

# SDN-Enabled 3C Resource Integration in Green Internet of Electrical Vehicles

Handi CHEN<sup>a</sup>, Xiaojie WANG,<sup>b,1</sup> Zhaolong NING<sup>a,b</sup>, and Lei GUO<sup>b</sup>

<sup>a</sup>*School of Software, Dalian University of Technology, Dalian, Liaoning, China*

<sup>b</sup>*School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China*

**Abstract.** With the advocacy of green renewable energy, Electric Vehicles (EVs) have gradually become the mainstream in the automobile market. Due to the finite edge resources of the Internet of EVs, this paper integrates idle communication, caching and computational resources of EVs to enrich the available resources for vehicular task migration. Considering the limited capacity and resources of EVs, a distributed lightweight imitation learning-based efficient Task cOoperative migration Policy Integrating 3C resource policy, named TOPIC, is proposed to maximize the obtained quality of service. The experimental results based on the real-world traffic dataset of Hangzhou (China) demonstrate the QoS obtained based on the expert policy and agent policy of TOPIC is about 3 times higher than other representative policies.

**Keywords.** Heterogeneous resource management, software-defined network, task migration, imitation learning

## 1. Introduction

### 1.1. Background

Considering the deterioration of the ecological environment and the shortage of energy, Electric Vehicles (EVs), utilizing renewable energy [1], are gradually occupying the vehicular market. However, the contradiction between users' requirements and EVs' limited resources in the latency-sensitive scenario with energy constraints is still one of the thorniest issues to be solved. Integrating heterogeneous edge resources distributedly to overcome overloaded vehicular tasks is a promising approach to reduce the workload of devices and execution cost [2, 3].

### 1.2. Literature Review

However, the edge resources, consisting of Communication, Computation and Caching (3C), provided by Mobile Edge Computing (MEC) servers deployed in real-world scenarios are rather limited [4]. Homogeneous resource management has been studied adequately, while the study of heterogeneous resource integration is still immature. Zhou *et*

---

<sup>1</sup>Corresponding Author: Xiaojie Wang, E-mail: xiaojie.kara.wang@ieee.org.

*al.* [5] integrated caching and computational resources through designing Service Function Chain (SFC), which is a common method to integrate heterogeneous resources. The 2C models in [6, 7] jointly optimized communication and computational resources allocation. In [6], the downloading communication phase was accelerated with computation replication in MEC networks to decrease the latency of communication. Ding *et al.* [7] minimized the weighted energy consumption for ground devices by joint ground user equipment association, multi-user transmit precoding, computation task assignment and communication resource allocation. The joint optimization problem was solved based on traditional optimization methods.

For the integration of 3C resources, how to comprehensively integrate these three heterogeneous resources is rather challenging. Luo *et al.* [8] proposed an efficient Fog-enabled 3C resource sharing framework (F3C) and solve resource allocation issues based on the auxiliary graph and proposed F3C algorithm to optimize energy consumption. Tang *et al.* [9] proposed a 3C resource integration framework to motivate device cooperation and schedule resources more flexibly. However, the investigated scenarios [8, 10] are relatively static, and the dynamics of mobile devices are not taken into account in the proposed algorithms.

However, the infrastructure resources are always regarded as sufficient in traditional studies [11], which is not realistic in the real world, especially in the dynamic scenario with fierce competition. The complexity of integrating heterogeneous resources promotes studies with leveraging Software-Defined Network (SDN), improving the resource pooling level via decoupling traditional network architecture.

### 1.3. Research Challenges and Contributions

Integrating heterogeneous edge resources with SDN is a promising approach for flexible resource scheduling and network service automation, motivating our work. In addition to the complexity of managing heterogeneous resources, scheduling resources in the dynamic ultra-dense scenario has fiercer competition among EVs with limited batteries than traditional vehicles with internal combustion engines, especially for the selfish and rational users. To solve the abovementioned issues, we construct a task migration framework integrating heterogeneous resources and propose a distributed Task cOoperative migration Policy Integrating 3C resources (TOPIC) based on Imitation Learning (IL) [2] to maximize the obtained Quality of Service (QoS). The main contributions are summarized as follows: (1) We construct an SDN-enhanced intelligent, cooperative task migration framework and formulate the optimization problem; (2) We design a distributed IL-based cooperative task migration policy, TOPIC, to make a trade-off between rent and the obtained service rate; (3) The efficiencies of our proposed policy has been validated with the datasets of Hangzhou (China), compared with representative policies.

The rest of this paper is organized as follows: the system model and problem formulation are illustrated in Section 2. Section 3 specifies the TOPIC policy. Experimental results and performance evaluation are analyzed in Section 4, followed by the conclusion Section 5.

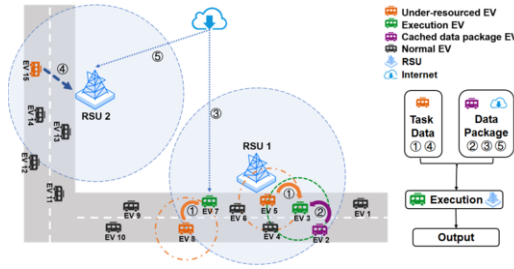


Figure 1. An SDN-enhanced architecture for 3C resource integration.

## 2. System Model and Problem Formulation

### 2.1. System Model

A dynamic vehicular network is considered as illustrated in Fig. 1. This paper investigates task migration problem via integrating 3C resources in one cluster-based vehicular network consisting of  $v$  EVs and an Road Side Unit RSU  $R$  deployed with an SDN controller. The vehicular task generated by EV  $V_i$  is denoted as  $S_i$  with service type  $S_i^{st}$ ,  $S_i^{st} \in \mathcal{H}$  where  $\mathcal{H}$  represents the set of service types. As shown in Fig. 1, vehicular task execution is divided into three parts, i.e., input, execution and output.

Herein, the input of vehicular task consists of task data and the data package required by the service type corresponding to the task. The task data is transmitted from the under-resourced EV, illustrated as process ① and ④ by Vehicle-to-Vehicle (V2V) communication and Vehicle-to-RSU (V2R) communication, respectively. Since the mobility of vehicles, Orthogonal Frequency-Division Multiple Access (OFDMA) is leveraged to guarantee steady communication. The signal noise ratio of channel  $ij$  is calculated by  $p_i g_{i,j} / \sigma^2$  without interference, and the signal to interference plus noise ratio between  $V_i$  and  $R$  is defined as  $p_i g_{i,r}^2 / \sigma^2 + I_{i,r}$ . Herein,  $p_i$ ,  $g_{i,*}$  (including  $g_{i,j}$  and  $g_{i,r}$ ),  $\sigma^2$  and  $I_{i,r}$  respectively indicate the transmission power, channel gain, additive white Gaussian noise and interference. The inference is generated by other users occupied V2R sub-channels simultaneously,  $I_{i,r} = \sum_{i=1}^v x_{i,r} p_{i,r} (g_{i,r})^2$  and  $x_{i,r}$  indicates the decision variable to indicate the channel occupation. The communication rate among devices (including EVs and RSU) can be obtained by Shannon equation.

The required data package can be obtained via downloading from the Internet (process ③ and ⑤) or sharing by other vehicles caching this data (②). The idle caching resource can be utilized to cache data rather than repeatedly downloading for reducing the redundant cost [12]. Herein, data package obtained rate can be denoted by  $\sum_{j=1}^v y_{j,h} R_{j,h}^{comm} + (1 - \sum_{j=1}^v y_{j,h}) R_j^{down}$ , where binary variable  $y_{j,h}$  indicate the decision of transmitting data package from  $V_h$  to  $V_j$ . In this paper, the output of task execution is viewed as small enough to be negligible [13]. After obtaining input data, task can be executed in device. The computing rate  $R_i^{comp}$  (Mbits per second) of  $S_i$  can be calculated as  $\sum_{j=0}^v x_{i,j} s_i / t_i^{comp}$ . Herein,  $j = 0$  indicate the RSU.

Since the rationality and selfishness of vehicular users, we set battery power and available computational capacity as the main aspects to define the unit price of renting computation resources of EVs and RSU, which are calculated by  $\rho^{comp}(f_j, \varepsilon_j) = (\alpha e^{\beta/\varepsilon_j}) / f_j$  and  $\rho_r^{comp}(f_r, 1) = (\alpha e^{\beta}) / f_r$ , respectively. Herein, parameters  $\alpha$  and  $\beta$  rep-

resents price coefficients to adjust the impact of computational resources and battery power on unit rent at a time slot.

## 2.2. Problem Formulation

In this investigation, we define QoS as the ratio of payment and the sum service rate, i.e.,  $U_i = R_i^{total} / P_i^{total}$ . The optimization objective is set as maximizing the obtained QoS, i.e.,

$$\max_{x_{i,j}, y_{j,h}} U_i, \quad (1)$$

and is subject to task migration delay constraint, i.e.,  $t_i^{mig} \leq \min\{d_i, d_{i,j}^{comm}\}$ , indicating the migration delay cannot exceed the delay limitations of  $S_i$  and communicable distance; power consumption constraint, i.e.,  $\epsilon_i^{cost} < \epsilon_i, V_i \in V$ , indicating the cost of each device cannot exceed the minimum power threshold; and communication capacity constraint, i.e.,  $\sum_{i=1}^V r_{i,r}^{comm} \leq Z_r, V_i \in V$ , representing the occupied communication resources cannot exceed the communication capacity.

## 3. Task Cooperative Migration Policy Integrating 3C Resources

To solve the optimization problem formulated in the previous section, we design a distributed IL-based TOPIC policy. The policy consists of an expert policy and an agent policy. Compared to vehicles, RSU deployed with an SDN controller can observe the global states, which can be viewed as the expert node to execute expert policy with global observation. We propose a Deep Reinforcement Learning (DRL)-based expert policy and the Markov decision process is modeled as a quaternion  $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P} \rangle$ , where  $\mathcal{S}$ ,  $\mathcal{A}$ ,  $\mathcal{R}$  and  $\mathcal{P}$  represent system state, system action, reward function and state transition probability of TOPIC, respectively. The definitions of state, action and reward are specified as follows: The environment state  $\mathcal{S}(t)$  is constructed with 3C states. The action  $\mathcal{A}(t)$  consists of task execution and required data package obtainment, which is defined as:  $\mathcal{A}(t) = \{x(t), y(t)\}$ , where  $x(t)$  and  $y(t)$  are two vectors of binary variables. After executing the  $\mathcal{A}(t)$ , agent obtains  $\mathcal{R}(t) = \sum_i U_i(t)$ , if the constraints are satisfied;  $\mathcal{R}(t) = 0$ , otherwise. To obtain the optimal policy, the accumulate reward of the optimization problem is formulated as follows:  $\mathcal{R}_i = \max_{x_{i,j}, y_{j,h}} \mathbb{E}[\sum_{t=0}^{T-1} \gamma^t \mathcal{R}_i(t)]$ , where  $\gamma$  represents a discount factor indicating the impact of the current feedback gradually decreases over time,  $\gamma \in (0, 1]$ .

When expert node (RSU) obtains a task migration request of  $S_i$ , the SDN controller constructs adjacent matrix  $\mathbb{M}_{V \times V}$  with communication rate between devices, i.e.,  $m_{i,j} = r_{i,j}^{comm}$ , and observed the 3C states  $\mathcal{S}(t)$ . The SDN controller makes task migration decisions with a Double Deep Q-Network (DDQN) algorithm to maximize the obtained QoS by making a satisfying trade-off between service rate and payment with the proposed pricing mechanism. The core of the DRL-based expert policy is to obtain the optimal action with the maximum accumulate reward. With the global observation and the two-hop communicable search range of the requesting device, the caching data hit rate can be improved to reduce the redundant cost.

RSU generates the trajectories of expert policy through offline training, and the trajectories are transmitted to covered vehicles for training Deep Neural Network (DNN)-

**Table 1.** Simulation parameter settings

Parameter	Value
Bandwidth of RSU	50MHz
Transmit power of RSU	0.5W
Gaussian channel noise	$5 \times 10^{-13}$
Available computational capacity of RSU	12Gcycles/s
Unit rent of RSU computational resource	0.7token/Gcycles
Unit rent of RSU communication resource	0.07token/Mbps
Unit rent of RSU download resource	0.2token/Mbps
Download rate of RSU	4Mbps
Available computational capacity of EV	[4, 8, 12, 16, 20]Gcycles/s
Download rate of EV	2Mbps
Unit rent of EV for resource downloading	0.1token/Mbps
Service type of tasks	{1, 2, 3, 4, 5}
Data size of task with service type $K$	$K \times 1Mbit$
Computational resource required by task with service type $K$	$K \times 2Gcycles$
Data size of data package required with service type $K$	$K \times 50Mbits$
Delay limitation of service type $K$	$K \times 30s$

based online agent policy. The observations obtained by requesting vehicles and decisions are viewed as the features and labels, respectively. Vehicles construct a DNN-based classifier to label the observation for obtaining task migration decisions.

#### 4. Performance Evaluation

In this section, the simulation setup is introduced first. Then, the proposed TOPIC policy (includes an expert policy and an agent policy) is evaluated based on the real-world traffic dataset in Hangzhou (China).

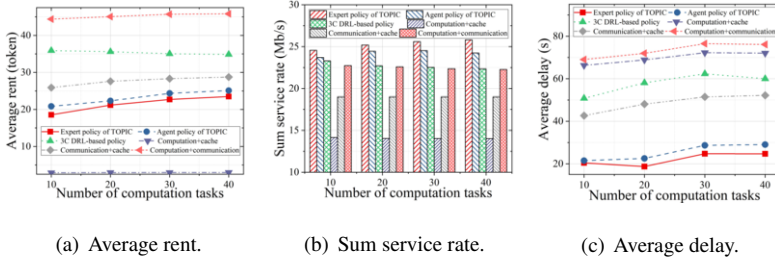
##### 4.1. Simulation Setup

We utilize PyCharm 2019.03 version based on Python 3.7.4 to perform experiments in a 64bit Windows 10 operating system computer with 128 GB RAM and an Intel(R) Xeno(R) CPU E5-2640 v4 with 2.4GHz frequency. TOPIC is realized on TensorFlow 1.14.0. Based on the characteristic of V2R communication, the channel gain of RSU and EVs are calculated by  $127 + 30 \times \log_2 L$  [14], where  $L$ , indicating the distance between devices (includes EVs and RSU), can be obtained by GPS. The central coordinate [30.2547, 120.2741] of the dataset is set as the location of RSU. The channel of V2R is divided into 10 sub-channels. To simplify, the number of service types is reduced to 5, i.e.,  $K = 5$ . Other simulation parameters are stated in Table 1.

To demonstrate the superiorities of TOPIC, we define six indicators, detailed as follows: (1) Obtained QoS: the ratio of the sum service rate to total payment; (2) Average rent: the average rent of migrating a task for execution; (3) Sum service rate: the sum service rate of executing a task; (4) Average power consumption: the average power consumption of under-resourced EVs; (5) Average increased Time-To-Live (TTL): the average increased TTL of the under-resourced EVs; (6) Average delay: the average execution latency of migrated tasks.

To demonstrate the effectiveness of the proposed TOPIC, the selected representative schemes are detailed as follows: (1) 3C DRL-based Policy [15]: It is a task offloading policy integrating 3C resources based on duelling DDQN to allocate 3C resources within one-hop communication range; (2) Computation+cache: computing can only be executed locally without communication resources; (3) Communication+cache: computing tasks can be migrated to RSU or executing locally by making a trade-off between the communication and caching resources; (4) Communication+computation: this policy allocates communication and computational resources based our proposed TOPIC policy with caching state locally.

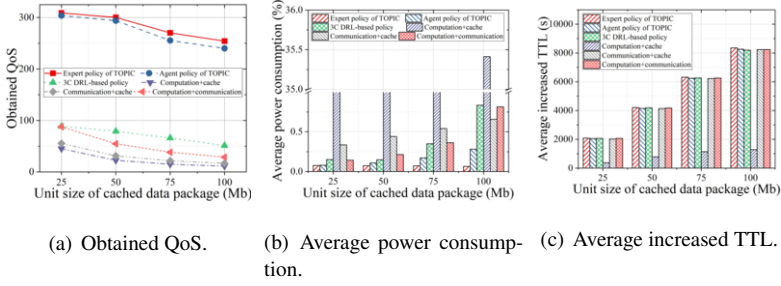
#### 4.2. Experimental Results



**Figure 2.** Performance of different computational workload.

Fig. 2 evaluates the performance with the distinct computational workload of the cluster. According to the definition of QoS, the performance of average rent cost and sum service rate of executing tasks under distinct computation workload are shown in Figs. 2(a) and 2(b), respectively. It can be observed that although the rent based on computation+cache policy is the lowest, computation+cache policy cannot utilize communication resources to reduce the local workload and increase the QoS, resulting in the lowest sum service rate for task execution. When the number of computing tasks is 40, the obtained sum service rate of TOPIC is 80% higher than computation+cache policy. The expert policy of TOPIC obtains the highest service rate with average rent second to computation+cache policy. Following the trajectories generated by the expert policy, the proposed agent policy can be better approximate to the performance of the expert policy. The rent of agent policy is only 7% higher than that of expert policy, and the obtained service rate is only 4% lower than that of expert policy. Figs. 2(a) and 2(b) indicate the proposed expert policy TOPIC makes a satisfying trade-off between the rent and the obtained service rate to maximize the total QoS under a large amount of computation workload. Fig. 2(c) shows the average delay cost of executing one task with distinct computation workloads, respectively. The average delay of the expert policy of TOPIC is much lower than the 3C DRL-based policy and other 2C policies when computation tasks are 40. That is because TOPIC policies within the two-hop search range decrease the competition for available resources increases as the computation workload becomes large. Agent policy of TOPIC imitates and follows the trajectories of expert policy for a satisfying trade-off between the obtained service rate and rent and further decreases the average delay for task execution.

The performance with distinct cached data package sizes is illustrated in Fig. 3. Fig. 3(a) evaluates the obtained QoS of different policies with distinct unit sizes of the data



**Figure 3.** Performance of different cached data package size.

package. Even though the QoS of TOPIC (including expert policy and agent policy) decreases with the increasing unit data package size, its advantages over the 3C DRL-based policy increase. That is because the 3C states are constructed by the expert policy of TOPIC within a larger search range, improving the cache hit rate, and the proposed DNN-based agent policy imitates the trajectories of expert policy to label the states for making decisions accurately. In Fig. 3(b), the increasing rate of the TOPIC is much slower compared with other policies. For computation+cache policy, the power consumption of executing tasks that require large data packages is huge. The power consumption of the expert policy and agent policy is 62.09% and 43.65% lower than that of 3C DRL-based policy, respectively. That is because although the power consumption of 3C DRL-based policy can be reduced by integrating 3C resources, while the power consumption of waiting for completing tasks increases with the increasing data package. Fig. 3(c) evaluates the increased TTL of under-resourced EVs compared with local execution. Policies greatly increase TTL except for the computation+cache policy. Experimental results show that the computation+caching policy has to execute tasks locally without communication resources, resulting in large energy consumption. However, policies integrating communication resources migrate tasks to other devices for execution, greatly reducing energy consumption and extending the TTL of EVs. Herein, TOPIC (including expert policy and agent policy) with the higher cache hit rate, reducing the most redundant cost to save the battery power and increase the life cycle of vehicles.

## 5. Conclusion

This investigation focused on integrating 3C heterogeneous edge resources in Internet of EVs based on the SDN technique in the scenario with limited infrastructure resources. We formulated the optimization objective as maximizing the obtained QoS per token to motivate device cooperations. To make a satisfying trade-off between devices available resources and obtained service rate, we considered the constraints, including mobility, delay limitation, power consumption and available resources. A distributed task cooperation migration method based on IL was proposed to maximize the obtained QoS, namely TOPIC. Performance evaluations based on the real-world traffic dataset in Hangzhou (China) illustrated the effectiveness of our proposed policy from the perspectives of computation workload and unit cached data package size. Experimental results demonstrated that the QoS obtained by the expert policy and agent policy of TOPIC proposed in this work is about 3 times higher than those of other policies.

Although distributed IL is safer than centralized policy, designing a security mechanism to monitor malicious nodes to protect privacy in the ultra-dense dynamic scenario with overloaded communication is our further research topic.

## Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grants 61971084 and 62001073, and in part by the Dalian Young Science and Technology Star under Grant 2020RQ002.

## References

- [1] H. Chen et al., "Green Internet of vehicles: Architecture, enabling technologies, and applications," *IEEE Access*, vol. 7, pp. 179 185–179 198, 2019.
- [2] X. Wang et al., "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.
- [3] Z. Ning et al., "Blockchain-enabled intelligent transportation systems: A distributed crowdsensing framework," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021.
- [4] Z. Ning et al., "5G-enabled UAV-to-community offloading: Joint trajectory design and task scheduling," *IEEE Journal on Selected Areas in Communications*, pp. 1–1, 2021.
- [5] Z. Zhou et al., "Online orchestration of cross-edge service function chaining for cost-efficient edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 8, pp. 1866–1880, 2019.
- [6] K. Li et al., "Exploiting computation replication for mobile edge computing: A fundamental computation-communication trade-off study," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4563–4578, 2020.
- [7] C. Ding et al., "Joint optimization of transmission and computation resources for satellite and high altitude platform assisted edge computing," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2021.
- [8] S. Luo et al., "Fog-enabled joint computation, communication and caching resource sharing for energy-efficient IoT data stream processing," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3715–3730, 2021.
- [9] M. Tang et al., "Enabling edge cooperation in tactile Internet via 3C resource sharing," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2444–2454, 2018.
- [10] L. Xu et al., "Socially driven joint optimization of communication, caching, and computing resources in vehicular networks," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2021.
- [11] Z. Ning et al., "Intelligent edge computing in Internet of vehicles: A joint computation offloading and caching solution," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020.
- [12] Z. Ning et al., "Joint computing and caching in 5G-envisioned Internet of vehicles: A deep reinforcement learning-based traffic control system," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.
- [13] Z. Ning et al., "Intelligent resource allocation in mobile blockchain for privacy and security transactions: a deep reinforcement learning based approach," *Science China Information Sciences*, vol. 64, no. 6, pp. 1–16, 2021.
- [14] Y. Sun et al., "EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2637–2646, 2017.
- [15] Y. He et al., "Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 44–55, 2018.