Personalized Vehicular Edge Computing in 6G

Yilong Hui, Nan Cheng, Yuanhao Huang, Rui Chen, Xiao Xiao, Changle Li, and Guoqiang Mao

Abstract

Sixth-generation (6G) space-air-ground integrated vehicular networks (SAGIVNs) are expected to provide customized edge computing services for moving vehicles (MVs). In this article, by integrating 6G SAGIVNs and edge computing, we propose a secure and personalized vehicular edge computing framework in 6G to satisfy the diversified requirements of MVs. In the framework, we first develop a collaborative edge computing mechanism, where each 6G infrastructure (e.g., satellite, drone, and base station) cooperates with parked vehicles to provide services for MVs in order to improve the efficiency of the edge computing services. Then, considering the security issues, a smart-contract-based secure collaboration mechanism is designed to establish a reliable transaction environment for edge computing in 6G. Next, based on the personalized service requirements of MVs and the available resources in 6G SAGIVNs, the resource allocation strategy of each infrastructure and the competitiveness of different infrastructures are discussed to provide MVs with the optimal personalized service. After that, compared with the conventional schemes, a case is studied to evaluate the effectiveness of the proposed framework. Finally, we outline open research topics to identify future research opportunities and directions.

Introduction

Sixth-generation (6G) wireless networks are expected to realize global coverage by developing space-air-ground integrated networks (SAGINs) [1]. SAGINs, which have received worldwide attention, mainly refer to the combination of space-based networks (i.e., satellites), air-based networks (i.e., aircraft and drones), and ground-based networks (i.e., cellular base stations (CBSs) and roadside units (RSUs)) [2]. Featured by network heterogeneity, the SAGINs can be designed to form 6G space-air-ground vehicular networks (SAGIVNs) with the target of breaking through the bottleneck of traditional vehicular networks (VNETs). As shown in Fig. 1, 6G SAGIVNs can be integrated with edge computing devices (ECDs) to provide various vehicular applications for moving vehicles (MVs) [3].

Unlike conventional VNETs, vehicular edge computing in 6G SAGIVNs has the following advantages.

Increase Computing Resources: In 6G SAGIVNs, the available computing resources in the networks increase with the number of ECDs deployed at the infrastructures. Therefore, the amount of resources in 6G SAGIVNs is much larger than that in conventional VNETs.

Support Ubiquitous Computing: 6G SAGIVNs have global communication coverage. Consequently, the heterogeneity of 6G SAGIVNs can be exploited to enhance the communication coverage and provide ubiquitous computing services for MVs.

Provide Flexible Services: ECDs deployed at different types of infrastructures have different performances in terms of transmission and computing. Therefore, each MV can select the optimal network to offload the computing service, where the flexibility of the computing services can be enhanced.

Although MVs can benefit from integrating 6G SAGIVNs and edge computing, vehicular computing services are still restricted by the following challenges.

Limited Computing Resources: 6G SAGIVNs can improve the efficiency of data transmission and task computing. However, the limited ability (e.g., communication coverage and computing resources) of some infrastructures (e.g., drones, CBSs and RSUs) makes it difficult to meet the needs of computing services. Namely, ECDs do not have sufficient transmission and computation resources to support the rapid growth of computationally demanding services [4].

Insecure Network Environment: In 6G SAGIVNs, the number of nodes participating in computing services will increase significantly. However, nodes such as ECDs and MVs are not completely trustworthy [5]. Consequently, the efficiency of the task computing services may be significantly reduced. Therefore, in 6G SAGIVNs, how to design a mechanism to securely provide services becomes a challenge.

Insufficient Personalized Services: In 6G SAGIVNs, MVs usually have various quality of experience (QoE) requirements for services. However, existing studies rarely consider the individual requirements that directly affect the QoE of MVs. Furthermore, in conventional networks, each ECD usually adopts a single mode to provide services for MVs, which ignores the personalized demands of MVs for different computing tasks. In contrast, the ECDs in 6G SAGIVNs show different capabilities in both transmission and computation [6]. Faced with the limited resources and personalized requirements, it is a challenge for each ECD to determine the customized service strategy (CSS) to efficiently complete the task. According to the CSSs provided by the 6G SAGIVNs, the

Digital Object Identifier: 10.1109/MNET.001.2100151

The authors are with Xidian University; Changle Li and Guoqiang Mao are corresponding authors.

0890-8044/21/$25.00 © 2021 IEEE

MV needs to select an ECD to obtain the highest QoE. Therefore, how to consider the competition between the ECDs and select the optimal ECD to execute the requested task becomes a challenge.

To this end, in this article, we propose a secure and personalized vehicular edge computing framework in 6G SAGIVNs. In the proposed framework, each 6G infrastructure can exploit the available computing resources of the parked vehicles (PVs) to collaboratively provide personalized computing services for MVs. Then, in view of the security issues, smart contracts are designed to establish a reliable transaction environment to protect collaborative computing services. After that, by jointly considering the personalized requirements of MVs and the available resources in the 6G SAGIVNs, we respectively discuss the resource allocation strategy of each ECD and the competitiveness of different ECDs to provide MVs with the optimal personalized service. Finally, a case is studied to evaluate the effectiveness of the proposed framework. The main contributions of this article are outlined as follows:

- We design a collaborative edge computing mechanism (CECM) to help each 6G infrastructure cooperate with PVs parked in its coverage and provide personalized services for MVs. With this mechanism, the available resources in the networks can be increased and the flexibility of computing services can be enhanced.
- We propose a smart-contract-based secure collaboration mechanism (SSCM) to deal with security issues in the networks. By using the SSCM, a reliable transaction environment can be established for supporting personalized vehicular edge computing in 6G SAGIVNs.
- We analyze the personalized service requirements of MVs, the resource allocation strategy of each ECD and the competitiveness among different ECDs in 6G SAGIVNs, respectively. Based on a case study, the CSS of each ECD is determined to optimally allocate resources for providing MVs with personalized computing services.

Personalized Vehicular Edge Computing Framework in 6G SAGIVNs

In this section, we first design the CECM and the SSCM in 6G SAGIVNs. After that, we discuss the CSS of each ECD and the competition among different ECDs to decide the optimal CSS for MVs.

Collaborative Edge Computing Mechanism

In this subsection, we design a cooperation mechanism that integrates ECDs and a large number of PVs. As studied in [7], the parking time of vehicles is about 95 percent. Therefore, PVs show great potential to support computationally intensive applications [8]. Thus, we model 6G SAGIVNs by the following components.

Infrastructures: The network infrastructures consist of satellites, drones, CBSs, RSUs, and so on. In the coverage of each infrastructure, there are a large number of PVs and MVs. Therefore, the ECD deployed at an infrastructure can share data with the PVs and MVs to provide computing services.

ECDs: ECDs aim to provide computing services for MVs at the edge of the 6G SAGIVNs. Generally, ECDs have different computing capabilities (e.g., idle computing resources and cost per unit resource). Consequently, MVs can send computing requests to different ECDs to maximize their QoE.

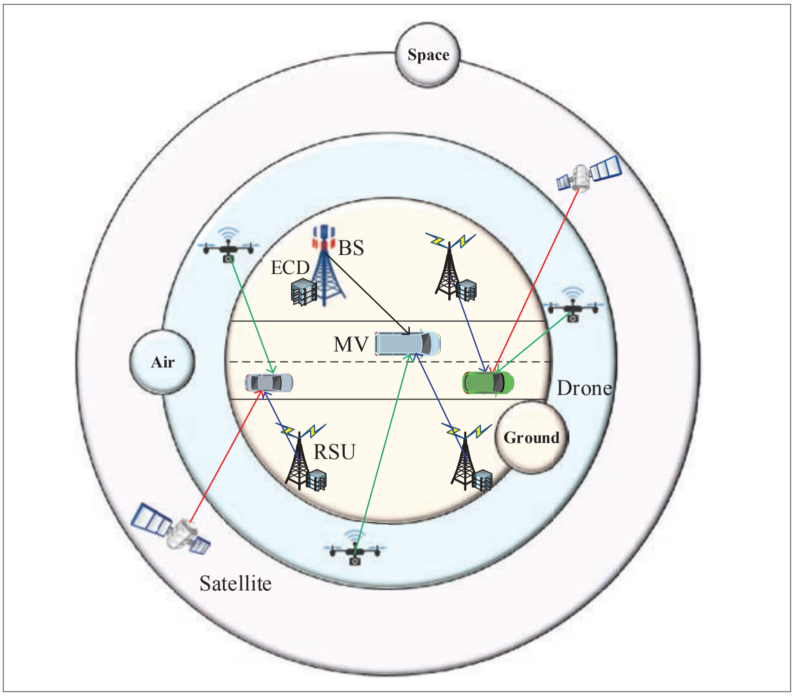

FIGURE 1. The system model of the 6G SAGIVNs.

MVs: For an MV, it is usually covered by more than one ECD in 6G SAGIVNs. Therefore, with an onboard unit (OBU), the MV can make connections with these ECDs to request services based on its requirements [9].

PVs: The PVs in 6G SAGIVNs can be exploited to execute tasks. The reward per unit resource declared by each PV is related to its idle computing resources. In general, the PV that has more idle resources typically declares a higher reward to join the computing process.

With this network architecture, as shown in Fig. 2, the designed CECM in 6G SAGIVNs includes the following steps:

- Each ECD collects the information of the available resources and reward per unit resource of the PVs within its communication coverage.
- The MV determines its QoE requirements.
- The MV delivers its task computing request to all connected ECDs.
- Each ECD determines the CSS by jointly considering the task requirements and the available computing resources of both the ECD and the PVs managed by the ECD.
- If the CSS can satisfy the requirements of the MV, the ECD delivers the determined CSS to the MV.
- The MV collects all the CSSs provided by the 6G SAGIVNs and selects the optimal one.
- Based on the determined CSS, the selected ECD executes the requested computing task with a number of PVs collaboratively.
- After the allocated subtasks are completed, the results of these subtasks will be collected by the ECD to output the final result.
- The ECD delivers the task result to the MV.

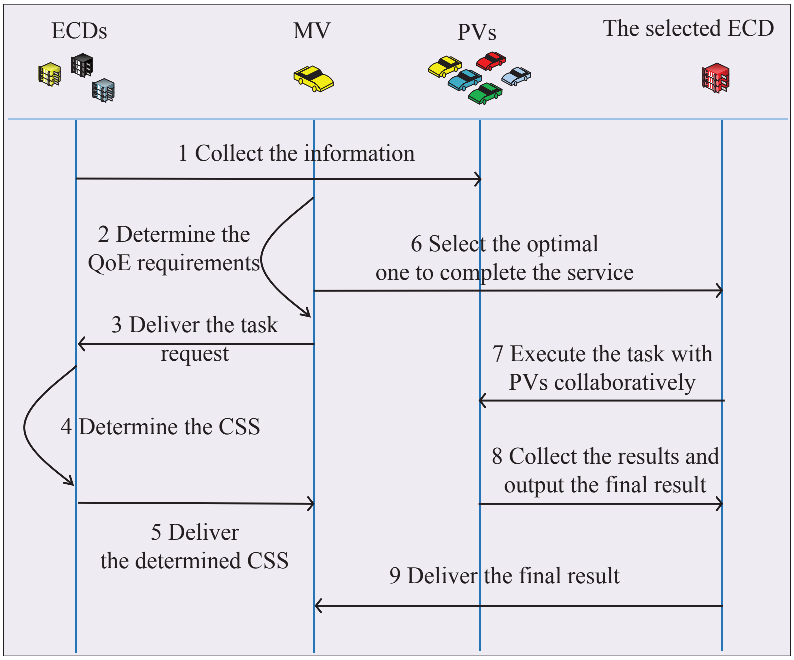

FIGURE 2. The collaboration mechanism.

With the designed CECM, PVs and different ECDs can be integrated to form distributed computing resource pools, where the available computing resources can be increased and the efficiency of computing services in the networks can be improved. On the other hand, each ECD can collaborate with a different number of PVs within its communication coverage. In this way, the diversified modes of computing services can be provided by each ECD for MVs.

Smart-Contract-Based Secure Collaboration Mechanism

In this subsection, we design the SSCM to address the security issues.

Step 1–Entity Registration: In 6G SAGIVNs, each entity needs to register to the certification authority (CA) [10]. The CA is in charge of the issuance and withdrawal of digital certificates (DCs) of the entities. If the submitted material of an entity is valid, it then becomes a legitimate node and can obtain the public key, the private key, the DC, and the wallet address (WA).

Step 2–Contract Deployment: The traditional transaction mechanism is susceptible to various malicious behaviors, reducing the reliability and stability of the system. To this end, two types of smart contracts are deployed in the proposed framework.

Contract Signed by the ECD and the PVs: As shown in Fig. 3a, a vehicle that parks in the communication coverage of an ECD can sign a contract with the ECD to earn rewards. The contract includes the following parts:

- Idle computing resources of the PV
- Reward per unit resource of the PV
- DCs and WAs of the PV and the ECD
- Signatures signed by the ECD and the PV.

After deploying the contract in the blockchain networks, the vehicle then becomes a PV. Namely, it is managed by the ECD to collaboratively provide resources for completing computing tasks. This contract is triggered when the new block related to the transactions between the ECD and the PVs is added to the blockchain.

Optimal ECD Selected to Sign the Contract: According to the personalized requirements, the MV which intends to offload its computing task can generate a request. The ECDs receiving the message verify the MV's DC and signature. If the message is valid, each ECD then makes the CSS and delivers the response message to the MV. The MV collects the response messages delivered by the ECDs and verifies their DCs. For the legal ECDs, the MV will select the ECD which provides the optimal CSS to execute the task.

Contract Signed by the ECD and the MV: After selecting the optimal ECD to execute the task, the MV then signs a smart contract with the ECD. Furthermore, the input data of the task will be delivered from the MV to the ECD. This contract takes effect when the selected ECD uploads the task result to the blockchain networks and the new block related to this transaction is added to the blockchain.

Step 3–Contract Execution:

Contract Executed by the PVs: After the contract is deployed, the ECD starts to execute the task by following the decided CSS. Specifically, based on the CSS, the task will be executed by the ECD individually if all the resources are provided by the ECD. In contrast, if the CSS indicates that the task will be completed by the ECD and a number of PVs collaboratively, the ECD then allocates the subtasks based on the CSS. As shown in Fig. 3b, the message which indicates the number of subtasks allocated to each PV will be written to the contract signed by the ECD and the PVs. Accordingly, the ECD transmits the input data of the allocated subtasks to the PVs. Similar to [11], the Byzantine fault tolerance (BFT) consensus mechanism is used to complete the verification of the transactions, where the legal ECDs serve as the nodes to complete the consensus process. After finishing the consensus and updating the contract, the PVs then start to execute the allocated subtasks. If a PV completes the subtasks, it signs the result message and encrypts it using the public key of the ECD. Furthermore, the encrypted result will be written to the blockchain networks.

Rewards Paid to the PVs: If the submitted results are verified, the smart contract will be executed automatically to transfer the rewards from the ECD's WA to the PV's WA.

Contract Executed by the ECD: If the number of subtasks allocated to the ECD is larger than 0, the ECD needs to cooperate with the selected PVs to complete the task. In contrast, if the number of subtasks allocated to the ECD is equal to 0, the ECD only needs to collect the results uploaded by the PVs to summarize the final result. After this, as shown in Fig. 3b, the ECD writes the encrypted result message to the blockchain networks.

Rewards Paid to the ECD: After the consensus process is finished, the MV then downloads the result from the blockchain networks. Along with this, the smart contract will be triggered automatically to transfer the rewards from the MV's WA to the ECD's WA, ending the execution of the task.

Based on the designed SSCM, the smart contracts signed by different nodes and the transac-

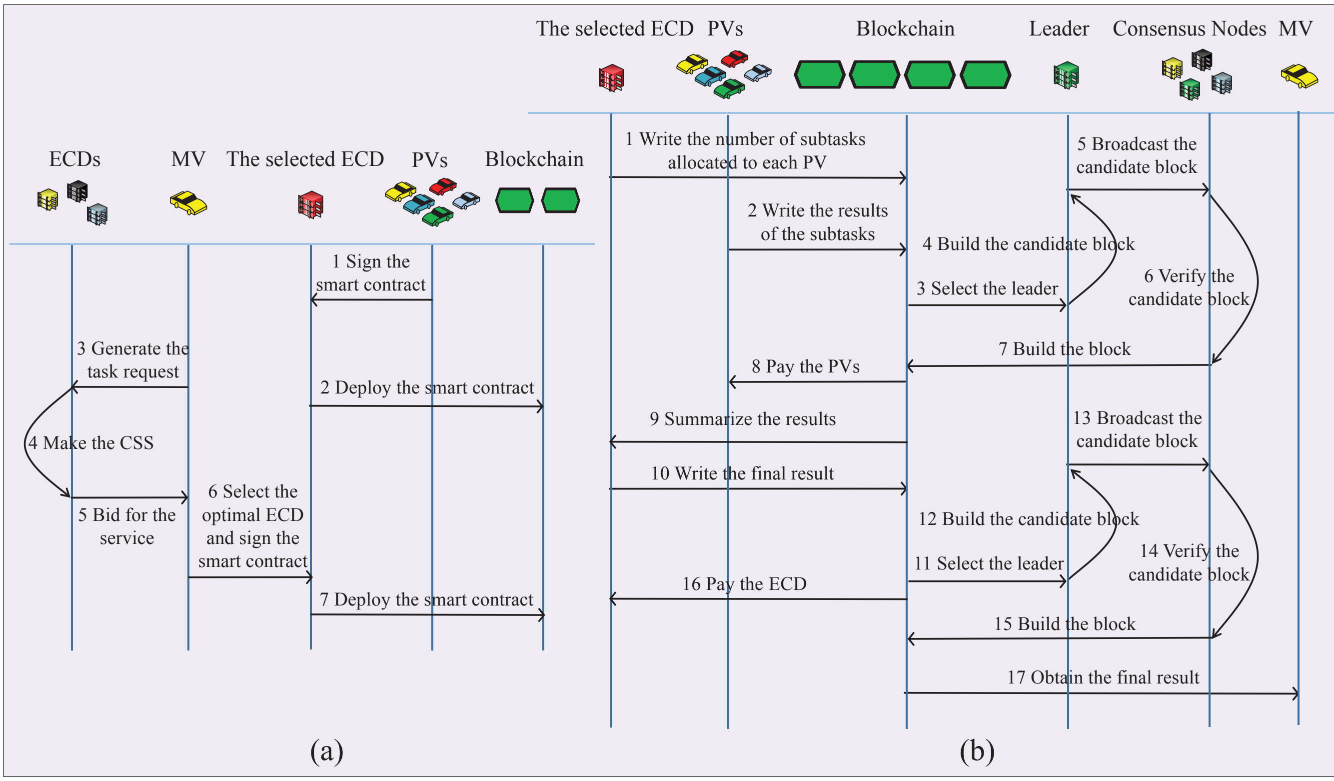

FIGURE 3. a) The deployment of the smart contract; b) the execution of the smart contract.

tion records generated in the 6G SAGIVNs are recorded on the blockchain. In addition, the execution of each smart contract requires blockchain nodes to complete the consensus process. Therefore, with the adoption of the SSCM, a secure computing environment can be established to guarantee the reliability, nonrepudiation, and traceability of transactions.

Customized Edge Computing Service

The ECDs deployed at different network infrastructures together with PVs in their communication coverage can form different service networks. As a result, different CSSs will be given to satisfy the MV's QoE requirements. In this subsection, we first introduce the requirements of the MV and analyze the CSS of each ECD. After that, the competition among different ECDs is considered to help the MV select the optimal CSS.

Personalized Requirements of MVs: In 6G SAGIVNs, compared with traditional service indicators (e.g., transmission rate, throughput, connection density, and so on), MVs are more concerned with those indicators that directly affect personal experience (i.e., delay and cost). Therefore, a task can be described by a message with five elements, that is, size of input data, requested computing resources, division unit, deadline of the task, and the estimated rewards for completing the task. Similar to [12], the tasks considered in this article are bit-wise independent so that each task can be partitioned into several subtasks. Therefore, the number of subtasks can be calculated by dividing the requested resources by the division unit of the task. The rest of the elements, that is, the deadline of the task and the estimated rewards for completing the task, can be set by the MV to reflect its preference for the task. Namely, the QoE of the MV can be customized by setting the deadline of the task and the estimated rewards for completing the task.

CSSs of ECDs: Each ECD has three cases to decide the CSS:

- If the MV has higher requirements for task execution time, each ECD needs to customize the service strategy for the MV to minimize the service time while meeting the requirements of rewards. As the computation module and transmission module of ECD and PVs are separated from each other, the task computing is generally a parallel process, where the service time is mainly affected by the longest calculation time. Specifically, if the task is completed by the ECD individually, the service time is the time that the ECD spent on completing the task. On the other hand, if the task is completed by the ECD and the PVs together, the service time is the longest execution time spent by one of the ECD and PVs to execute the allocated subtasks.
- If the MV has higher requirements for the rewards, the ECD needs to customize the service strategy for the MV to minimize the rewards while meeting the requirements of task execution time. In this case, the QoE of the MV is mainly related to the reservation rewards and the rewards paid for the service. With the reservation rewards, the MV will obtain a higher QoE if the rewards the MV pays for this service are lower. According to the CSS, if the task is completed by the ECD and the PVs collaboratively, the cost of completing the task includes the transmission cost and the computing cost of the ECD as well as the rewards paid to the selected PVs.

```
Phase 1: Initialization Phase
1  flag=0; K={0,1,...,k,...,K}, L={l_0,l_1,...,l_k,...,L_K}, X={x_0,x_1,...,x_k,...,X_K}
2  While d_g>0
3    [value, location]←min{L}
4    Allocate X(location) the most subtasks under the condition that
   the time constraint is satisfied
5    L =L\location
6    If d_g>0 & L=Ø
7      flag=1; Cancle the task; Break
8    End If
9  End While
10 If flag=0
11   For k=0 to k=K
12     Calculate the time to complete the allocated subtasks t_k
13     T(k)=t_k
14   End For
15   For k=1 to k=K
16     T_temp(k)←T(k)
17   End For
18   For k=1 to k=K
19     [value, location]←max{T_temp}; Δt=Δt+X(location)/d_g(s_g/r);
20     T(location)=T(location)+Δt; T_temp(location)=0
21   End For
Phase 2: Iteration Phase
22   While max{T}>T_g
23     [value, location]←max{T}; X(location)←X(location)-move
24     [value, location]←min{L}; X(location)←X(location)+move
25     Call lines 11-21
26     [value, location]←max{L}
27     If T(location)>T_g
28       Cancle the task; Break
29     End If
30   End While
31 End If
Output: The optimal allocation vector CSS X
```

ALGORITHM 1. The proposed CCRA algorithm.

- The MV jointly considers two factors, that is, task execution time and rewards paid for the service. In this case, the weight to balance the two factors can be set between 0 and 1 to reflect the requirements of the MV.

In view of the personalized requirements of MVs, different algorithms can be designed to optimize the allocation of computing resources. Specifically, for the tasks with time or rewards requirements, greedy-based algorithms can be used to allocate resources to determine the CSSs. If the MV intends to optimize the time and the rewards simultaneously, resource allocation can be achieved by designing schemes based on heuristic or deep learning algorithms [13].

In this way, each ECD can use the optimization algorithm to decide its CSS based on the task request of the MVs and its own available resources. With the determined CSS, the computing resources provided by the ECD and the PVs can be decided. Accordingly, the rewards and time of each ECD for completing the task can be obtained.

Competition among ECDs: Given the personalized requirements of the MV, the ECDs in 6G SAGIVNs will provide different CSSs based on the determined time and rewards for executing the task. Therefore, we consider the competition among different ECDs to help the MV select the optimal one. Specifically, based on different QoE requirements of MVs, game theory can be used to establish the competition among ECDs deployed at different network infrastructures [14]. For example, if the MV intends to minimize the rewards paid for the computing service, an auction-based model can be established to describe the competition among different ECDs. By doing this, the MV can select the ECD which bids with the lowest reward to complete the computing service on the premise that the task execution time is satisfied.

Case Study

In this section, a case is studied to evaluate the performance of the proposed framework. We first introduce the simulation scenario, followed by the simulation results and discussions.

Simulation Scenario

In the simulation, there are five ECDs deployed at different 6G network infrastructures, where each ECD can cooperate with PVs in its coverage to provide computing services. The idle resources of the five ECDs are set to be 30, 40, 50, 70, and 100. The number of parking lots in the coverage of the five ECDs is selected from [1, 3], (3, 6], (6, 9], (9, 15] and (15, 30], respectively. The input data size and the execution time of the task requested by the MV are 5Mbits and 0.5s. To evaluate the performance of the designed framework, we change the resources requested by the task from 100 to 300. In addition, we consider that an MV is more sensitive to the rewards paid for the services. In other words, the goal of the MV is to minimize the rewards under the condition that the requirement of task execution time is satisfied.

In this scenario, we use the designed SSCM to provide a secure transaction environment for providing customized services, where each ECD needs to select a set of optimal PVs to collaboratively execute the computing task. To this end, we design a collaborative computing resource allocation (CCRA) algorithm to help each ECD decide its CSS. Let $\boldsymbol{K} = \{0, 1, ..., k, ..., K\}$ denote the set of an ECD and the PVs, where element 0 indicates the ECD and the other elements are the PVs. For task $g(g \in \boldsymbol{G} = \{1, ..., g, ..., G\})$, we denote s_g, T_g, and d_g as the input data size, the requested deadline, and the number of subtasks of task g. In addition, we define two sets $\boldsymbol{L}$ and $\boldsymbol{X}$, where $l_k \in \boldsymbol{L}$ and $x_k \in \boldsymbol{X}$ respectively represent the reward per unit resource declared by node k and the number of subtasks allocated to node k.

As shown in Algorithm 1, the designed CCRA algorithm consists of the initialization phase and the iteration phase. In the initialization phase, the subtasks are allocated to the ECD and the PVs based on a greedy strategy. If the allocation in this phase satisfies the time requirement, the algorithm ends. Otherwise, the algorithm moves to the second phase. In this phase, the subtasks allocated to the PVs that do not satisfy the time

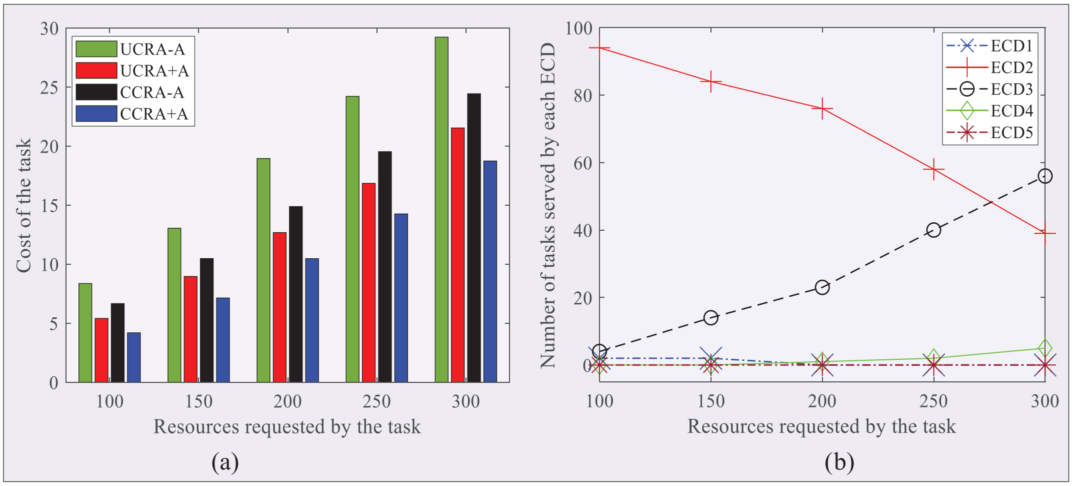

FIGURE 4. a) The cost of the task by changing the resources requested by the task; b) the number of tasks served by each ECD by changing the resources requested by the task.

requirement will be iterated to the PVs that satisfy the time requirement. If the task can be completed before the deadline, the iteration ends and the algorithm outputs the CSS. Otherwise, the algorithm will continue to iterate until the time requirement cannot be satisfied by the node that declares the highest reward per unit resource. If this occurs, the ECD will give up this service.

For the competition among the ECDs deployed at 6G SAGIVNs, a model based on the second price sealed auction is developed. In the game, based on the determined CSS, each ECD sends its requested rewards to the MV, where the ECD that bids with the lowest reward is selected to execute the computing task. Based on the rules of the game, the ECD that wins the auction will obtain the second-lowest rewards. As such, each ECD needs to decide its requested rewards to win the game and obtain rewards. As studied in [15], the optimal bidding strategy of each ECD to win the auction is to bid honestly based on its own cost.

With this scenario, the schemes used in the simulation are summarized as follows.

UCRA-A: The subtasks are uniformly allocated to the ECD and PVs. In addition, the task will be executed by the ECD that is randomly selected by the MV.

UCRA+A: The subtasks are uniformly allocated to the ECD and PVs. Moreover, the task will be executed by the ECD that bids with the lowest reward.

CCRA-A: The subtasks are allocated to the ECD and PVs based on the proposed CCRA algorithm. In addition, the task will be executed by the ECD that is randomly selected by the MV.

CCRA+A: The subtasks are allocated to the ECD and PVs based on the proposed CCRA algorithm. Furthermore, the task will be executed by the ECD that bids with the lowest reward.

Simulation Results

Figures 4a and 4b show the cost spent on completing the task and the number of tasks served by each ECD by changing the values of the resources requested by the task. From Fig. 4a, we can see that the cost of the task in all the schemes increases with the increase of the resources requested by the task. In addition, it can be seen that the proposed CCRA+A can lead to the lowest cost. The reasons are as follows. First, the ECD can decide the optimal CSS based on the algorithm to decrease the task execution cost. Second, the game model can help the MV select the ECD that bids with the lowest reward to execute the task. In Fig. 4b, when the resources requested by the task is 100, ECD 1 is selected two times to execute the task because this ECD has the least computing resources. For ECDs 4 and 5, they are relatively expensive in terms of computing resources. As a result, they are rarely selected to execute the task. In comparison, we can see that the tasks are mainly completed by ECDs 2 and 3. This is because the available resources of ECDs 2 and 3 are more than ECD 1. Furthermore, the rewards for computing the tasks declared by ECDs 2 and 3 are lower than ECDs 4 and 5.

Future Research Directions

Dynamic Cooperation Between 6G Infrastructures

In 6G SAGIVNs, an MV is usually covered by multiple network infrastructures with heterogeneous resources. For the personalized service requirements of an MV, a single infrastructure and the PVs in its coverage may not be able to provide high QoE for the MV. In contrast, the cooperation between multiple network infrastructures can enhance the flexibility of services and significantly improve the QoE of the MV. In addition, high-speed MVs usually pass through the network coverage of multiple infrastructures during driving. Therefore, the dynamic collaboration of different 6G infrastructures is an efficient solution to provide continuously high QoE for MVs.

Digital Twin for Personalized Computing Services

The frequent transmission of large amounts of data is required to provide personalized computing services for MVs in 6G SAGIVNs. Furthermore, the massive personalized requirements of MVs and the limited resources of heterogeneous infrastructures need to be reasonably planned and matched. Digital twin (DT) is regarded as a promising technology to address these challenges. By deploying DTs of the MVs and the infrastructures at the edge or cloud, the DTs in virtual networks can replace the nodes in physical networks to make decisions, thereby reducing the interaction delay while integrating the resources of the infrastructures to provide personalized computing services for different MVs.

Edge Intelligence for Customized Resource Allocation

Faced with a large number of MVs and personalized indicators, it is difficult for traditional methods to make optimal decisions. Furthermore, cloud-based artificial intelligence (AI) generally leads to a long latency and is difficult to satisfy the high QoE requirements of MVs. To this end, as a new AI paradigm, edge intelligence is regarded as a key technology in 6G SAGIVNs. By transferring intelligence from the cloud to the edge or on-board devices and coupling computing and intelligence at the edge of the networks, edge intelligence can significantly reduce the decision-making delay and achieve customized resource allocation to ensure high QoE for MVs.

Conclusion

In this article, we developed a novel personalized edge computing framework in 6G SAGIVNs. In the framework, we established the CECM to help each infrastructure in 6G SAGIVNs cooperate with PVs to collaboratively provide services for MVs. In addition, a secure mechanism, that is, SSCM, was designed to provide a reliable transaction environment for edge computing in 6G to address the security issues. With the designed CECM and SSCM, the resource allocation of each ECD and the competition among different ECDs were discussed to provide MVs with the optimal personalized service by jointly considering the personalized requirements of MVs and the available resources in 6G SAGIVNs. After that, we studied a case to evaluate the effectiveness of the proposed framework. Finally, the open research topics were discussed to identify future research opportunities and directions.

Acknowledgment

This work was supported in part by the National Key R&D Program of China under Grant 2020YFB1807700; in part by the NSFC under Grant 61901341 and Grant 62071356; and in part by the China Postdoctoral Science Foundation under Grant 2021TQ0260.

References

[1] S. Zhang *et al.*, "Envisioning Device-to-Device Communications in 6G," *IEEE Network*, vol. 34, no. 3, May 2020, pp. 86–91.
[2] N. Cheng *et al.*, "Space/Aerial-Assisted Computing Offloading for IoT Applications: A Learning-Based Approach," *IEEE JSAC*, vol. 37, no. 5, May 2019, pp. 1117–29.
[3] N. Cheng *et al.*, "A Comprehensive Simulation Platform for Space-Air-Ground Integrated Network," *IEEE Wireless Commun.*, vol. 27, no. 1, Feb. 2020, pp. 178–85.
[4] Y. Wu *et al.*, "Secrecy-Driven Resource Management for Vehicular Computation Offloading Networks," *IEEE Network*, vol. 32, no. 3, June 2018, pp. 84–91.
[5] Z. Su *et al.*, "A Secure Charging Scheme for Electric Vehicles with Smart Communities in Energy Blockchain," *IEEE Internet of Things J.*, vol. 6, no. 3, June 2019, pp. 4601–13.
[6] Y. Hui, Z. Su, and T. H. Luan, "Collaborative Content Delivery in Software-Defined Heterogeneous Vehicular Networks," *IEEE/ACM Trans. Networking*, vol. 28, no. 2, Apr. 2020, pp. 575–87.
[7] W. M. Griggs *et al.*, "Localizing Missing Entities Using Parked Vehicles: An RFID-Based System," *IEEE Internet of Things J.*, vol. 5, no. 5, Oct. 2018, pp. 4018–30.
[8] C. Li *et al.*, "Parked Vehicular Computing for Energy-Efficient Internet of Vehicles: A Contract Theoretic Approach," *IEEE Internet of Things J.*, vol. 6, no. 4, Aug. 2019, pp. 6079–88.
[9] Y. Hui *et al.*, "Reservation Service: Trusted Relay Selection for Edge Computing Services in Vehicular Networks," *IEEE JSAC*, vol. 38, no. 12, Dec. 2020, pp. 2734–46.
[10] Y. Wang *et al.*, "Challenges and Solutions in Autonomous Driving: A Blockchain Approach," *IEEE Network*, vol. 34, no. 4, Aug. 2020, pp. 218–26.
[11] H. Liang *et al.*, "MBID: Microblockchain-Based Geographical Dynamic Intrusion Detection for V2X," *IEEE Commun. Mag.*, vol. 57, no. 10, Oct. 2019, pp. 77–83.
[12] Y. Zhang *et al.*, "HetMEC: Heterogeneous Multi-Layer Mobile Edge Computing in the 6G Era," *IEEE Trans. Vehicular Technology*, vol. 69, no. 4, Apr. 2020, pp. 4388–4400.
[13] L. Hu *et al.*, "Ready Player One: UAV-Clustering-Based Multi-Task Offloading for Vehicular VR/AR Gaming," *IEEE Network*, vol. 33, no. 3, June 2019, pp. 42–48.
[14] Y. Yu *et al.*, "Multi-Leader follower Game for MEC-Assisted Fusion-Based Vehicle On-Road Analysis," *IEEE Trans. Vehicular Technology*, vol. 68, no. 11, Nov. 2019, pp. 11 200–12.
[15] Y. Hui *et al.*, "A Game Theoretic Scheme for Collaborative Vehicular Task Offloading in 5G HetNets," *IEEE Trans. Vehicular Technology*, vol. 69, no. 12, Dec. 2020, pp. 16 044–56.

Biography

Yilong Hui [M'19] (ylhui@xidian.edu.cn) received the Ph.D. degree from Shanghai University, Shanghai, China, in 2018. He is currently a lecturer with the State Key Laboratory of ISN, Xidian University, China. He has published over 40 scientific articles in leading journals and international conferences. His research interests include vehicular networks and autonomous driving.

Nan Cheng [M'16] (dr.nan.cheng@ieee.org) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo in 2016. He is currently a professor with the State Key Lab of ISN, Xidian University, Shaanxi, China. He worked as a post-doctoral fellow with the Department of Electrical and Computer Engineering, University of Toronto, from 2017 to 2018. His current research focuses on space-air-ground integrated system and application of AI for vehicular networks.

Yuanhao Huang (huangyh@xidian.edu.stu.cn) received the B.Eng. degree in communication and information systems from Xidian University, China, in 2020. He is currently pursuing the Master degree with the School of Telecommunications Engineering, Xidian University, Xi'an, China. His current research interests include blockchain and vehicular networks.

Rui Chen [M'11] (rchen@xidian.edu.cn) received the Ph.D. degree in communications and information systems from Xidian University, Xi'an, China, in 2011. He is currently an associate professor with the School of Telecommunications Engineering, Xidian University. He has published about 60 papers in international journals and conferences. His research interests include broadband wireless communication systems and intelligent transportation systems.

Xiao Xiao [M'21] (xiaoxiao@xidian.edu.cn) received the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2009. He is currently a lecturer with the School of Telecommunications Engineering, Xidian University, China. He was a research fellow with the Department of Electrical Engineering and Computer Science, University of Tennessee, from 2019 to 2020. His research interests include computer vision and deep learning.

Changle Li [SM'16] (clli@mail.xidian.edu.cn) received the Ph.D. degree in communication and information systems from Xidian University, China, in 2005. He was a post-doctoral researcher in Canada and the National Institute of Information and Communications Technology, Japan. He is currently a professor with the State Key Laboratory of ISN, Xidian University. His research interests include intelligent transportation systems and wireless sensor networks.

Guoqiang Mao [F'17] (g.mao@ieee.org) has published over 200 papers in international conferences and journals, which have been cited more than 10000 times. He is/has been an editor of *IEEE Transactions on Intelligent Transportation Systems*, *IEEE Transactions on Wireless Communications*, and *IEEE Transactions on Vehicular Technology*. He received the Top Editor award for outstanding contributions to *IEEE Transactions on Vehicular Technology* in 2011, 2014 and 2015. He was a co-chair of the IEEE Intelligent Transport Systems Society Technical Committee on Communication Networks. His research interests include intelligent transport systems and wireless localization techniques.