Mobile Edge Computing-Enabled Internet of Vehicles: Toward Energy-Efficient Scheduling

Zhaolong Ning, Jun Huang, Xiaojie Wang, Joel J. P. C. Rodrigues, and Lei Guo

Abstract

Although modern transportation systems facilitate the daily life of citizens, the ever-increasing energy consumption and air pollution challenge the establishment of green cities. Current studies on green IoV generally concentrate on energy management of either battery-enabled RSUs or electric vehicles. However, computing tasks and load balancing among RSUs have not been fully investigated. In order to satisfy heterogeneous requirements of communication, computation and storage in IoVs, this article constructs an energy-efficient scheduling framework for MEC-enabled IoVs to minimize the energy consumption of RSUs under task latency constraints. Specifically, a heuristic algorithm is put forward by jointly considering task scheduling among MEC servers and downlink energy consumption of RSUs. To the best of our knowledge, this is a prior work to focus on the energy consumption control issues of MEC-enabled RSUs. Performance evaluations demonstrate the effectiveness of our framework in terms of energy consumption, latency and task blocking possibility. Finally, this article elaborates some major challenges and open issues toward energy-efficient scheduling in IoVs.

Introduction

Due to the rapid development of sensors and intelligent vehicles, Internet of Vehicles (IoV) is becoming a fundamental technology for intelligent transportation systems, calling for mobility and safety improvements on roads [1]. As shown in Fig. 1, IoV supports various kinds of communication patterns, for example, Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Sensor (V2S). On one hand, traffic congestion, caused by the ever-increasing number of vehicles, results in around 5 percent of fuels purchased by Americans are wasted. According to a report from the U.S. energy information administration, totally about 142.86 billion gallons were consumed in the United States in 2018 (https://www.eia.gov/tools/faqs/faq.php?id=23&t=10). On the other hand, each vehicle generates 30 TB of data on average in one day, resulting in the traffic demands from vehicles to the Internet sky rocketing.

A large number of services are with distinct location characteristics and latency requirements, such as entertainment resource sharing, location-based services, public emergency and safety, smart grid and transportation. The produced huge amount of data requires the coordination of computing resources and flexible network management. Road side units (RSUs) have been widely deployed along the main streets to provide wide coverage of network access. By caching contents before delivering them to the vehicles passing by, RSUs can play as agents for information dissemination without fetching from backhauls. However, smart grids and RSUs are not always connected in rural areas. Therefore, efficient utilization of RSU resources is a realistic problem, and deserves to be well investigated to cope with the mass service requests from vehicles.

Because a majority of vehicular applications are delay-sensitive, how to provide real-time communications is rather challenging in IoVs. By placing the communication and computing resources close to end terminals, Mobile Edge Computing (MEC) is becoming an emerging ecosystem, extending the centralized cloud computing capability to the edge in proximity to terminals. An illustration of a MEC-enabled application is illustrated in Fig. 2. Only the image acquisition part is processed at the mobile terminal, while other parts can be offloaded to MEC servers. MEC is flexible to locally optimize network resources and host computing-intensive applications, so that computing at the edge can affect network throughput in terms of execution latency and energy consumption in distinct ways. Our designed method can be leveraged for computation-intensive and latency-critical tasks, such as face recognition applications and augmented reality applications.

On the vehicle side, the handheld devices of passengers or drivers in moving vehicles and even vehicles themselves can offload tasks of applications to MEC servers. The offloaded tasks of applications can be expressed by packages, and several device ports can be utilized to monitor the sending and receiving data flows. On one hand, local computing can significantly lower the execution time based on energy consumption. On the other hand, from the economic and scalable viewpoints, the computing ability of the

Zhaolong Ning is with Chongqing University of Posts and Telecommunications, Xidian University, and Dalian University of Technology; Jun Huang (corresponding author) and Lei Guo are with Chongqing University of Posts and Telecommunications; Xiaojie Wang is with The Hong Kong Polytechnic University; Joel J. P. C. Rodrigues is with National Institute of Telecommunications, Brazil, Instituto de Telecomunicações, Portugal, and Federal University of Piauí.

Digital Object Identifier: 10.1109/MNET.2019.1800309

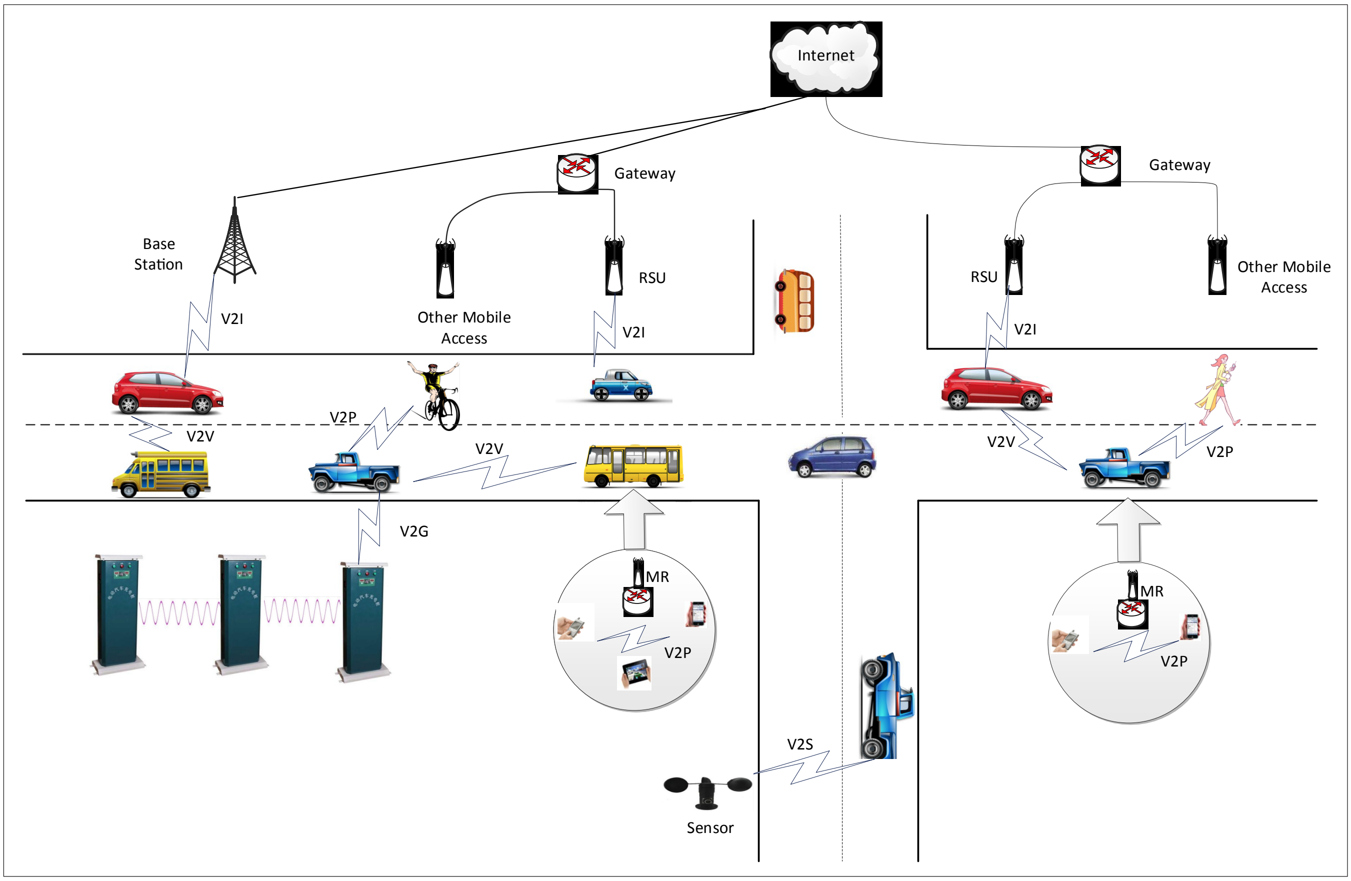

FIGURE 1. An illustration of communication patterns in IoV systems.

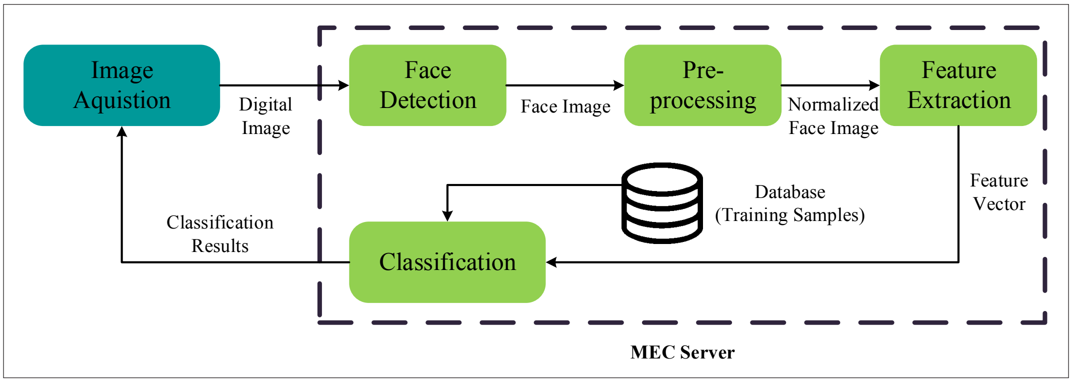

FIGURE 2. An illustration of a MEC-enabled application [2].

MEC server is constrained, and computation offloading in highly dense and heterogeneous IoVs can cause unexpected transmission delay owing to serious network interferences. Therefore, it is not always the best choice to offload all tasks to the MEC server. The trade-off between execution latency and energy consumption of tasks deserves to be investigated, especially in heterogeneous IoVs with distinct energy and computing capacities.

The ever-increasing number of network infrastructures leads to a shift of energy consumption from user devices to connected infrastructures via wireless links in IoV systems. With the objective of minimizing the total energy consumption of MEC-enabled RSUs under latency constraints, this article first establishes an energy-efficient scheduling framework in IoVs to balance computing tasks among RSUs. Due to the computation complexity of the formulated problem, a heuristic algorithm is further put forward, which jointly considers task scheduling among MEC servers and downlink energy consumption of RSUs. To the best of our knowledge, it is a prior work to focus on the energy consumption control issue of MEC-enabled RSUs.

The rest of this article is organized as follows. We first provide an overview of several MEC-enabled research aspects for energy-efficient IoVs. After that, we elaborate the designed energy-efficient scheduling for MEC-enabled RSUs. Performance evaluations illustrate the improvement of our solution. Finally, we discuss some research challenges and open issues toward MEC-enabled IoVs.

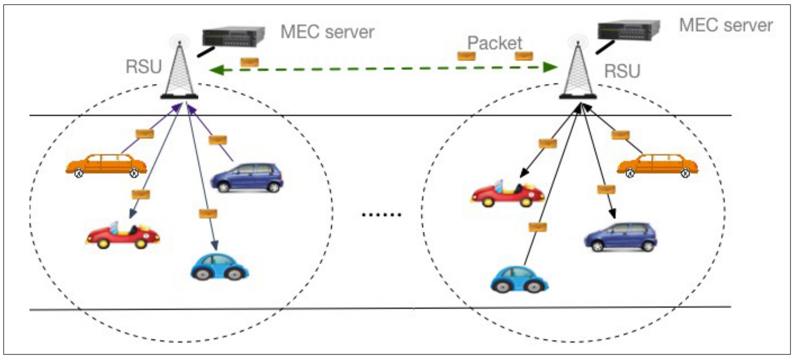

FIGURE 3. System model.

Overview of MEC-Enabled Energy-Efficient IOVS

Energy-efficient scheduling in IoVs embraces distinct kinds of technologies, such as the flexible power control strategy, the energy-efficient protocol design, the energy-harvesting framework and the energy storage system. This section mainly discusses the following four promising aspects for energy-efficient scheduling in IoVs from the viewpoint of MEC technologies.

MEC-Enabled Offloading

Computation offloading is critical for energy saving and computation process acceleration in IoVs. A major question is whether to offload a particular task by MEC or not. Due to the limitation of computing and hardware capabilities of mobile devices, another question is whether to offload their network loads fully or partially. It is challenging to determine how many tasks should be offloaded, since this number is influenced by various factors, such as the capabilities of the cloud and user terminals, users' preferences and backhaul connection. The objective of partial offloading is to achieve a trade-off between energy consumption and execution delay. A MEC-enabled offloading method in IoVs is presented in [3] to choose suitable MEC servers for task management, which comprehensively takes vehicle mobility and computation tasks into consideration for offloading decisions. The authors in [4] attempt to minimize both energy consumption and execution latency for mobile devices, by jointly optimizing the computation speed and the transmission power of devices. However, only one user circumstance is considered.

Collaborative MEC

It is almost impossible for resource-constrained devices to handle all their computing tasks. Thus, collaborative MEC is promising to integrate heterogeneous computing and storage resources among connected entities (e.g., vehicles). Three promising use cases (i.e., mobile edge orchestration, multi-layer interference cancellation, and collaborative processing and caching) are introduced for collaborative MEC in delay-sensitive networks (e.g., 5G and vehicular networks) [5]. The core idea of vehicular fog computing is to integrate the underutilized resources of vehicles (e.g., vehicles in the parking lot) in a cooperative manner for task computing [6]. A Device-to-Device (D2D) enabled collaborative MEC framework is presented in [7] to execute energy-efficient tasks, supporting mobile data offloading, mobile data stream processing and D2D-enabled cloud offloading. With the objective of relieving urban traffic congestion, a traffic management architecture is constructed in [8] by comprehensively considering the technologies of vehicular networks, software defined networks, MEC and 5G networks. A succeeding accident rescue application demonstrates its high-efficiency road accident responses.

Green MEC

The dense deployment of MEC servers is energy consuming, challenging the design of green MEC infrastructures. Computation and radio resources should be well managed, not to mention the unpredictable computation workload in MEC servers. Some promising green MEC schemes are summarized in [2], such as energy-proportional MEC according to computation loads, geographical load balancing for MEC by considering network spatial diversities, and renewable MEC frameworks. V2V-based transmissions are investigated for green cities in [1]. An English-auction is leveraged to match preferences and avoid conflict of users, by which the energy efficiency of V2V-based and cellular networks can be promoted iteratively. RSU assignment for downlink traffic scheduling is investigated in [9]. Various online solutions are provided to minimize the maximum consumed energy by RSU scheduling. Since edge devices in MEC-enabled networks may become inefficient owing to network failure or inter-channel interferences, MEC networks should be survivable to satisfy dynamic task requirements in IoV systems.

Energy-Efficient Scheduling

In order to minimize the energy cost of RSUs, the knowledge of vehicular routes is very helpful. In [10], the authors utilize recent historical traffic data for vehicle routing decision prediction, so that vehicle requests to be scheduled can be balanced. MEC plays a key role in IoVs, where vehicles connect to RSUs equipped with MEC servers to perform real-time or significant computation tasks. Although some MEC-enabled scheduling efforts have been made, they mainly concentrate on reducing energy consumption of users' mobile devices. The corresponding study on energy-efficient scheduling for MEC-enabled RSUs is still in the initial stage. With the objective of promoting the computational capabilities of vehicles, a MEC-enabled framework is designed by considering deep reinforcement learning based solutions [11]. However, most existing works merely consider scheduling among vehicles while ignoring the counterpart among RSUs. How to leverage the limited number of RSUs to cope with the ever-increasing traffic requirements of vehicles is very challenging.

Energy-Efficient Scheduling for MEC-Enabled RSUs

This section specifies the system model and formulates the energy-efficient scheduling problem. Due to the computation complexity of the formulated problem, a heuristic algorithm is further presented, aiming at reducing the energy consumption of MEC-enabled RSUs while satisfying the latency requirements of computation tasks in IoVs.

Computation offloading is critical for energy saving and computation process acceleration in IoVs. A major question is whether to offload a particular task by MEC or not. Due to the limitation of computing and hardware capabilities of mobile devices, another question is whether to offload their network loads fully or partially.

System Model

The system model is illustrated in Fig. 3. For simplicity, M RSUs are deployed along a road. RSUs are connected by fault-free and delay-free wired links, and they periodically broadcast their current network status, such as traffic loads and computation abilities, to other RSUs. When a vehicle is under the coverage range of several RSUs, it sends tasks to the nearest RSU, since the transmission delay and transmitted power are lower than those further RSUs. After the RSU receives a task, it can be scheduled among RSUs. Each RSU is equipped with a MEC server, and its computational ability is distinct. Each MEC server can be modeled as a $M/G/n/k$ queueing model according to the queueing theory. In addition, vehicular flows are considered to be bidirectional on the road.

In the designed system, the passing vehicles can upload their computation tasks to a RSU when they move into its wireless coverage area. Then, the RSU estimates the service time for these computation tasks based on its incoming traffic flows and processing ability. It is assumed that the vehicular computation tasks are atomic and cannot be split anymore. A task can be represented by three elements, that is, the required amount of computation resources, the task size, and the maximum tolerated delay. The transmission time of the computation task from the vehicle to a RSU depends on the channel status. The service time for a task includes the sum of the queueing time and the processing time by the MEC server. If the estimated service time of a RSU is longer than the maximum tolerated delay, the computation task can be transferred to other RSUs through multihop transmissions. The total time for transmitting a task among RSUs can be obtained by the transmission time between each two neighboring RSUs multiplying the total hop count. Generally, the size of the computation result is small, and its transmission time in the network can be ignored. Therefore, the total delay for task processing contains three parts, that is, the transmission time from the vehicle to a RSU, the service time on a RSU, and the transmission time among RSUs.

The energy consumed by task uploading from a vehicle to a RSU is ignored for simplicity, because it mainly consumes the vehicle's energy and has little impact on the energy consumption of the RSU [12]. The total energy consumption for the task contains three parts, that is, the consumed energy for task processing, the consumed energy for task transmission among RSUs, and the downlink consumed energy for feeding back computation results to vehicles. The first part depends on the required amount of computation resources. The second part has a direct relationship with the energy consumed by task transmissions among RSUs. Specifically, the distance and total transmission hop among RSUs before results returning back to vehicles are the main consideration. The third part is directly proportional to the distance between the RSU and the vehicle. In order to minimize the energy consumption of MEC-enabled RSUs, the designed system intends to minimize the total energy consumption for all network tasks, satisfying the constraint of the maximum tolerated delay.

MEC-Enabled Energy-Efficient Scheduling

In order to both minimize the energy consumption of MEC-enabled RSUs and satisfy latency constraints of tasks, this subsection presents a MEC-enabled Energy-Efficient Scheduling (MEES) scheme, which jointly considers task scheduling among MEC servers and downlink energy consumption of RSUs. It contains four steps, that is, delay estimation, energy consumption estimation, task scheduling and processing, and result feeding back. The possible delay caused by MEC servers for task processing is estimated first, and then the energy consumption for MEC servers is computed while satisfying the latency constraint. After that, the computation tasks among MEC servers are scheduled and the required tasks are processed in suitable MEC servers. Finally, results are fed back to target vehicles in an energy-efficient manner to reduce the downlink energy consumption of RSUs. The involved steps are specified in the following.

Delay Estimation: When a vehicle goes into the communication range of a RSU, it can upload the corresponding information of computation task, current location, speed and moving direction to the RSU. After receiving the uploaded information, the RSU begins to search candidate RSUs that can process the task before its maximum delay tolerance. First, the RSU computes the possible location of the vehicle within the tolerated delay time. After a given time period, it is assumed that the vehicle moves in one direction of the road, and there are some RSUs from its initial location to the current location. After that, candidate RSU group R can be obtained. According to the historical beacon information broadcasted by RSUs, task delay processed by RSUs in R can be calculated. Therefore, a group of RSUs can be collected, satisfying the condition that the task can be finished before its maximum tolerated delay. Otherwise, the other direction starts to be searched.

Energy Consumption Estimation: After finding the RSU group, the energy consumption for task processing by RSUs in R can be estimated. For one RSU, its energy consumption for task processing and transmission can be obtained. When task processing is finished by the RSU, the location of the target vehicle can be obtained according to its speed and previous location information. Then, the nearest RSU to the target vehicle can be found. The energy consumption for transmitting the result from this RSU to the nearest one, as well as that from the nearest RSU to the vehicle, can also be estimated, respectively. After obtaining all the energy consumption for RSUs in R, these results are sorted in an ascending order, and the first RSU in R is selected for task processing.

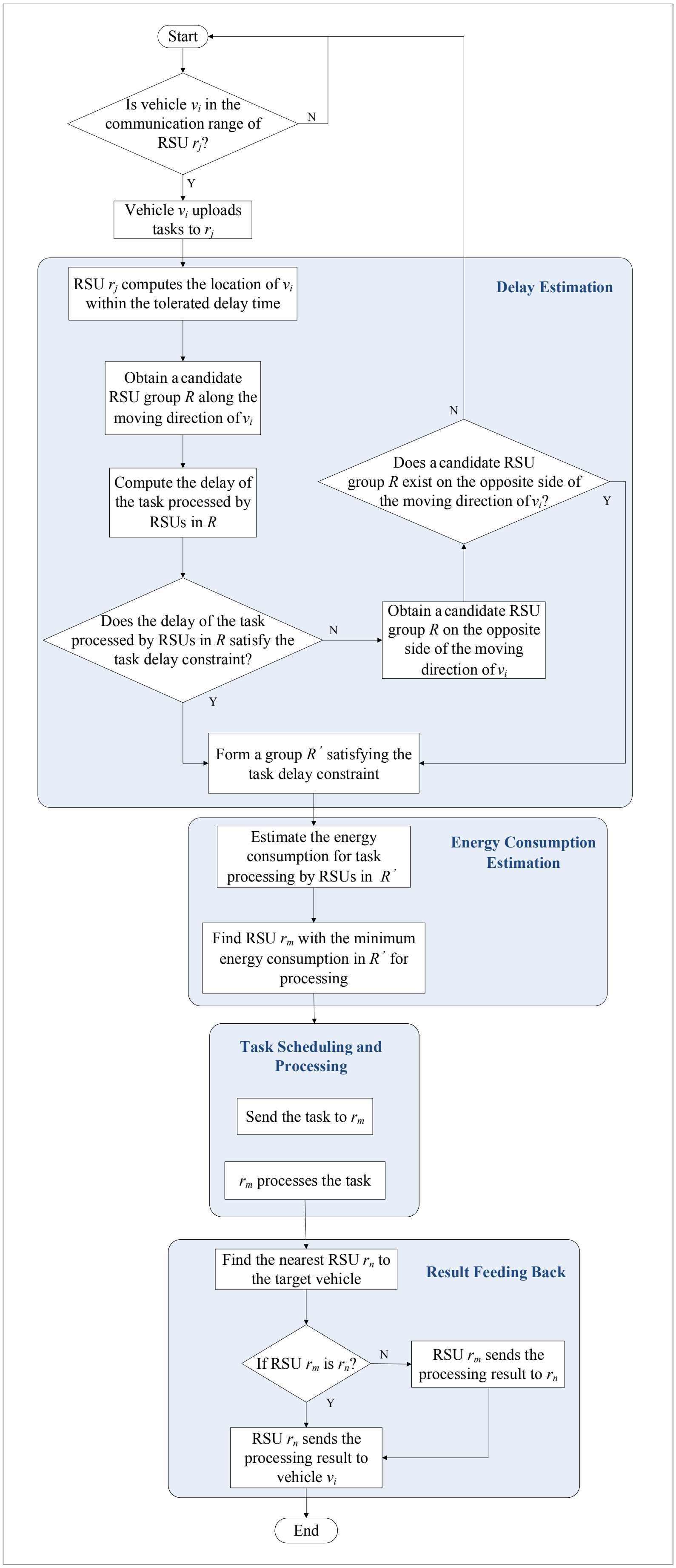

FIGURE 4. The flowchart of our MEC-enabled energy-efficient scheduling method.

Task Scheduling and Processing: According to the estimation of delay and energy consumption, tasks among MEC servers are scheduled based on the computation results. If the energy consumption of one RSU for task processing is the same as that of the first RSU in *R*, this RSU processes the task locally and does not need to transmit the task to other RSUs. In each MEC server, tasks are processed according to the ascending order of their residential lifetime. During each process, the MEC server picks a task with the minimum residential life time into its waiting queue, and processes it to reduce the package blocking possibility.

Result Feeding Back: After the RSU finishes task processing, it transmits the result to the nearest RSU, so that the obtained result can be forwarded to the target vehicle. In this way, the downlink energy consumption can be largely reduced. The flow chart of our designed method is illustrated in Fig. 4.

Performance Evaluation

Since the vehicles entering the road follow a Poisson process, Monte Carlo simulations are leveraged to randomly generate the input (traffic flow), and obtain the statistical value for the output. Similar to [13], a path loss channel model is considered. The shadowing gain is simplified to be a constant value during one scheduling period, while the small-scale fading gain is normalized to a random value with unit mean. Since the channel resource is contended by multiple vehicles, a signal-to-interference-plus-noise ratio based model is considered, that is, if the calculated value is above the threshold, transmission links between vehicles and RSUs can be activated. The access layer technology is 802.11p, and the transmission power of nodes ranges between 10 dBm and 30 dBm.

The collected real-world taxi traces along G25 highway, from Hangzhou to Ningbo (China) between September 2017 and October 2017, are leveraged for evaluation, including the real-time records of latitudes and longitudes of vehicles, their speeds, directions, and the recorded time. Ten RSUs were simulated along the highway. The wireless communication range of a RSU is 250m. The average task arrival rate at each RSU is 0.3 vehicle per second, and the rate of the generated task is three tasks per second. The generated task is delivered to the waiting queue of the nearest RSU for processing. In order to demonstrate the effectiveness of our proposed scheme, two algorithms are considered as follows:

All Task Admission Algorithm (ATAA) [3]: The MEC server accepts all the computing tasks and puts them in the waiting queue. Then, the server processes these tasks in sequence, while they cannot be transferred among RSUs. Therefore, tasks will be dropped if vehicles move out of the wireless communication range of the RSU.

GMCF [12]: An optimization solution to minimize the energy consumption for downlink traffic of RSUs by minimizing cost flow while satisfying task latency requirements.

Figure 5 shows the performance of MEES, ATAA and GMCF with different distance between RSUs. In Fig. 5a, it is obvious that when the distance between two neighboring RSUs increases, the task blocking possibility increases. This is because long distance between two neighboring

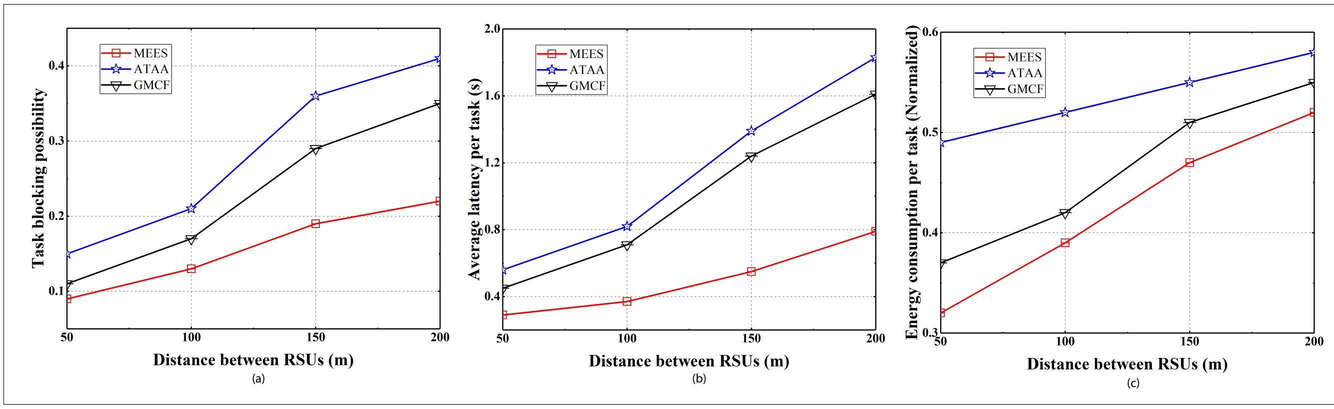

FIGURE 5. Performance in terms of distance between RSUs: a) task blocking possibility; b) average latency per task; c) energy consumption per task.

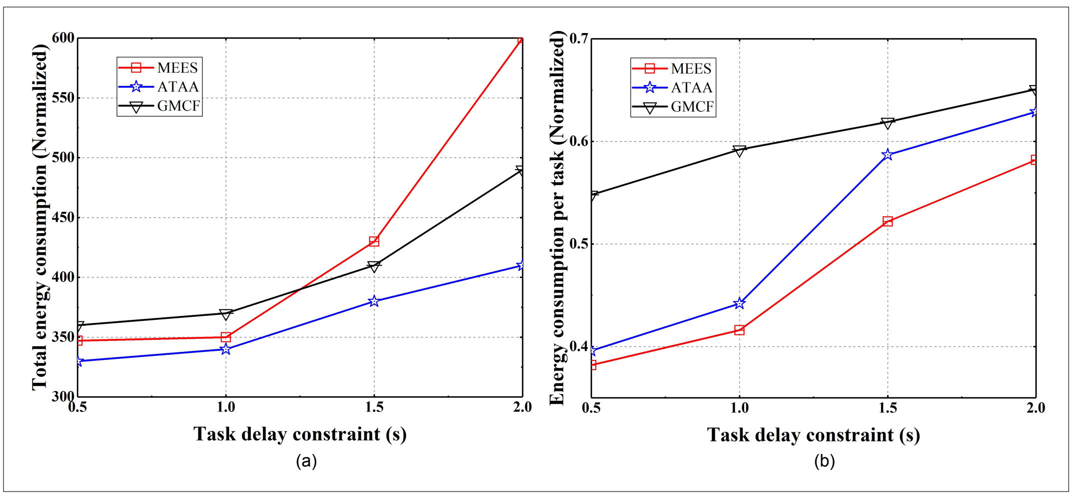

FIGURE 6. Relationship between task delay and energy consumption: a) total energy consumption; b) average energy consumption.

RSUs leads to few RSUs available for task processing in the road network. The task blocking possibility of MEES is much lower than those achieved by ATAA and GMCF, especially when the distance between RSUs is large. The reason is that computing tasks can be transferred among RSUs to select a suitable MEC server in MEES. However, the other two algorithms do not allow transmissions among RSUs. Similar trends can be obtained in Fig. 5b. This is because when the distance among RSUs increases, fewer tasks can be processed by MEC servers and more time is consumed for resource scheduling among MEC servers. In Fig. 5c, we notice that the consumed energy per task by MEES is lower at the beginning but increases quickly. This is because with more tasks to be handled, our method needs to balance tasks among MEC servers, so that tasks can be handled within the delay constraint. Although some additional energy consumption is required, more tasks can be processed by scheduling. Different from the MEES and GMCF methods, the consumed energy by ATAA increases slightly as the distance between two neighboring RSUs increases. This is because a large RSU distance results in few available RSUs existing in the communication range of vehicles, and few tasks can be processed by ATAA.

Finally, we compare the relationship between the consumed energy and the delay constraint in Fig. 6. Fig. 6(a) illustrates the total consumed energy. It can be observed that when the delay constraint increases, the consumed energy also increases, especially for MEES. After the delay constraint is above a threshold (around 1.2s), the total consumed energy by MEES increases sharply. This is because a large task delay allows tasks to be scheduled among RSUs, and more tasks can be fulfilled at the cost of additional energy consumption. For the other two schemes, tasks are processed locally so that the consumed energy increases gradually. We notice that the consumed energy by ATAA is the lowest, because a large delay constraint causes few tasks to be handled. Fig. 6(b) shows the average consumed energy per task. Although the total consumed energy of our method is the highest, the average consumed energy is the lowest since the requirement of most tasks can be fulfilled.

Research Challenges and Open Issues

Since the study of energy-efficient scheduling for MEC-enabled IoVs is at the very beginning, this section discusses some research challenges and open issues.

Renewable Energy Supplement for RSUs

Apart from energy provided by power grids, the renewable energy supplement for RSUs (such as wind or solar-powered RSUs) is promising

Real-time decisions, such as driving decision and offloading decision, are challenging for data-driven network scenarios. Other challenges include how much information should be learned by the system to make a suitable decision, and how to acquire the learned information.

for energy-constrained vehicular circumstances, and renewable energy based RSUs will occupy around 40 percent on rural freeways by 2050 [3]. However, the collection of the renewable energy is intermittent due to its relative random arrival. The unbalanced power supply and the high demand of the renewable energy make the corresponding energy supplement for RSUs very challenging. For example, traffic demands in IoVs are heavy during the evening peak period while the solar-enabled RSUs cannot offer sufficient energy. Another major challenge is to fulfill delay tolerated traffic transmission with the minimum energy consumption. One promising solution is to dynamically schedule tasks among several RSUs, and another choice is to jointly optimize radio and computational resources. In addition, how to leverage free resources of vehicles to form edge nodes and construct an energy-efficient model is significant to release the energy requirement of RSUs.

Sustainable and Survivable MEC

Due to the unpredictable computation workload in RSUs and MEC servers, computation performance and radio resources should be well managed. With the development of electric vehicles (EVs), efficient utilization of their computing resources is challenging due to their mobility. For example, when EVs leave their current access networks before completing task offloading or computation, the connected communication links will be interrupted. Therefore, link reconstruction for searching and routing has to be performed, which is both time consuming and energy wasting. In addition, edge devices deployed at MEC networks may become inefficient owing to network failure or inter-channel interferences. Therefore, MEC should be survivable to satisfy various and dynamic task requirements.

Incentive and Trusted Offloading

Since resources in MEC-enabled IoVs are limited, how to stimulate users to share their local resources is challenging. Incentive mechanisms based on competition-based pricing models, for instance, double auction and non-cooperative game, are promising for computation offloading, traffic forwarding, and facility utilization in MEC-enabled IoVs. A dynamic pricing strategy charged by operators is preferred to regulate the switch of RSUs between active and idle states for the sake of maximizing profit while minimizing congestion. However, by content sharing and traffic offloading among RSUs, security issues may become serious, such as weak authentication and privacy leakage. The reason is that MEC-enabled IoVs are obviously weaker than central clouds with sufficient resources to protect themselves. In addition, a device supporting MEC can be a vehicle on the road or a sensor in the smart house, which does not have global knowledges to detect threats. Crowd sourcing/sensing in a trusted edge server is promising to protect a user's privacy in location-based services.

Deep Learning Based Scheduling

Due to the insufficient resources of edge devices, the realization of deep learning based solutions for IoVs with strict delay limitation is very challenging. Currently, only a tiny fraction of neurons and brains are leveraged for the deep learning based framework. According to the report by Intel, an autonomous vehicle can generate up to 4000GB of data each day [14]. Real-time and online deep learning algorithms are significant to cope with the huge generated data. Real-time decisions, such as driving decision and offloading decision, are challenging for data-driven network scenarios. Other challenges include how much information should be learned by the system to make a suitable decision, and how to acquire the learned information.

Conclusions

Since the connection between RSUs and smart grids is either expensive or not always available in many areas, efficient utilization of RSU resources is significant. Energy-efficient scheduling of RSUs is promising for MEC-enabled IoVs, which has various benefits for delay-sensitive and energy-consuming applications. In order to minimize the total energy consumption of RSUs under latency constraints, we present the MEES method, including the processes of delay estimation, energy consumption estimation, task scheduling and processing, and result feeding back. To the best of our knowledge, it is a prior attempt for the energy consumption control of MEC-enabled RSUs. After that, performance evaluations demonstrate its effectiveness in terms of task blocking possibility, latency and energy efficiency. Finally, we discuss some research challenges and open issues, concentrating on the aspects of energy management of RSUs and high-efficient offloading solutions. In further work, we will focus on developing artificial intelligence based methods to jointly optimize energy consumption and task delay according to history records.

Acknowledgments

This work was supported by the State Key Laboratory of Integrated Services Networks, Xidian University (ISN20-01); the China Postdoctoral Science Foundation (2018T110210); the Fundamental Research Funds for the Central University under grant (DUT19JC18); national funding from the Fundação para a Ciência e a Tecnologia (FCT), through the UID/EEA/50008/2019 Project; by RNP, with resources from MCTIC, Grant no. 01250.075413/2018-04; under the Centro de Referência em Radiocomunicações (CRR) project of the Instituto Nacional de Telecomunicações (Inatel), Brazil; and by the Brazilian National Council for Research and Development (CNPq) via Grant no. 309335/2017-5.

References

[1] X. Wang *et al.*, "Privacy-Preserving Content Dissemination for Vehicular Social Networks: Challenges and Solutions," *IEEE Commun. Surveys & Tutorials*, vol. 21, no. 2 , Second Quarter 2019, pp. 1314–45.

[2] Y. Mao *et al.*, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Commun. Surveys & Tutorials*, vol. 19, no. 4, 2017, pp. 2322–58.

[3] K. Zhang *et al.*, "Mobile-Edge Computing for Vehicular Networks: A Promising Network Paradigm with Predictive Offloading," *IEEE Vehicular Technology Mag.*, vol. 12, no. 2, 2017, pp. 36–44.
[4] Y. Wang *et al.*, "Mobile-Edge Computing: Partial Computation Offloading Using Dynamic Voltage Scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, 2016, pp. 4268–82.
[5] Z. Ning *et al.*, "A Cooperative Partial Computation Offloading Scheme for Mobile Edge Computing Enabled Internet of Things," *IEEE Internet of Things J.*, vol. 6, no. 3, June 2019, pp. 4804–14.
[6] Z. Ning *et al.*, "Vehicular Fog Computing: Enabling Real-Time Traffic Management for Smart Cities," *IEEE Wireless Commun.*, vol. 26, no. 1, 2019, pp. 87–93.
[7] X. Chen *et al.*, "Exploiting Massive D2D Collaboration for Energy-Efficient Mobile Edge Computing," *IEEE Wireless Commun.*, vol. 24, no. 4, 2017, pp. 64–71.
[8] J. Liu *et al.*, "High-Efficiency Urban Traffic Management in Context-Aware Computing and 5G Communication," *IEEE Commun. Mag.*, vol. 55, no. 1, 2017, pp. 34–40.
[9] A. Khezrian *et al.*, "Energy-Efficient Scheduling in Green Vehicular Infrastructure with Multiple Roadside Units," *IEEE Trans. Vehicular Technology*, vol. 64, no. 5, 2015, pp. 1942–57.
[10] X. Wang *et al.*, "Optimizing Content Dissemination for Real-Time Traffic Management in Large-Scale Internet of Vehicle Systems," *IEEE Trans. Vehicular Technology*, vol. 68, no. 2, 2019, pp. 1093–1105.
[11] Z. Ning *et al.*, "Deep Reinforcement Learning for Vehicular Edge Computing: An Intelligent Offloading System," *ACM Trans. Intelligent Systems and Technology*, DOI: 10.1145/3317572, 2019.
[12] A. Hammad *et al.*, "Downlink Traffic Scheduling in Green Vehicular Roadside Infrastructure," *IEEE Trans. Vehicular Technology*, vol. 62, no. 3, 2013, pp. 1289–1302.
[13] J. Zhang *et al.*, "On Fast Optimal STDMA Scheduling over Fading Wireless Channels," *IEEE INFOCOM*, 2009, pp. 1710–18.
[14] Z. Fadlullah *et al.*, "State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrows Intelligent Network Traffic Control Systems," *IEEE Commun. Surveys & Tutorials*, vol. 19, no. 4, 2017, pp. 2432–55.

Additional Reading

[1] W. Atoui *et al.*, "Offline and Online Scheduling Algorithms for Energy Harvesting RSUs in VANETs," *IEEE Trans. Vehicular Technology*, vol. 67, no. 7, 2018, pp. 6370–82.

Biographies

Zhaolong Ning [M'14, SM'18] received the M.S. and Ph.D. degrees from Northeastern University, China in 2011 and 2014, respectively. He was a research fellow with Kyushu University, Fukuoka, Japan. He is a professor with Chongqing University of Posts and Telecommunications, China. He has authored or co-authored over 100 scientific papers in international journals and conferences, including 3 ESI 0.1 percent hot papers, 12 ESI 1 percent high-cited papers, one journal best paper, and four conference best papers. His research interests include vehicular networks, edge computing, and network optimization.

Jun Huang received the Ph.D. degree (Hons.) from the Institute of Network Technology, Beijing University of Posts and Telecommunications, China, in 2012. He is currently a professor with Chongqing University of Posts and Telecommunications, China. He has authored more than 80 refereed papers. His current research interests include network optimization and control and quality of service. He has chaired and co-chaired several conferences in the communications and networking areas and organized multiple workshops at major IEEE and ACM events.

Xiaojie Wang received the Ph.D. degree from Dalian University of Technology, China, in 2019. From 2011 to 2015, she was a software engineer at NeuSoft Corporation, China. Currently, she is a postdoc research fellow with The Hong Kong Polytechnic University, Hong Kong. She has authored more than 20 refereed papers. Her research interests are vehicular social networks and mobile edge computing.

Joel J. P. C. Rodrigues [S'01, M'06, SM'06] is a professor at the National Institute of Telecommunications (Inatel), Brazil; a senior researcher at IT, Portugal; and a visiting professor at Federal University of Piau, Brazil. He is the leader of the Internet of Things Research Group (CNPq), Director for Conference Development-IEEE ComSoc Board of Governors, IEEE Distinguished Lecturer, Technical Activities Committee Chair of the IEEE ComSoc Latin America Region Board, the Past-Chair of the IEEE ComSoc TCs on eHealth and on Communications Software, Steering Committee member of the IEEE Life Sciences Technical Community. He is the editor-in-chief of an international journal and editorial board member of several journals. He has authored or co-authored over 700 papers in refereed international journals and conferences, three books, and two patents.

Lei Guo [M'06] received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2006. He is currently a full professor with Chongqing University of Posts and Telecommunications, Chongqing, China. He has authored or co-authored more than 200 technical papers in international journals and conferences. He is an editor for several international journals. His current research interests include communication networks, optical communications, and wireless communications.