] A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clients in cloud computing,” in *Proc. USENIX HotCloud*, 2010.
- [57] P. A. Lopez *et al.*, “Microscopic traffic simulation using SUMO,” in *Proc. IEEE ITSC*, 2018. [Online]. Available: <https://elib.dlr.de/124092/>



Suhwan Jung is a Ph.D. student at the Department of Electrical Engineering (EE), the Ulsan National Institute of Science and Technology (UNIST), Ulsan, Korea, since September 2019. He received his B.S. degree in the School of Electrical and Computer Engineering (ECE) from the UNIST in 2017, and his M.S. degree in the Department of Electrical Engineering (EE) from the UNIST in 2019. His research interest includes 5G/6G, V2X, and WLAN Multi-Link Operation (MLO).



Hyoil Kim is a Professor at the Department of Electrical Engineering (EE), the Ulsan National Institute of Science and Technology (UNIST), Ulsan, Korea. He received his B.S. degree in Electrical Engineering from Seoul National University, Seoul, Korea in 1999, and M.S. and Ph.D. degrees in Electrical Engineering: Systems from the University of Michigan in 2005 and 2010, respectively. Before joining the UNIST in 2011, he was a Postdoctoral Researcher in the IBM T.J. Watson Research Center, Hawthorne, NY, USA in 2010–2011. His research interest includes AI-empowered networking, 5G/6G, WLAN, V2X, Hyperloop, and Dynamic Spectrum Access (DSA). He has served as a TPC Member for various international conferences, including IEEE INFOCOM (2014, 2016–2017, 2021–2024), IEEE SECON (2020–2023), IEEE Globecom (2011–2015, 2017–2018), and also served as a Local Arrangement Co-Chair of ACM MobiHoc 2022. In IEEE INFOCOM 2022, he has been awarded as a Distinguished Member of the TPC. Currently, he is an Area Editor of Computer Networks (ComNet) and an Editor of the Journal of Communications and Networks (JCN).



Xinyu Zhang is an Associate Professor with the Department of Electrical and Computer Engineering, University of California San Diego. He received the Ph.D. degree from the University of Michigan in 2012. Prior to joining the University of California San Diego in 2017, he was an Assistant Professor at the University of Wisconsin–Madison. His research interests include wireless systems and ubiquitous computing. He was a Recipient of two ACM MobiCom Best Paper Award in 2011 and 2020, the Communications of the ACM Research Highlight in 2018, the ACM SIGMOBILE Research Highlight in 2018, the NSF CAREER Award in 2014, the Google Research Award in 2017, 2018, and 2020, and the Sony Research Award in 2018 and 2020. He served as the TPC Chair for ACM MobiCom 2019, IEEE SECON 2017, a Co-Chair for NSF millimeter-wave research coordination network, and an Associate Editor for IEEE TRANSACTIONS ON MOBILE COMPUTING from 2017 to 2020.



Sujit Dey is a Professor in the Department of Electrical and Computer Engineering, the Director of the Center for Wireless Communications, and the Director of the Institute for the Global Entrepreneur at University of California, San Diego. He heads the Mobile Systems Design Laboratory, developing innovative and sustainable edge computing, networking and communications, multi-modal sensor fusion, and deep learning algorithms and architectures to enable novel 5G and 6G use cases like predictive personalized health, X-Reality, smart mobility and smart manufacturing. He has created inter-disciplinary programs involving multiple UCSD schools as well as community, city and industry partners; notably the Connected Health Program in 2016 and the Smart Transportation Innovation Program in 2018. In 2017, he was appointed as an Adjunct Professor, Rady School of Management, and the Jacobs Family Endowed Chair in Engineering Management Leadership. Dr. Dey served as the Faculty Director of the von Liebig Entrepreneurism Center from 2013-2015, and as the Chief Scientist, Mobile Networks, at Allot Communications from 2012-2013. In 2015, he co-founded igrenEnergi, providing intelligent battery technology and solutions for EV mobility services. He founded Ortiva Wireless in 2004, where he served as its founding CEO and later as CTO and Chief Technologist till its acquisition by Allot Communications in 2012. Prior to Ortiva, he served as the Chair of the Advisory Board of Zyray Wireless till its acquisition by Broadcom in 2004, and as an advisor to multiple companies including ST Microelectronics and NEC. Prior to joining UCSD in 1997, he was a Senior Research Staff Member at NEC C&C Research Laboratories in Princeton, NJ. He received his Ph.D. in Computer Science from Duke University in 1991. Dr. Dey has co-authored more than 250 publications, and a book on low-power design. He holds 18 U.S. and 2 international patents, resulting in multiple technology licensing and commercialization. He has been a Recipient of nine IEEE/ACM Best Paper Awards, and has chaired multiple IEEE conferences and workshops. Dr. Dey is a Fellow of the IEEE.