Joint Offloading and IEEE 802.11p-Based Contention Control in Vehicular Edge Computing

VanDung Nguyen, *Member, IEEE*, Tran Trong Khanh, Nguyen H. Tran, *Senior Member, IEEE*, Eui-Nam Huh, *Member, IEEE*, and Choong Seon Hong, *Senior Member, IEEE*

***Abstract*—One key element towards improving network performance with lower computing latency and better quality of service in vehicular edge computing (VEC) is offloading, which allows computation-intensive mobile applications to offload their tasks to VEC servers. In vehicular ad-hoc networks, VEC can offload the computation tasks to the road-side unit (RSU), which improves vehicular service and reduces energy consumption of the vehicle. In this letter, we investigate the impact of transmission time between vehicles and RSUs, which significantly impacts the offloading decision. In particular, due to the high mobility topology and IEEE 802.11p medium access control protocol, we propose an algorithm design of joint contention window control and offloading decision formulated as a mixed-integer non-linear problem to maximize system utility. The numerical results show that the proposed algorithm significantly outperforms benchmark policies in terms of the total processing completion time.**

***Index Terms*—Vehicular edge computing, task offloading, contention window, IEEE 802.11p.**

I. Introduction

Vehicle edge computing (VEC) is a promising new technology that integrates computing resources in order to process computational tasks using vehicular driving systems, and onboard mobile devices to offer computing services for pedestrians. Applications such as autonomous driving and vehicular video streaming require enormous tasks, high computational complexity, and tight delay sensitivity [1]. Because vehicles have substantial computing and storage resources, vehicles and road-side units (RSUs) can contribute their computing resources to vehicle networks, thereby serving as the main components of VEC.

The computing task is a process where vehicles offload their computation-heavy and latency-sensitive tasks to RSUs for edge execution. By processing the computational task near the vehicle and considering the characteristic of vehicle networks, VEC can reduce the computation response time and alleviate traffic congestion problems in capacity-constrained backhaul links. Based on dependency applications (i.e., an augmented reality) or independence applications (i.e., virus scans and image compression), each task can either be processed locally or offloaded from the vehicle to a VEC server. The benefits of using a VEC for these tasks are as follows: 1) low total processing completion time due to limited the data passing through the network, 2) increased quality of service for various application requirements in vehicular ad-hoc networks (VANETs) [2], and 3) lower communication delay due to the proximity between vehicles and VEC servers.

Although task offloading decisions in VEC can alleviate network burden and reduce packet transmission delay, there remain critical challenges. First, the traditional offloading algorithm design depends on human experience [3], and the network topology of the vehicular environment is highly dynamic due to the high mobility of vehicles. Therefore, vehicular environment analysis is required for accuracy and real-time evaluation, thereby making task optimization crucial. Second, it is difficult to characterize the time variance between channels in urban vehicular environments due to existing obstacles, such as trees and buildings. Therefore, management of the dynamic resource demands, complicated traffic environments, and diverse application characteristics is challenging.

Approaches have been proposed to determine optimal offloading decisions. Binary offloading was proposed to minimize energy consumption and computation latency. This method accelerates independent decisions by users regarding the local execution of tasks or task offloading to edge servers [4]. To maximize the utilities of vehicles and computing servers, each vehicle selects one VEC server as the target offload server [5]; however, this results in all vehicles selecting the VEC server with the highest utility [6]. Moreover, these approaches exclude wireless technology in a vehicular environment, which does not comply with the IEEE 802.11p standard. A previous study [7] considered computation offloading based on IEEE 802.11p; however, the trade-off between vehicle density and the contention window (CW) was excluded. Another study [8] proposed a VEC framework (autonomous vehicular edge (AVE)) to increase the computational capabilities of vehicles without requiring the deployment of a particular infrastructure. Additionally, based on the time consuming nature of computational task and vehicle mobility, Zhang *et al.* [9] designed an efficient predictive combination-mode relegation scheme to reduce

Manuscript received December 22, 2019; accepted February 27, 2020. Date of publication March 4, 2020; date of current version July 9, 2020. This work was supported in part by the Institute for Information & Communications Technology Planning & Evaluation grant funded by the Korea Government (MSIT, Service Mobility Support Distributed Cloud Technology) under Grant 2017-0-00294, and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) under Grant NRF-2017R1A2A2A05000995. The associate editor coordinating the review of this article and approving it for publication was H. Zhou. *(Corresponding authors: Eui-Nam Huh; Choong Seon Hong.)*

VanDung Nguyen, Tran Trong Khanh, Eui-Nam Huh, and Choong Seon Hong are with the Department of Computer Science and Engineering, Kyung Hee University, Yongin 446-701, South Korea (e-mail: ngvandung85@khu.ac.kr; khanhtrannnn@khu.ac.kr; johnhuh@khu.ac.kr; cshong@khu.ac.kr).

Nguyen H. Tran is with the School of Computer Science, University of Sydney, Sydney, NSW 2006, Australia (e-mail: nguyen.tran@sydney.edu.au).

Digital Object Identifier 10.1109/LWC.2020.2978482

computational costs and improve task transmission efficiency by focusing on computation transfer strategies via vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communication modes without consideration of the IEEE 802.11p standard. Moreover, Zhou *et al.* [10] proposed a consensus alternating direction method of multipliers (ADMM) based energy-efficient resource-allocation algorithm to achieve significant improvements in energy savings. Furthermore, although some approaches applied medium access control (MAC) addresses for the vehicular network [4]–[10], they did not consider contentions with IEEE 802.11p standard.

VANETs connect and exchange information through V2V or vehicle-to-RSU (V2R) communication mode [11], with MAC addresses playing a vital role in providing efficient and fair access to a wireless network between vehicles and providers. IEEE 802.11p [12] represents the legacy IEEE 802.11 standard that supports VANETs and the standard deployed for vehicular communication. The IEEE 802.11p-based MAC protocol uses a priority-based access scheme that employs both enhanced distributed-channel access and carrier-sense multiple access with collision avoidance (CSMA/CA) mechanisms to promote driving safety and comfort.

The intermittent connectivity between vehicles and RSUs remains a critical challenge. In VEC, vehicles can access the channels to transmit their tasks to RSUs [12]; however, the rapid movement of vehicles leads to frequent topology changes, resulting in an easy disconnection of links that leads to deterioration of communication quality. Additionally, conventional static decision-making schemes result in frequent offloading failures due to connectivity issues between vehicles and RSUs before the completion of the workload-data upload. Moreover, task transmission time in V2R communication is used to calculate computation-intensive tasks; therefore, reducing access delays on the MAC layer requires consideration. Recent research has increasingly focused on addressing joint computation and transmission time, as well as the use of VEC in vehicle networks.

The main contribution of this letter is three-fold: i) we investigate the impact of transmission time between vehicle and RSUs to offloading decisions; ii) we address the V2R transmission time problem to maximize the system utility by designing an adaptive joint CW and offloading-based scheme as a mixed-integer nonlinear problem; iii) under various vehicle density conditions, with simulation results, we show that this method can reduce the processing time for each RSU coverage area. Additionally, the algorithm complies with the IEEE 802.11p standard used by VANETs.

II. System Model

A. VEC Network

The VANET under consideration comprises a set of RSUs and a set of vehicles moving on a unidirectional road (Fig. 1). Each RSU is equipped with a VEC server with limited computing resources. The set of RSUs is denoted as $\mathscr{M} = \{1, \ldots, M\}$, with the RSUs deployed along a unidirectional road. Based on VEC architecture [1], [7], under a variety of the vehicular environments and the coverage area of $\mathscr{M}$ RSUs with exhibiting different transmission strengths, the road is

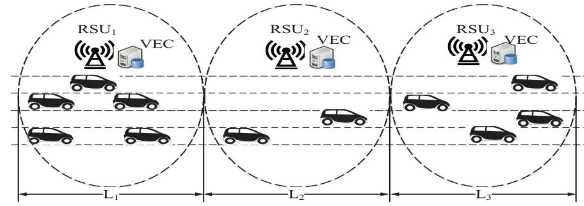


Fig. 1. VEC offloading in a vehicular network.

divided into *M* segments of different size, $\{L_1, L_2, \ldots, L_M\}$, respectively. A vehicle communicates only with RSU_m when it is located in segment *m*, where $m \in \mathscr{M}$.

We assume that *N* vehicles are arriving at the starting point and running at speed *v*. By denoting the set of vehicles as $\mathscr{N} = \{1, \ldots, N\}$, each vehicle has a computational task, described as $D_i \triangleq (d_i, c_i, T_i^{max})$, $i \in \mathscr{N}$, where d_i is the size of the computational task, c_i denotes the required computational resources for computing task D_i, and T_i^{max} is the maximum latency required to accomplish the task. For the selection decision, we denote $x_{ij} \in \{0, 1\}$ (i.e., if vehicle *i* chooses the VEC server on RSU *j* as the target offloading server and offloads its task to this server, $x_{ij} = 1$; otherwise $x_{ij} = 0$.)

We divide the computation-intensive task into three major time components: movement time, wireless transmission time, and computation time. First, vehicle *i* moves from the starting point to the coverage area of RSU *j*, with the movement time represented as $(\sum_{k=1}^{j-1} L_k)/v$. Second, the vehicle sends its task from vehicle *i* to RSU *j* based on the IEEE 802.11p MAC protocol. Finally, the computation time depends on the allocated computational resources, task size, and the CW affects the transmission time between a vehicle and RSU based on IEEE 802.11p. Therefore, modeling the transmission time is essential.

B. Transmission Time

The IEEE 802.11p standard [12] supports VANETs by providing V2V and V2R communications and supporting transmission rates from 3 Mb/s to 27 Mb/s over a bandwidth of 10 MHz [13]. Additionally, IEEE 802.11p uses the CSMA/CA protocol for packet transmission; therefore, the offloading task uses request-to-send/clear-to-send (RTS/CTS) to reduce collisions. We assume that each vehicle or RSU has one transceiver, which is dedicated to short-range communication (DSRC) devices in vehicular communication networks [9]. Moreover, IEEE 802.11p multiple access mechanism (CSMA/CA) arranges each station to attempt random access only when the channel is sensed not in use by another station.

As previously described [14], the probability, τ, of vehicle *i* transmitting its task to RSU *j* in a randomly chosen slot time when the backoff time counter is equal to zero is as follows:

$$\tau = \frac{2(1-2p)}{(1-2p)(W+1) + pW(1-(2p)^b)}. \quad (1)$$

where *p* refers to that, when one particular vehicle is transmitting, the probability that other vehicles collide with the given vehicle; *b* is the maximum backoff stage, and *W* is the maximum CW of backoff stage 0. The probability p is given as

$$p = 1 - (1-\tau)^{N_j - 1}. \tag{2}$$

where N_j is the number of vehicles, and determined by the VEC server on RSU j, $N_j = \sum_{i=1}^{N} x_{ij}$. Consequently, τ and p are solved using numerical methods [14]. Note that $0 < p < 1$ and $0 < \tau < 1$.

Let P_{ij}^{suc} be the probability of successful transmission of an offloaded task, and P_{ij}^{col} be the collided transmission. When the channel is idle with probability P_{ij}^{idle}, we have:

$$\begin{cases} P_{ij}^{idle} = (1-\tau)^{N_j}, \\ P_{ij}^{suc} = N_j \tau (1-\tau)^{N_j - 1}, \\ P_{ij}^{col} = 1 - (1-\tau)^{N_j} - N_j \tau (1-\tau)^{N_j - 1}. \end{cases} \tag{3}$$

Let T_{ij}^{idle}, T_{ij}^{col}, and T_{ij}^{suc} denote the duration of a free time slot, the duration of a transmission collision, and the duration of a successful reservation, respectively. As previously described [15], we have

$$\begin{cases} T_{ij}^{idle} = \sigma, \\ T_{ij}^{suc} = AIFS + RTS + \delta + SIFS + CTS + \delta \\ \qquad + SIFS + H + T_{ij}^{d_i} + \delta + SIFS + ACK + \delta, \\ T_{ij}^{col} = RTS + AIFS + \delta. \end{cases} \tag{4}$$

where H denotes the overhead of the packet header, *SIFS* is the interval of the short inter-frame space, *ACK* is the time of acknowledgement, *AIFS* is the interval of the arbitration inter-frame space, *RTS* is the *RTS* transmission time, *CTS* is the *CTS* transmission time, σ is the duration of a time slot, and δ is the propagation time. Note that $T_{ij}^{d_i}$ is a transmission delay of task d_i taking on the wireless channel, given as

$$T_{ij}^{d_i} = \frac{d_i}{B \log(1 + P_i \frac{g_{ij}}{N_0})}. \tag{5}$$

where B is the RSU bandwidth, P_i is the transmission power of vehicle i, g_{ij} is the channel gain between vehicle i and RSU j, and N_0 is the noise power [1], [7].

Let S_{ij} be the normalized throughput:

$$S_{ij} = \frac{P_{ij}^{suc} d_i}{P_{ij}^{idle}\sigma + P_{ij}^{suc} T_{ij}^{suc} + P_{ij}^{col} T_{ij}^{col}}. \tag{6}$$

Because the throughput is represented as a data rate, the transmission time for offloading task d_i from vehicle i to RSU j is given as

$$T_{ij}^{t} = \frac{d_i}{S_{ij}} = T_{ij}^{suc} + \frac{P_{ij}^{idle}\sigma + P_{ij}^{col} T_{ij}^{col}}{P_{ij}^{suc}}. \tag{7}$$

We consider a simple scenario of one RSU, a various number of vehicles, N, and the maximum backoff stage, b. The relationships between transmission time, N and b are shown in Fig. 2, and simulated using the communication parameters described in Section IV. Table I shows the default contention set according to different IEEE 802.11 standards, and b can vary from 1 to 6. Similar to a previous study [12], we vary b from 1 to 5. As shown in Fig. 2, we determined that the high vehicle density and maximum backoff stage, b, affect the transmission time. For example, when the number of vehicles is 80, the average transmission time at $b = 3$ is higher than that at $b = 5$ (Fig. 2). To obtain high performance, we need to determine the optimal (minimal) transmission time. In this case, the RSU chooses $b = 5$, which corresponds to the optimal average transmission time (0.67 ms). Therefore, under a specific vehicle density, non-optimal b can lead to longer transmission time, and thus, b should be carefully optimized.

TABLE I
IEEE 802.11 DEFAULT CW SET

AC	802.11b/ax CW_min	802.11b/ax CW_max	802.11g/a/n/e /p/ax CW_min	802.11g/a/n/ e/p/ax CW_max
AC_VO	7	15	3	7
AC_VI	15	31	7	15
AC_BE	31	1023	15	1023
AC_BK	31	1023	15	1023

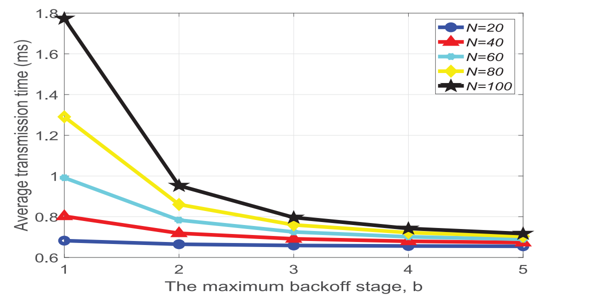


Fig. 2. The average transmission time versus the maximum backoff stage.

C. Computation Time

When vehicle i offloads its task to the VEC server on RSU j, the computational resource of the VEC server is limited, because N_j vehicles want to offload their tasks to the server on RSU j. Therefore, the computational resource for N_j must be allocated. Let f_{ij} be the amount of computational resources that the VEC server on RSU j assigns to vehicle i. Note that $\sum_{i=1}^{N} x_{ij} f_{ij} < F_j$, where F_j is the total computation time of the VEC server on RSU j, and the computation time, T_{ij}^{VEC}, of the VEC server on RSU j is given as

$$T_{ij}^{\text{VEC}} = \frac{c_i}{f_{ij}}. \tag{8}$$

D. System Utility Function

Task processing delay comprises three components: movement time, transmission time, and computation time on the VEC server. Therefore, this delay is calculated as follows:

$$T_i = \sum_{j=1}^{M} x_{ij} \left(\left(\sum_{k=1}^{j-1} L_k \right) / v + T_{ij}^{t} + T_{ij}^{\text{VEC}} \right). \tag{9}$$

According to previous studies [5], [6], [7], the utility function balances the load among VEC servers to achieve proportional fairness, which is defined as

$$U_i = \alpha \log(1 + \beta - T_i). \tag{10}$$

where α is the satisfaction parameter, and β normalizes the satisfaction of being non-negative [5], [6], [7].

III. Problem Formulation and Solution

We formulated the joint CW and offloading problem as a mixed-integer non-linear programming problem to maximize system utility. Let $\mathbf{x} = \{x_{ij}\}$ be a set vector of decisions made by VEC server selection, $\mathbf{f} = \{f_{ij}\}$ be the computational resource vector, and $\mathbf{b} = \{b_j\}$ be the maximum backoff stage among vehicles moving within coverage of RSU j, $b_j \in \mathbb{N} \setminus \{0\}$. We formulate the optimization problem, as follows:

$$\max_{\mathbf{x},\mathbf{f},\mathbf{b}} \quad \sum_{i=1}^{\mathscr{N}} \alpha \log(1+\beta - T_i) \tag{11a}$$

$$\text{subject to} \quad T_i \leq T_i^{max}, \quad \forall i \in \mathscr{N}, \tag{11b}$$

$$\sum_{j=1}^{M} x_{ij} = 1, \quad \forall i \in \mathscr{N}, \tag{11c}$$

$$\sum_{i=1}^{N} x_{ij} f_{ij} \leq F_j, \quad \forall j \in \mathscr{M} \tag{11d}$$

$$0 \leq f_{ij} \leq F_j, \quad \forall i \in \mathscr{N}, \forall j \in \mathscr{M}, \tag{11e}$$

$$x_{ij} \in \{0,1\}, \quad \forall i \in \mathscr{N}, \forall j \in \mathscr{M} \cup \{0\}, \tag{11f}$$

$$0 \leq b_j \leq 5, \forall j \in \mathscr{M}, b_j \in \mathbb{N} \setminus \{0\}. \tag{11g}$$

The first constraint (11b) guarantees that the task processing delay cannot exceed the maximum allowed latency, T_i^{max}. Constraints (11c) and (11f) state that each vehicle offloads its task to one VEC server. With constraints (11d) and (11e), the sum of the computational resource is assigned to all tasks, and ensures that the VEC server on RSU *j* does not exceed its total computation capacity. Constraint (11e) guarantees that CW_i cannot exceed the maximum contention window ($CW_{max} = 1023 = 2^b CW$) corresponding to $b = 5$ ($CW = 32$) [12]. The solution to this problem follows determination of an **x** given **f** and **b**, followed by **b** and **f** according to **x**, with this process repeating until convergence. As described previously [7], the optimal **x*** and **f*** are determined, and the number of vehicles chosen by each VEC server on an RSU is obtained. The **b*** is chosen to minimize the transmission time:

$$\min_{\mathbf{b}} \quad T_{ij}^{suc} + \frac{P_{ij}^{idle}\sigma + P_{ij}^{col} T_{ij}^{col}}{P_{ij}^{suc}} \tag{12a}$$

$$\text{subject to} \quad 0 \leq b_j \leq 5, \forall j \in \mathscr{M}, b_j \in \mathbb{N} \setminus \{0\}. \tag{12b}$$

The optimal maximum CW **b*** is obtained using linear programming. Based on above analysis, our joint optimization for server selection, resource computation, and contention window algorithm are summarized in Algorithm 1. Note that **b** depends only on **x**; therefore, we update **b** after obtaining **x** based on Eq. (12b). We observe the computation resource allocation on each vehicle by plotting the convergence evolution of the loop of step 10 in Algorithm 1, as shown in Fig. 3 under different initial $\mathbf{f^{(0)}}$ points. Similar to the JSCO algorithm [7], the optimal **x***, **f***, and **b*** converge within 10 iterations.

IV. Simulation Results

We simulated our proposed algorithm using a MATLAB event-driven simulator and implemented a straight highway traffic model [16] to aid in simulation creation. The simulation parameters are summarized in Table II. For regular highway traffic, the arrival process of the vehicles is a normal distribution, and the number of vehicles on the highway segment remains constant during the simulation time. In the highway traffic model, every vehicle is running at a constant speed (60.300 *km/h*). The computational resources of the VEC servers from RSUs #1 to #5 are 5 GHz,10 GHz,15 GHz,20 GHz and 25 GHz, respectively. We assumed that d_i, c_i, and T_i^{max} were uniformly distributed in the range of *U*[100, 300] KB, *U*[0.5, 1.5] GHz, and *U* [8, 10], respectively. We calculated the average of 200 runs for a random highway traffic model.

We considered that vehicles were moving at a constant speed of 120 *km/h*, as shown in Figs. 4a and 4b. The average number of vehicles corresponding to $\overline{N_i} = \overline{\sum_{j=1}^{N} x_{ij}^*}$, which is the average number chosen by the VEC_i, $i \in [1, 5]$ is shown in Fig. 4a indicating that most of the vehicles offloaded their tasks onto the VEC servers of RSU {2, 3, 4}, because the vehicles in the coverage of RSU #1 could choose the next RSU to offload their tasks based on their movements. This guaranteed strict delay sensitivity and limited computational capacity.

Algorithm 1: Joint Optimization for Selection, Computation and Contention Window Algorithm

```
1  Initialization:
2      Set selection decision x^(0);
3      Set computation resource f^(0);
4      Set contention window b^(0);
5      Set the number of iteration: k = 0;
6  Repeat
7      Based on f^(k-1) and b^(k-1), each vehicle computes
       its selection decision, x^(k), based on rounding
       method [7];
8      Based on x^(k), calculate b^(k) by using linear
       programming to solve (12b);
9      Repeat
10         Calculate f^(k) by using Lagrangian method to
           solve the computation resource allocation
           problem for a given x^(k);
11     Until
12         Convergence.
13     k = k + 1;
14 Until
15     Convergence.
```

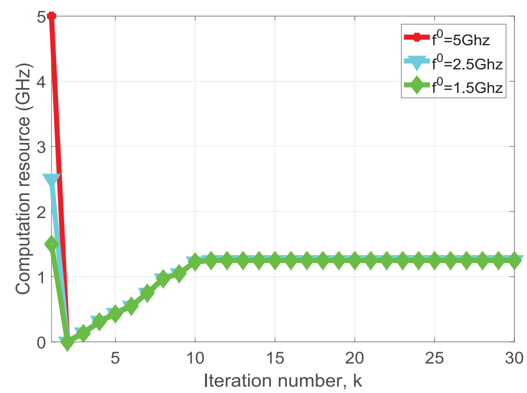
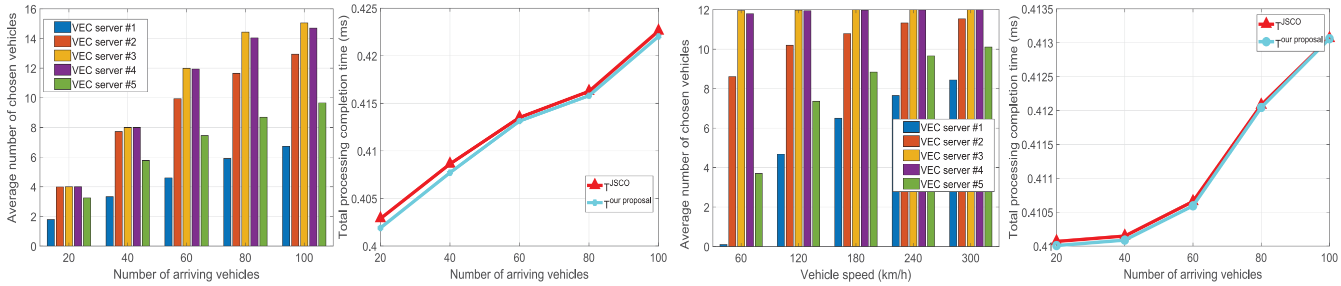


Fig. 3. Convergence of the computation resource allocation with $N = 60veh$.



(a) $\#$ vehicles, $v = 120km/h$ (b) Processing time, $v = 120km/h$ (c) $\#$ vehicles, $N = 60veh$ (d) Processing time, $N = 60veh$

Fig. 4. Average number of chosen vehicles, processing time for the number of arriving vehicles, and vehicle speeds.

TABLE II
COMMUNICATION PARAMETERS [1], [7]

Parameters	*Value*	*Parameters*	*Value*
Number of RSUs	5	Highway length	100 m
Bandwidth of each RSU	10 MHz	Transmission power	30 dBm
Noise power	-114 dBm	H	107μs
SIFS	10μs	AIFS	50μs
RTS	144μs	CTS	120μs
ACK	101μs	Slot time σ	13 μs
Propagation time δ	1 μs	W	32

Fig. 4b shows that processing time increases with respect to the number of vehicles. Moreover, the results shown in Fig. 4b demonstrated the different processing times between our algorithm and the JSCO-modified algorithm [7], which is defined as the JSCO algorithm with binary offloading and executed at a vehicle or VEC server. Note that the JSCO-modified algorithm used a maximum backoff stage of 3 ($b = 3$). Based on the number of vehicles, our algorithm calculated the optimal b as **b*** in order to reduce the processing time. Therefore, compared with the JSCO-modified algorithm (Fig. 4b), the average of total processing completion time using our algorithm was lower. For example, at a vehicle number of 80, the processing time of the JSCO-modified algorithm was higher than that of our algorithm using an **b*** of 5. When N increases from 20 to 60 vehicles, the value of **b*** may increase. Since the b can vary from 1 to 5 according to [12], when vehicle N is greater than 80, the value of **b*** does not change, $\mathbf{b}* = 5$. Additionally, the probability of channel collision also increased along with the number of vehicles [14]. Therefore, the difference in processing time between our proposal and JSCO-modified algorithm is small when vehicle N is greater than 80.

Additionally, we fixed the arriving vehicles at 60 and varied the vehicle speed (Fig. 4c), resulting in most vehicles offloading their tasks onto the VEC servers at the center of the road. Given the use of the same optimal CW **b***, and the different variance in speed of the vehicles resulted in small average differences in processing times (Fig. 4d). Consequently, vehicle speed did not affect processing time between our proposal and the JSCO-modified algorithm under a fixed number of arriving vehicles.

V. CONCLUSION

In this letter, we described an IEEE 802.11p-based method of task offloading for VEC processing by vehicles using a joint CW and addressing it as a mixed-integer non-linear programming problem to maximize system utility. We found that combining task offloading decisions with an adaptive CW reduced processing times using the IEEE 802.11p standard. Our future work will evaluate the proposed method on a more complex vehicular environment with various mobility patterns in order to achieve significant reductions in task processing time.

REFERENCES

[1] P. Liu, J. Li, and Z. Sun, "Matching-based task offloading for vehicular edge computing," *IEEE Access*, vol. 7, pp. 27628–27640, 2019.
[2] H. Hartenstein and L. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 6, pp. 164–171, Jun. 2008.
[3] Z. Xu *et al.*, "Experience-driven networking: A deep reinforcement learning based approach," in *Proc. Conf. Comput. Commun. (INFOCOM)*, Honolulu, HI, USA, Apr. 2018, pp. 1871–1879.
[4] K. Zhang *et al.*, "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.
[5] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–6.
[6] K. Zhang, Y. Mao, S. Leng, A. Vinel, and Y. Zhang, "Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks," in *Proc. 8th Int. Workshop Resilient Netw. Design Model. (RNDM)*, Halmstad, Sweden, Sep. 2016, pp. 288–294.
[7] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4377–4387, Jun. 2019.
[8] J. Feng, Z. Liu, C. Wu, and Y. Ji, "AVE: Autonomous vehicular edge computing framework with ACO-based scheduling," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10660–10675, Dec. 2017.
[9] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 36–44, Jun. 2017.
[10] Z. Zhou, J. Feng, Z. Chang, and X. Shen, "Energy-efficient edge computing service provisioning for vehicular networks: A consensus admm approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5087–5099, May 2019.
[11] V. Nguyen, C. Pham, T. Z. Oo, N. H. Tran, E.-N. Huh, and C. S. Hong, "MAC protocols with dynamic interval schemes for VANETs," *Veh. Commun.*, vol. 15, pp. 40–62, Jan. 2019.
[12] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments*, IEEE Standard 802.11p(TM)-2010, Jul. 2010.
[13] Y. Wang, X. Duan, D. Tian, G. Lu, and H. Yu, "Throughput and delay limits of 802.11p and its influence on highway capacity," *Procedia Soc. Behav. Sci.*, vol. 96, pp. 2096–2104, Nov. 2013.
[14] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.
[15] C. Han, M. Dianati, R. Tafazolli, R. Kernchen, and X. Shen, "Analytical study of the IEEE 802.11p MAC sublayer in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 873–886, Jun. 2012.
[16] H. Su and X. Zhang, "Clustering-based multichannel MAC protocols for QoS provisionings over vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3309–3323, Nov. 2007.