JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

VEHICULAR CLOUD FORMING AND TASK SCHEDULING FOR ENERGY-EFFICIENT COOPERATIVE COMPUTING

Mrs. J. SRAVANTHI - Assistant Professor, Department of CSE, Anurag College of Engineering (Aushapur, Ghatkesar, Telangana 501301)

Ms. PITTALA SHARANYA KRUTHI - Student, Department of CSE, Anurag College of Engineering (Aushapur, Ghatkesar, Telangana 501301)

Ms. RAMAVATH SAI NIKITHA - Student, Department of CSE, Anurag College of Engineering (Aushapur, Ghatkesar, Telangana 501301)

Mr. MADANU BALA PRAKASH - Student, Department of CSE, Anurag College of Engineering (Aushapur, Ghatkesar, Telangana 501301)

Abstract

A vehicular cloud (VC) is a network of vehicles that perform cooperative computing through vehicle-to-vehicle (V2V) communication. Existing research on vehicular cloud computing (VCC) is mostly based on cloud servers or edge servers, not VCs. However, vehicles, by constructing a Vehicular Ad-Hoc Network (VANET), can perform applications requiring the large amount of computation cooperatively on their own without the help of edges or cloud servers. One of important issues for the VANET cooperative computing is how to handle the frequent topology change due to vehicle mobility. The unstable network topology limits the advantage of cooperative computing and even makes its operation stop sometimes. This paper proposes a cooperative computing method based on vehicle-to-vehicle (V2V) communication. For stable and energy-efficient cooperative computing, the proposed method considers the distance when selecting vehicles that it will cooperate with and delays task offloading back as far as possible. The proposed method outperforms previous static scheduling methods in terms of energy efficiency and network stability.

1. INTRODUCTION

Recently, vehicles have not only been connected to Internet via wireless communication, but have also become network nodes that can perform various applications in real-time [1]. Particularly, applications such as big data analysis and image processing using machine learning require high computing power. Vehicular Cloud Computing (VCC) might be a promising solution to run these kinds of application smoothly [2], [3], [4].

The VCC has a 3-tier architecture of Cloud-tier, Edge-tier, and Vehicle-tier. Each tier has different features, so each is suitable for different VCC applications. First, the Cloud-tier has the highest computing power and largest memory, but it may take the longest latency to reach because cloud servers are located far away. Thus, the Cloud-tier is good for applications that require large amounts of memory and computation but do not require short latency, such as infotainment services. Information services are information and entertainment services that are not related to driving safety, requiring a latency of 1 second and a bandwidth of 80 Mbps or more [5].

On the contrary, the Vehicle-tier consists of vehicles. Because a vehicle can receive data from other vehicles directly, this tier requires the shortest latency. Therefore,

this tier is appropriate for applications requiring very short latency like collision warning of which the acceptable end-to-end delay is just 20-50 ms [5]. However, this tier is not good for performing computation-intensive applications due to the lack of computing power. Finally, the Edge-tier can provide shorter latency than the Cloud-tier; and more computing power and memory than the Vehicle-tier. Thus, it is suitable for applications such as traffic information sharing and analysis services, which need delay of 100-500 ms and a throughput of 10-45 Mbps

Due to the moderate latency and computing power, some papers [6], [7] suggest that most of the computation work on vehicles is offloaded to edge servers. However, edge servers always have certain constraints on both computing power and memory unlike cloud servers, thus if too many tasks are given to edge servers, the service quality might be degraded and even the security might be endangered [6], [8]. In this situation, vehicles can relieve the burden of edge servers. Vehicles in the previous VCC model [9] were considered as just end-terminals sending requests to cloud or edge servers, but this is not all that a vehicle can do. If they construct a Vehicular Ad-hoc Network (VANET) and cooperatively run many applications by themselves through vehicle-to-vehicle (V2V) communication, both the burden on edge servers and the network traffic load can be reduced. In addition, it is better for privacy since private information does not pass through external networks [10]. We suggest a platform that enables vehicles to efficiently and reliably perform cooperative computing in VANETs built through V2V communication. In the cooperative computing, tasks of a vehicle are offloaded to other vehicles. The vehicle that requests task offloading is called client vehicle (CV), and the vehicle that helps a CV is called worker vehicle (WV). The task offloading takes two steps: A CV finds some candidate WVs around itself and then distributes tasks to them according to the task execution schedule. The task offloading improves application running speed, but on the other hand, it requires additional cost to transfer data. Minimizing both time and energy for data transfer is a key performance metric for a vehicular cloud. The latest vehicle network modules support both direct V2V communication using Dedicated Short-Range Communication (DSRC) [11] or C-V2X mode4 [12]; and cellular connectivity using LTE C-V2X [13]. The cellular-based connection can cover a wider range, but is expensive, so it is more economical to transmit over V2V connections as much as possible. However, V2V connections may not be reliable because vehicles move fast. The distance between two vehicles in a VANET continuously changes due to their different speed. Whereas the cellular communication can reach several kilometers, the transmission range of the DSRC V2V communication is just from several hundred meters to 1 km, so vehicles can easily be out of the range from each other. In this case, an attempt can be made to resume the communication over the cellular network, but it increases cost and delay. Therefore, in order to improve VANET stability and minimize transmission cost, a CV needs to select WVs that are least likely to go out of communication range of the CV. Another important factor in improving the efficiency of task offloading is task scheduling to find the best order to run tasks. The task scheduling determines the amount of data to be transferred, the associated transfer cost, and eventually the final completion time for all tasks. Most previous studies have focused only on minimization of this execution time. On the other hand, we consider not only execution time but also energy costs when designing our task scheduling method. The energy cost is particularly important for electric vehicles since it can affect their driving range.

2. LITERATURE SURVEY

Task offloading using cloud computing was initially proposed in a mobile environment and then extended to VCC. A typical example of using VCC is the intelligent transportation system (ITS), which provides services to reduce traffic accidents and facilitate traffic flow. Mao et al. presented computation offloading based on the Lyapunov function to minimize cost on edge servers using dynamic voltage and frequency scaling (DVFS) and power control. Assuming that each task was independent of each other, they decided whether to offload each task to edge server or not. Sun et al. proposed a scheduling scheme that runs on vehicular cloud or edge cloud. Their scheme firstly computes the cell dwell time of each vehicle based on the distance to a base station. This cell dwell time is used to form a vehicular cloud, and then tasks are scheduled to minimize the completion time. It was a genetic algorithm- based scheduling method with low complexity considering task dependency for the optimization.

While most studies enable a client to select workers, Zhou et al. suggested a method in which candidate workers choose an actual worker by voting. This voting system works since it is assumed that all workers in this paper are wired edge servers and all other servers' computing resource information is known to all.

However, this system is not suitable for wireless vehicular cloud networks because too many packets must be exchanged between vehicles to share resource information in real time. One of the most important issues in cooperative computing in VCCs is how to schedule the execution of tasks. List scheduling a typical DAG scheduling method, consists of two steps. First, priorities are calculated and assigned to tasks in order. For the priority calculation, the cumulative sum of node cost and edge cost from the starting point to each task is used in general. And in the second step, a node or device that will run each task is determined using its own algorithm.

Heterogeneous Earliest Finish Times (HEFT), a representative list scheduling method, assigns tasks to devices that can complete them in the shortest amount of time, using the insertion-based scheduling. Although HEFT is a static algorithm that requires information about all devices and tasks in advance, it has long been considered a leader in DAG scheduling due to outstanding performance improvement. The duplication-based algorithms create copies of tasks on participating devices to minimize data transfer overhead. Significant performance gains can be expected if the resources of the devices are abundant, but we cannot say that it is efficient given the amount-of resources it uses.

Hu et al. proposed a platform in which dynamically formed vehicle clusters act as edge servers that perform computations for surrounding mobile devices. It is noteworthy that vehicles on this platform do not request offloading, but act as edge servers. An application to be offloaded is scheduled with the Greedy-based Task Scheduling Algorithm (GBTSA), which uses a greedy-based task copy technique, considering the inter-task dependency. This scheduling technique is to apply the task scheduling proposed by to the vehicle cloud. Existing studies mostly assumed offloading to edge servers and proposed scheduling techniques to minimize execution time. This paper attempts to minimize energy consumption and the probability of WVs out of CV's coverage by considering the distance between vehicles when selecting WVs and assigning tasks.

3. OVERVIEW OF THE SYSTEM

3.1 Existing System

Existing systems mostly assumed offloading to edge servers and proposed scheduling techniques to minimize execution time. Due to the moderate latency and computing power, some papers suggest that most of the computation work on vehicles is offloaded to edge servers. While most studies enable a client to select workers, Zhou et al. suggested a method in which candidate workers choose an actual worker by voting.

Heterogeneous Earliest Finish Times (HEFT), a representative list scheduling method, assigns tasks to devices that can complete them in the shortest amount of time, using the insertion-based scheduling.

Another platform is proposed in which dynamically formed vehicle clusters act as edge servers that perform computations for surrounding mobile devices. It is noteworthy that vehicles on this platform do not request offloading, but act as edge servers. An application to be offloaded is scheduled with the Greedy-based Task Scheduling Algorithm (GBTSA), which uses a greedy-based task copy technique, considering the inter-task dependency. This scheduling technique is to apply the task scheduling proposed by to the vehicle cloud.

3.1.1 Disadvantages of Existing System

Assumption of Offloading to Edge Servers: Many existing methods assume tasks are offloaded to edge servers, which may not always be the most efficient approach, considering the limitations of edge servers in computing power and memory.

Limited Flexibility: Some methods rely on predetermined scheduling techniques without considering dynamic factors like varying network conditions or real-time resource availability.

Potential Latency Issues: Offloading tasks to edge servers or remote resources may introduce latency, especially if the communication distance is significant. This latency can impact application responsiveness and user experience negatively.

Complexity in Implementation: Certain scheduling techniques proposed in existing methods may involve complex algorithms or overhead, making their implementation and deployment challenging, especially in real-world vehicular environments.

3.2 Proposed System

This proposed system Stepwise Computation Offloading for Cost-efficient Cooperation (SCOCC) to form a vehicular cloud and find an energy-efficient task schedule. Main idea of SCOCC is to minimize the time interval between task offloading and task execution, and to consider the distance between a CV and WVs when choosing WVs. An application is decomposed into tasks, which are logical minimum units of work, being represented in a Directed Acyclic Graph (DAG) form by a CV. The tasks are then offloaded only to a fixed number of WVs in the order closest to the CV. Note here that not all tasks are offloaded at once, but rather some parts of them

are offloaded sequentially to minimize the time difference between
scheduling and task execution. The proposed method outperforms previous static scheduling methods in terms of energy efficiency and network stability.

3.2.1 Advantages of Proposed System

Minimized Time Interval: This approach aims to reduce the time gap between task offloading and execution, ensuring tasks are swiftly executed without unnecessary delays.
Consideration of Distance: By factoring in the distance between client vehicles (CVs) and worker vehicles (WVs) during the selection of WVs for task offloading, it guarantees tasks are assigned to WVs within communication range. This consideration enhances network stability.
Energy Efficiency: The method takes energy costs into account, which is particularly advantageous for electric vehicles, given its potential impact on their driving range. By optimizing task scheduling to minimize energy consumption, it contributes to prolonging the operation of electric vehicles.
Sequential Offloading: Tasks are offloaded sequentially rather than all at once, which minimizes the time discrepancy between scheduling and task execution. This sequential approach aids in efficiently managing computational load.

3.3 Proposed System Design

In this project work, there are four modules and each module has specific functions, they are:

1. Client Vehicle Module

2. Network Module

3. Task Schedular Module

4. Worker Vehicle Module

3.3.1 Client Vehicle Module

The vehicle that requests task offloading is called client vehicle the task offloading takes two steps: A CV finds some candidate WVs around itself and then distributes tasks to them according to the task execution schedule. The task offloading module will manage task assigned by network module to schedule task and send to WVs. Using this module vehicle client will register with application upload task and send to network manager to complete given task as soon as possible and able to communicate with nearby WVs.

3.3.2 Network Module

Using this module network will login to application and manage WVs by changing communication details and send available details to task scheduler who will manage for assignment and get confirmation to network module.

3.3.3 Task Schedular Module

Using this module task manager will login with application view requests received by network manager and check status of nearby WVs and calculate which WVs are near to client vehicle and send data to that WVs data will be in encrypted format.

3.3.4 Worker Vehicle Module

Using this module worker module will register with application can view details send by task manager and decrypt data and see based on key sent by task manager the nearest worker module will receive data based on network module assignment.

3.4 Architecture

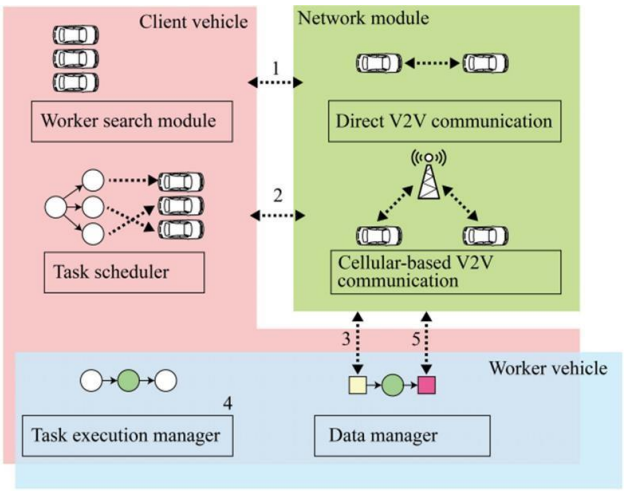
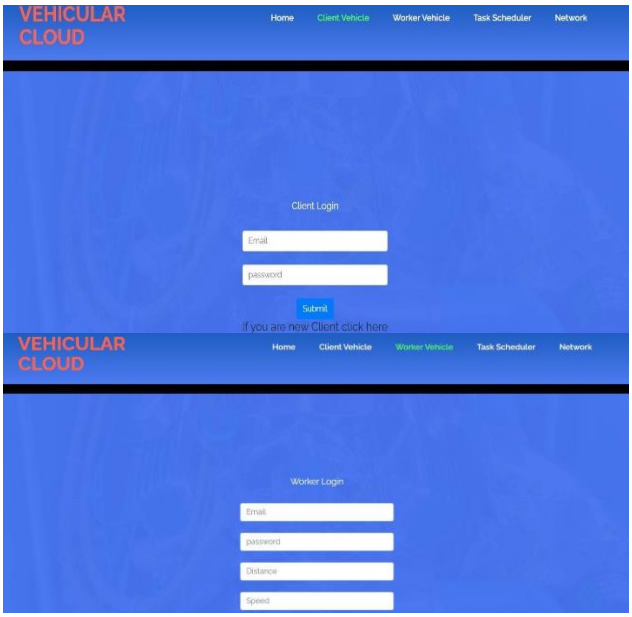
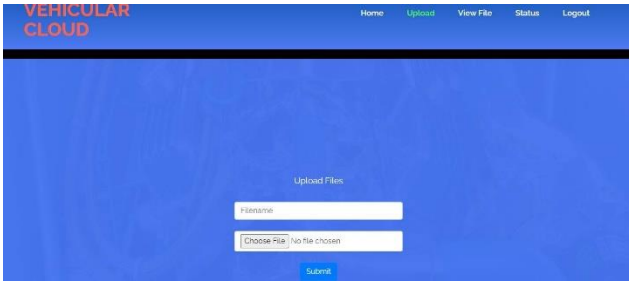
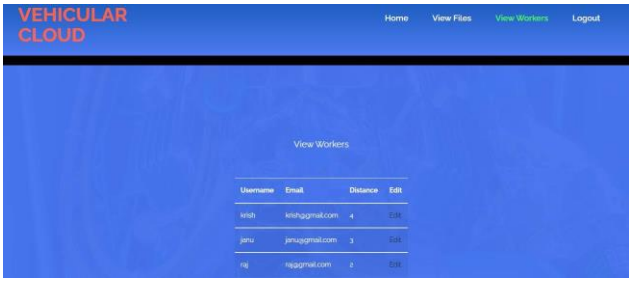


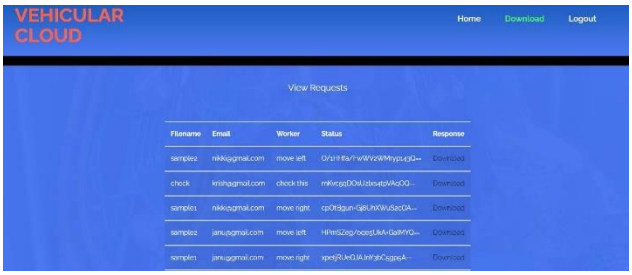
Fig 1: System Architecture

4. RESULT SCREEN SHOTS









5. CONCLUSION

This paper proposed SCOCC to establish a stable VANET cloud and perform an energy-efficient computation offloading between vehicles. To achieve the goal, SCOCC i) delays task assignment as much as possible and ii) considers the distance from a client vehicle when selecting worker vehicles. The reduction of the time interval between task assignment and its execution enhanced the VANET stability, and the reduction of average inter-vehicle distance was helpful for both the stability and energy consumption. Unlike the static algorithms such as HEFT and GBTSA, SCOCC does not try to optimize execution speed of entire tasks at once, so it does not show the better performance in terms of execution time. However, SCOCC achieves essential stability in a fast-moving vehicle environment and minimization of energy consumed for wireless communication. Vehicle to Vehicle communication can be a real game changer, because of its importance on our daily lives, it is a major challenge but it will be resolved with time. It includes a lot of points of interest to declare and develop, starting from the protocol design to obtain a standard communication model, then to go through the performance evaluation to have stable KPIs. After that to explore the implementation phase for commercial usage with zero percent failure, and finally to discover the integration mechanisms to release a full automated ITS solution ready to use for the real life. Vehicle to Vehicle communication real life implementation till now requires RSUs in order to have accurate results and full safety for the society, but due to lack of resources in some of the developing countries most

of them don't have well established infrastructure and on the other side, the accidents and loss of lives are increasing exponentially year by year. Also, the developing countries require to increase their investments, and this depends on providing full automated Intelligent Transportation System including V2V communication in order have daily traffic flow without bottlenecks.

Current effort is being how to envisage the possibilities of enhancing the above-named suggestions starting with tamperproof and onboard secure system performance status application. In a wider way trying to envisaging the Physical and Technical Control of V2V communication to leverage potential solutions and mitigate setback. Significant encryption systems have been put in place yet there is no current security model that is secured enough. There are some concerns in widespread deployment of V2V but progress in technology and anticipated benefits can make V2V happen.

6. REFERENCES

[1] T. do Vale Saraiva, C. A. V. Campos, R. D. R. Fontes, C. E. Rothenberg, S. Sorour, and S. Valaee, ''An application-driven framework for intelligent transportation systems using 5G network slicing,'' IEEE Trans. Intell. Transp. Syst., vol. 22, no. 8, pp. 5247– 5260, Aug. 2021.

[2] G. Dimitrakopoulos and P. Demestichas, ''Intelligent transportation systems,'' IEEE Veh. Technol. Mag., vol. 5, no. 1, pp. 77–84, Mar. 2010.

[3] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, ''Big data analytics in intelligent transportation systems: A survey,'' IEEE Trans. Intell. Trans. Syst., vol. 20, no. 1, pp. 383–398, Jan. 2018.

[4] A. K. Haghighat, V. Ravichandra-Mouli, P. Chakraborty, Y. Esfandiari, S. Arabi, and
A. Sharma, ''Applications of deep learning in intelligent transportation systems,'' J. Big Data Anal. Transp., vol. 2, no. 2, pp. 115–145, 2020.

[5] A. Moubayed, A. Shami, P. Heidari, A. Larabi, and R. Brunner, ''Edge-enabled V2X service placement for intelligent transportation systems,'' IEEE Trans. Mobile Comput., vol. 20, no. 4, pp. 1380–1392, Apr. 2021.

[6] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, ''Joint load balancing and offloading in vehicular edge computing and networks,'' IEEE Internet Things J., vol. 6, no. 3, pp. 4377–4387, Jun. 2018.