Received June 19, 2019, accepted August 2, 2019, date of publication August 12, 2019, date of current version August 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2934770

Multipath Transmission Workload Balancing Optimization Scheme Based on Mobile Edge Computing in Vehicular Heterogeneous Network

ZHAO HAITAO[1,2], (Member, IEEE), DING YI[1,2], ZHANG MENGKANG[1,2], WANG QIN[2,3,4], (Member, IEEE), SHI XINYUE[1,2], AND ZHU HONGBO[1,2,3]

[1]College of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
[2]Jiangsu Key Laboratory of Wireless Communications, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
[3]Engineering Research Center of Health Service System Based on Ubiquitous Wireless Networks, Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
[4]College of Engineering and Computing Sciences, New York Institute of Technology at Nanjing, Nanjing, China

Corresponding author: Zhao Haitao (zhaoht@njupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61771252, in part by the Basic Research Program of Jiangsu Province under Grant BK20171444, in part by The Six Talents High Peaks Funding Project of Jiangsu Province under Grant DZXX-041, in part by the Major Natural Science Research Projects of Jiangsu Universities under Grant 18KJA510005, in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant KYCX19_0949, and in part by the Young Talents Support Project of Jiangsu Scientific and Technological Association, The Object of Jiangsu Province 5th "333 Project," "1311" Talents Funding Project of NUPT.

ABSTRACT With the rapid development of intelligent transportation, various applications which have millisecond delay requirements appear in the vehicular heterogeneous network. Offloading these delay-sensitive applications into edge nodes is a trend and direction of development. However, with the increase in the number of vehicular applications, the conventional methods of distance-based edge node workload allocation make the workload allocation unbalanced, causing some edge nodes to be overloaded, resulting in response time of corresponding application being too long to meet low delay requirements. In this paper, we tackle the problem of edge node overload and propose a multipath transmission workload balancing optimization scheme, which uses multipath transmission in the edge computing architecture as the transport protocol support for the communications between the vehicles and the edge nodes and the real-time virtual machine (VM) migration happens between the edge nodes. First, the application is assigned to the edge node closest to each vehicle. When the workload of the edge node exceeds its capacity, the scheme iteratively selects the application with the longest response time. Second. if its response time exceeds the response time passed to the cloud computing center, it is assigned to the cloud computing center for processing; if not, it is reassigned to the standby edge to minimize its response time until all vehicle applications cannot find a better edge node. Then, computing resources are allocated to each edge node, and resources of different sizes are allocated to different types of VMs in each edge node through convex optimization. Finally, the extensive simulation results illustrate that the multipath transmission workload balancing optimization scheme can effectively reduce the average response time of the vehicular applications compared to existing schemes.

INDEX TERMS Vehicular heterogeneous network, mobile edge computing, multipath transmission, workload allocation.

I. INTRODUCTION

The Internet of Things (IoT) is currently an important development target of the new generation of information technology. It can interconnect a large variety of smart devices in a wide geographical area, and use the Internet and telecommunication networks as information carriers to enable these smart devices to exchange information. IoT has become part of the infrastructure for advanced applications in many smart cities and connected communities [1]. It has a quite wide variety of applications, many IoT applications represented by smart transportation, smart city, smart home and smart

The associate editor coordinating the review of this article and approving it for publication was Qing Yang.

health have been widely studied, which can effectively enrich people's life and improve the life quality of people.

Intelligent transportation is an important branch of IoT. The development of intelligent transportation has promoted the development of vehicular intelligent devices. Applications such as real-time video surveillance, safe driving, and audio-visual entertainment have become popular, which can prove the diversification trend of vehicular applications. Although the related technologies of smart transportation are developing rapidly, a major bottleneck of vehicular applications is the limited computing resources such as CPU and memory in each vehicular device, which will have a bad effect on resource-intensive applications involving big data analytics or real-time processing, and the proliferation of vehicular applications further exacerbates this situation [2]. In order to meet the explosive computing requirements of vehicular applications, cloud computing has become an important way to improve application performance [3].

Vehicular networks that support cloud computing allow applications to run on remote clouds by integrating communication and computing technologies, effectively reducing the local computing burden. However, when a vehicular user issues a service from an edge network located remote from the cloud through the core network, it may cause a long delay [2], which greatly affects QoE (Quality of Experience) of the delay-sensitive vehicular application, resulting in a poor user experience. Mobile Edge Computing (MEC) is an effective method to solve this problem. We can set up edge clouds near vehicle users and vehicular equipment to offload computing resources to edge nodes, reducing the traffic load on the core network and minimizing the response time of vehicular applications [5].

On the other hand, with the rapid development of new network access technologies, the traditional Internet of Vehicles (IoV) is developing towards the vehicular heterogeneous network [6]. Vehicles provide a variety of network interfaces, which are used to connect vehicles to a variety of different types of networks. These different networks assist each other to ensure the diversity of applications in the vehicle environment. The Multipath Transmission Control Protocol is a protocol proposed by the Internet Engineering Group (IETF) in 2009 [7]. It is an extension protocol of the Transmission Control Protocol, which allows multiple paths to be used for data transmission at the same time. MPTCP can aggregate bandwidth and increase transmission rate, especially for wireless environments.

The configuration of edge nodes can reduce the network delay caused by the distance to the cloud [8]. However, there are many vehicles in the current urban environment, especially in the morning and evening peak hours. In this case, it will still cause delay due to transmission congestion. Because of the mobility of the vehicles, the network location of the vehicle devices will change, and the edge nodes will also switch accordingly. It will close the connection between the existing vehicular device and the source edge node, and re-establishing the connection will inevitably lead to unnecessary network delay. In addition, there is a startup time before the vehicular device offloads the application to the edge node, which also affects Quality of Experience (QoE).

In order to solve the above problem, this paper uses multipath transmission to establish connection and perform VM migration from the source edge node to the destination edge node, reducing the startup time and network delay caused by reestablishing the connection. Multipath transmission refers to the application information sent by the source node is sent to the destination node along two or more paths. On the other hand, since the response time includes three aspects: network delay, computing delay and startup time, allocating the workload of all vehicular users to the nearest edge node simply cannot effectively reduce the response time. The workload allocation for different types of requests can greatly affect the response time of user requests. For each edge node, the resource allocation for different types of applications also affects the computing delay for different types of requests. Since the computing size of each request for different applications is different, the computing capacity of the edge nodes should be optimally allocated for different types of applications in order to reduce the computing delay of all vehicular applications.

Aiming at solving the problem that the long response time of vehicular applications causes poor service quality, we propose a multi-path transmission workload balancing optimization scheme based on mobile edge computing. The solution takes full account of network delay, computing delay, and startup time, dynamically allocating application requests based on the workload of the edge nodes to minimize the total response time of vehicular applications. The following are the main contributions of this paper:

1) Using multipath transmission instead of the original transmission protocol for communication. This can effectively reduce the network delay caused by re-establishing the connection. Multipath transmission can support real-time Virtual Machine (VM) dynamic migration between edge nodes without disrupting vehicular applications running on the VM. The original IP address changes after migration, but the multipath transmission connection with the edge node will still be established. This paper proposes that the VM uses two interfaces to configure the target IP address information before VM migration. This can achieve seamless migration of VMs and effectively reduce startup time.

2) We propose a multipath workload balancing optimization (MT-WBO) scheme. The scheme takes the network delay and computing delay into consideration, and divides the workload distribution problem of edge nodes into vehicular application allocation and computing resource allocation. Aiming at solving the problem of vehicular application allocation, a heuristic scheme is designed. According to the workload of the edge node and the network delay between the vehicular user and the edge node, different types of requests from each user are assigned to the proper edge nodes. For the problem of computing resource allocation, this paper regards

it as a convex optimization problem, and solves the convex optimization problem to optimally allocate the computing resources of each edge node to different types of VMs.

The rest of this paper is organized as follows. In section II, we briefly review the related work. In section III, we describe the edge node network architecture and system model, and introduce the method of VM migration using multipath transmission and propose the MT-WBO scheme to solve the vehicular application workload allocation of edge nodes. Section VI describes the simulation results and section VII summarizes the full paper.

II. RELATED WORKS

A large number of scholars and research institutions have studied MEC. Li *et al.* [9] proposed a platform, named WiCloud, to provide edge networking, proximate computing and data acquisition for innovative services. Wei *et al.* [10] described the offloading system model and presented an innovative architecture, called ''MVR'', contributing to computation offloading in MEC. Luo *et al.* [11] proposed an edge computing layered architecture that efficiently allocates large-capacity vehicular data. Wang *et al.* [2] proposed a collaborative vehicular edge computing framework to support scalable vehicular services and applications. Quang *et al.* [12] proposed an offloading decision scheme that analyzes the trade-offs in computing strategies to offload visual data processing from low to high loads. Reference [13] proposed a multi-access edge computing framework and corresponding communication protocol for content distribution and processing in vehicular networks. The proposed protocol used a cluster-based approach in which fuzzy-logic-based schemes were employed to select efficient gateway nodes that bridged licensed Sub-6 GHz communications and mm-Wave communications to maximize overall network throughput. Reference [8] designed an application-aware workload allocation scheme based on IoT edge computing, which fully considered the trade-off between network delay and computing delay, dynamically allocated application requests and computing resources to minimize the response time of IoT application requests. However, this study did not take into account the network delay caused by application migration between cloudlets during scheme iteration. Reference [14] introduced a software-defined vehicular edge computing (SD-VEC) architecture and designed a motion-aware greedy algorithm (MGA) to determine the amount of edge cloud resources allocated to each vehicle. Simulation results showed that the scheme could improve the success rate of task execution. In [4], a hierarchical cloudlet-based vehicular edge computing offloading framework has been proposed, and a multi-level offloading scheme based on Stackelberg Game theory has been designed to maximize the utility of vehicles and computing servers.

Literature [15] solved the problem of workload allocation in the fog network, and proposed a centralized fog network workload allocation framework, which is beneficial to reduce the computing delay of unrelated tasks. In [16], a task assignment scheme for delay and quality optimization in Vehicular Fog Computing (VFC) has been proposed. The authors of [17] studied the problem of Cloudlet placement and mobile user allocation in the Wireless Metropolitan Area Network (WMAN), and designed a workload allocation scheme. The scheme can place the Cloudlet in the user-intensive area of the WMAN and assign mobile users to the placed Cloudlet, and balance their workload to reduce the response time of the request. These schemes consider the environment where a single edge node provides services for users, or how to select multiple cloudlets. However, all of the above work does not consider the impact of user mobility on services that are already running in edge nodes in multiple edge node environments. The authors of [18] studied the VM layout and workload allocation of collaborative MEC system, and developed a Mixed Integer Linear Program model (MILP) to optimally place VM and allocate workload while meeting application latency. Numerical results show that remote VM placement and workload aggregation can optimize MEC application utilization. The authors of [19] designed a layered Cloudlet network and proposed a Workload Allocation (WALL) scheme, which can minimize the average response time of user by deciding which Cloudlet to assign the user application to and how much computing resources to provide.

However, all of the above work assume that the application is isomorphic, and then distribute the workload between the edge nodes to minimize the response time of the user request. In addition, they did not consider the impact of user mobility on services that are already running in edge nodes in multiple edge node environments. Therefore, it is necessary to propose a workload balancing optimization scheme for different types of applications to minimize their response time in mobile environment.

III. SYSTEM MODEL

Figure 1 shows the distributed edge node network architecture. Each edge node is located near the corresponding roadside unit and cooperates with the roadside unit. The cellular core network is deployed through Software Defined Network (SDN), which consists of an SDN controller and an OpenFlow switch, enabling flexible routing between roadside units. Applications for vehicular devices can be transferred via multipath transmission to the nearest edge node. When the vehicle changes its IP address or needs to dynamically change the edge node during the iterative process of workload allocation, MPTCP can be used to perform VM migration from the source edge node to the destination edge node to reduce startup delay and network delay caused by reestablishing connection.

For an edge node, we assume that each VM is only responsible for the workload of a vehicular application, i.e. each vehicular application is mapped to its corresponding dedicated VM. We allocate computing resources to different types of VM by the percentage of different types of workloads. The computing capacities of the VMs in the edge nodes is not

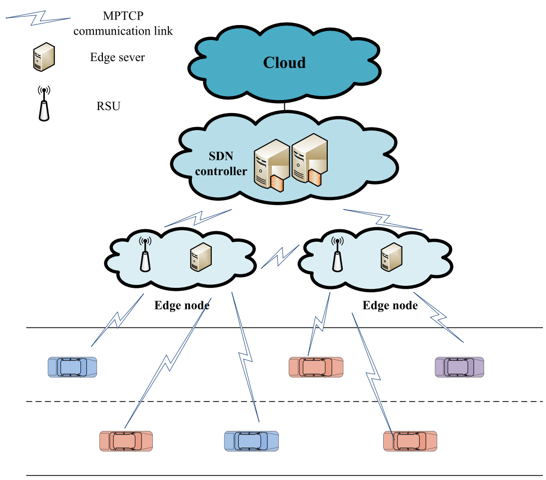


FIGURE 1. Distributed edge node network architecture.

equal, it can be dynamically changed according to the type of vehicle application, so each type of vehicular application can be offloaded to an edge node with a corresponding type of VM.

A. SYSTEM DELAY MODELING

As mentioned above, the response time of the vehicular application request consists of two parts: network delay and computing delay. Network delay is the time it takes for the vehicular application to send to the edge node and the processed application to return to the vehicle. The computing delay is the time it takes to process the vehicular application request. Communications and VM migration using multipath transmission can effectively reduce network delay and start-up delay.

Suppose the total number of request task application instances that can be processed simultaneously is:

$$N = \sum_{i=1}^{w} N_i \tag{1}$$

The total bandwidth B is equally divided into N request task application instances, so that the frequencies that each user can be occupied do not interfere with each other to simultaneously send their data to the edge node and the cloud computing center. Where,

$$r_{u,i} = \frac{B}{N}\log_2\left(1 + \frac{p_{u,i}h_i}{B/Nn_o}\right) \tag{2}$$

$$r_{d,i} = \frac{B}{N}\log_2\left(1 + \frac{p_{d,i}h_i}{B/Nn_o}\right) \tag{3}$$

The uplink and downlink transmission rates of the vehicle terminal $V_i \in \{V_1, V_2, \ldots, V_w\}$ are respectively indicated. Where n_0 is the noise power spectral density, h_i is the channel gain between the base station and the application N_i, then $p_{d,i}$ and $p_{u,i}$ are the downlink and uplink power of the vehicle respectively. The network delay d_{ik} between the vehicle and the edge node is:

$$d_{ik} = \frac{M_{ik}}{r_{u,i}} + \frac{N_{ik}}{r_{d,i}} \tag{4}$$

where M_{ik} is the initial packet size of the vehicle application k to the edge node i server, and N_{ik} is the packet size of the edge node i server returning to the vehicle application k.

Suppose there is a set of edge nodes in the system model, the number of which is i, denoted by $I = \{I_1, I_2, \ldots, I_i\}$. Each vehicle terminal has a limited number of applications, and the input application set of the jth vehicle is represented as $J_j = \{J_{i1}, J_{i2}, \ldots, J_{in_j}\}$. Assuming that k-type application requests of vehicle is generated by a Poisson process with an average arrival rate λ_{ik}, the workload of a VM of type k in edge node i is represented by:

$$\eta_{ik} = \sum_{j\in J} a_{ijk}\eta_{jk} \tag{5}$$

where a_{ijk} is the number of k-type application requests for vehicle j assigned to edge node i and η_{ik} is average request arrival rate for k-type applications in vehicle j.

On the other hand, the computing capacity (in CPU Hz) of the k-type of VM in the edge node i is fixed in each slot, and the computing size of the k-type of application request (in terms of CPU cycles) follow the exponential distribution with an average of l_k. Therefore, the serving time of the k-type request running in the VM of the edge node is l_k/γ_{ik}, which also follows the exponential distribution. Since the arrival rate of each VM of the edge node follows the Poisson process, and the corresponding serving time follows an exponential distribution, each VM of the edge node can form an M/M/1 queuing model to process its corresponding application request. In order to keep the queue stable, the average arrival rate of the VM (i.e. η_{jk}) should be less than its average service rate (i.e. γ_{ik}/l_k), then we have $\gamma_{ik}/l_k - \eta_{ik} > 0$. The computing delay for an k-type application request in the edge node is:

$$t_{ik} = \frac{1}{\gamma_{ik}/_{l_k} - \eta_{ik}} \tag{6}$$

Therefore, the response time S_{ik} of the vehicle application request is:

$$S_{ik} = d_{ij} + t_{ik} \tag{7}$$

B. VM MIGRATION OF EDGE NODES BASED ON MULTIPATH TRANSMISSION

As mentioned above, when the vehicle moves to a new network location resulting in a change in its IP address, the corresponding VM running the vehicular application should also migrate from the source edge node to the destination edge node. VM migration forces the IP address of VM to change, and further forces the VM to restart all established connections, which results in high delay and poor QoE. In addition, VM migration is also involved in the dynamic allocation of workload for vehicular applications. Therefore, we propose to use MPTCP for VM migration, which can effectively reduce the network delay caused by edge node switching.

We configure two virtual interfaces (interface A and interface B, respectively) for the VM in the edge node, one of which runs in the normal VM operation mode and the other that works after migrating the VM. The two interfaces cannot work at the same time. If interface A is in the startup state, interface B must be in the shutdown state. After the VM is migrated, the states of the two are exchanged. We always use the interface that is down as the interface used during VM migration. Once it is detected that the IP address of vehicle has changed, the VM knows that the corresponding vehicle has changed its position. At this point, the VM opens the closed interface ready for migration. The IP address of the interface is determined by the IP address of the vehicle with which it is matched.

We use a pre-established network configuration policy to build the migrated IP address of VM. We specify that the IP address of vehicle uses an even address, and the IP address of its corresponding VM is its address incremented by 1, which is an odd address. As mentioned above, after the vehicle changes position, the vehicle will be assigned a new even IP address. Once the VM is accessed using this new IP address, the VM knows that its new IP address of new location will be the next odd IP address.

The VM uses the ADD_ADDR22 option of MPTCP to inform the client that there is a new IP address, but the sub-interface cannot be started using the additional interface until migrating the VM. Since the IP address assigned to the additional interface comes from a different network, the VM cannot be accessed through this interface, but the vehicle-side connection will continue to try to create a new sub-flow using the new IP address of VM.

The VM will immediately notify the additional interface that it is available after the migration is complete. At this point, the interface that was previously working becomes inactive. Since the vehicle knows the new IP address of the VM that was notified before migration, the vehicle initiates the connection through the interface. Once a new sub-flow is started, the previous interface is removed to prepare for the next migration.

VM migration through multipath transmission not only avoids another startup time added by vehicular application offloading, but also reduces network delay caused by traditional VM migration.

IV. MULTIPATH TRANSMISSION WORKLOAD BALANCING OPTIMIZATION SCHEME

Since VM migration using multipath transmission has reduced the potential startup delay, and the startup delay is independent of the workload allocation, this paper only considers network delay and computing delay in workload allocation. Because the workload between different edge nodes is dynamically allocated, the overloaded edge node will generate a higher computing delay than other lightly loaded edge nodes. Thus if the edge node closest to the vehicle is overloaded, then each vehicle application requests should be allocated to alternate edge nodes to reduce computing delay. However, offloading the request of the vehicular application from its nearest edge node to other edge nodes will increase network delay.

Therefore, this paper proposes a modified multipath transmission workload balancing optimization scheme (MT-WBO) based on the system model for the workload allocation problem of vehicular applications. The proposed scheme minimizes the response time for all vehicular applications in the network by allocating the requests of each vehicular application between edge nodes and flexibly assigning the computing resources of each edge node to different types of VMs to serve the assigned vehicular applications.

This paper breaks down the problem into two subproblems: workload allocation problem and computing resource allocation problem. We first assign different types of vehicular applications in the edge nodes, i.e. determine a_{ijk}, and then allocate computing resources to each edge node optimally according to the different types of VMs.

A. WORKLOAD ALLOCATION

When assigning workloads to vehicle applications in edge nodes, we should prioritize assigning each vehicle application to its nearest edge node to reduce overall network latency. Therefore, this paper first assigns all vehicle applications to the nearest edge node to initiate workload balancing using multipath transport. Considering that the workload of the initial edge node is zero, the initial vehicle application allocation is determined by the network delay between the vehicle and the edge node. Then, the MT-WBO scheme will iteratively select the application with the longest response time. Once its response time exceeds the response time passed to the cloud computing center, it will be assigned to the cloud computing center for processing. If it is not exceeded, it will be reassign to the alternate edge. The node minimizes its response time until all vehicle applications cannot find a better edge node.

Since each application of the vehicle is allocated respectively in the edge node, we denote W_1 as a set of all vehicular applications waiting to be allocated between edge nodes, and I_1 denotes a set of edge nodes with redundant computing resources. At start time, all vehicular applications have not been assigned and they are placed in W_1, and at this time $W_1 = W$. All edge nodes are not assigned with any applications, so all edge nodes are placed in the I_1 set. Let d_{iw} denote the network delay between application w (i.e. $w \in W_1$) and the edge node i. j_w indicates the vehicle in which the application is located, then we have $d_{iw} = d_{ij_w}, \forall i \in I$, $\forall w \in W_1$.

In initialization, for the application w, the optimal edge node $i^* \in I_1$ causes the lowest network delay, i.e. $i^* = \arg\min \{d_{iw} | i \in I_1\}$. The sub-optimal edge node i' is the edge node that causes the second low network delay in the set I_1, i.e. $i' = \arg\min_i \{d_{iw} | i \in I_1 \backslash i^*\}$.

The capacity of each edge node is limited, so it is not possible to assign all vehicular applications to their corresponding optimal edge nodes. The basic idea of initialization is to iteratively select the appropriate application. If the suboptimal edge node causes a much more severe network delay than the optimal edge node, then the application is assigned to its best edge node. It can be seen that the network delay of the sub-optimal edge node determines the priority of assigning the application to its best edge node. For example, if the sub-optimal edge node B of the application results in significantly higher delay than its best edge node A compared to other vehicular applications, then assigning the application to the suboptimal edge node will significantly affect the total network delay for all applications. In this case, the application has a higher priority than other applications that are assigned to its best edge node A.

Where θ_{d_w} represents the difference between the network delay from application w to the best edge node i^* and the network delay from w to the sub-optimal edge node, i.e. , where θ_{d_w} as a percentage represents the ratio of the network delay from application w to the best edge node i^* minus the network delay from the application w to the sub-optimal edge node $i^{'}$ to the network delay from the application w to the best edge node i^*.

$$\theta_{d_w} = \frac{d_{i'w} - d_{i^*w}}{d_{i^*w}}, \quad \forall w \in W_1 \tag{8}$$

Thus, as shown in Algorithm 1, the scheme will select and assign the appropriate application with the largest network delay difference (i.e. $w = \arg\max\{\theta_{d_w} | w \in W_1\}$) in each iteration of initialization. Then, if the workload of edge node exceeds its capacity, the edge node will be removed from I_1. Each time the set is updated, the algorithm recalculate i^*, $i^{'}$, and θ_{d_w} for each application w in set W_1. Repeat the above process until all the vehicular applications are allocated in the edge node, i.e. $W_1 = \varnothing$.

Algorithm 1 The Initial Distribution of the Workload

Input: d_{ij}, average task arrival rate vector for vehicular applications $\Psi = \{\eta_{ik} | j \in J, j \in K_j\}$
Output: Initial application allocation matrix $A = \{a_{ijk} | i \in I\ j \in J, k \in K_j\}$
Let $W_1 = W, I_1 = I$
$\forall W_1 = W$, calculate θ_{d_w}
For $W_1 \neq \varnothing$ **do**
  Search for application w, $w = \arg\max\{\theta_{d_w} | w \in W_1\}$
  Assign an application w to its nearest edge node i^*, i.e. $i^* = \arg\min_i \{d_{ij}\}$
  Let $a_{ij_wk_w} = 1$
  Update application set W_1
  if edge node i^* is filled, **then**
    Remove i^* from I_1;
    $\forall W_1 = W$, recalculate θ_{d_w}.
  End if
End for
Return A

After the initial distribution of the workload, as shown in Algorithm 2, the MT-WBO scheme iteratively selects the application with the longest response time. If its response time exceeds the response time passed to the cloud computing center, it is assigned to the cloud computing center for processing, and if not, it is reassigned to the standby edge to minimize its response time until all vehicle applications cannot find a better edge node. Initially, all vehicular applications are unmarked and we define W_2 as unmarked applications set. Then, in each iteration, the MT-WBO finds the application with the longest response time in all unmarked applications and searches for new edge nodes for the application to minimize its response time. In each iteration, the computing resources of each application in the edge node are determined by the percentage of the application workload of the total edge node workload, so we can deduce the response time of the application in different edge nodes. If a new edge node is found, the MT-WBO goes to the next iteration. Otherwise, the scheme marks the application (i.e. removes the application from W_2) and proceeds to the next iteration. The MT-WBO repeats iterations until $W_2 = \varnothing$. Suppose the response time that the application w passes to the cloud computing center is C_w.

Algorithm 2 Workload Allocation

Initialize the workload assignment according to Algorithm 1 and the obtained A
Let $W_2 = \{w | w \in W\}$
For $W_2 \neq \varnothing$ **do**
  Find the application with the highest response time $w \in W_2$
  Get the current edge node i of application w
  if Application w has a response time greater than C_w, **then**
  Find the proper edge node i^* for the application w, i.e. $i^* = \arg\min(S_{ik})$;
  if $i^* \neq i$, **then**
    Assign the application w to the new edge node i^*, and it uses MPTCP for VM migration, then update A
  else
    Mark the application w;
  end if
  else
    Transfer the app w to the cloud and mark the application w
  end if
end for
Return A

B. COMPUTING RESOURCE ALLOCATION

After assigning all vehicular applications to different edge nodes, we continue to allocate computing resources to each edge node. Then with the constraints, the computing resource allocation problem is:

$$\begin{aligned}
&\min_{\gamma_{ik}} \sum_{j\in J}\sum_{k\in K} a_{ijk} S_{ik} \\
&s.t. \sum_{k\in K} \gamma_{ik} \le H_i \\
&\quad \gamma_{ik}/l_k - \sum_{j\in J} a_{ijk}\eta_{jk} > 0
\end{aligned} \tag{9}$$

where H_i represents the computing power of the edge node. When each a_{ijk} is determined, this constraint problem is a convex optimization problem, where:

$$f = \sum_{j\in J}\sum_{k\in K} a_{ijk} S_{ik} \tag{10}$$

$$\frac{\partial^2 f}{\partial\gamma_{ik}\partial\gamma_{ik'}} = \begin{cases} \sum_{j\in J} 2a_{ijk} l_k^{-2}\left(\frac{1}{t_{ik}}\right)^{-3}, & if\ k = k' \\ 0, & otherwise \end{cases} \tag{11}$$

Since $1/t_{ik} > 0$, the Hessian matrix of $fH = \left[\partial^2 f / \left(\partial\eta_k \partial\eta_{k'}\right)\right]$ is a positive definite matrix, and the function f is a convex function. Furthermore, since the constraints are linear, the optimization problem is a convex optimization problem. For the convex optimization problem, we can derive its optimal solution by solving its KKT condition. Therefore, the computing resources of each edge node are optimally allocated to different VMs to minimize the response time.

V. SIMULATION AND ANALYSIS

In this simulation, 1000 vehicles are evenly allocated within the range of 30 square kilometer, and there are 10 different types of vehicular applications. We randomly select four types of applications for each vehicle, i.e. the total number of applications in the network is 4000. The length of each time slot is set to 3 minutes. Since the task arrival rate of each application follows the Poisson distribution, we randomly select the average task arrival rate for each application between 0 and $\lambda_{\max}$. Since the computing size of the k-type of application request follows the exponential distribution of the mean l_k, the average size of requests for different types of applications is selected based on a normal distribution with a mean of 10^6 and a variance of $2*10^5$. The maximum allowed computing delay for different types of applications is selected according to N (65, 20) (ms). We compare it with existing workload balancing scheme of MEC AREA and the DBC scheme proposed in [17].

Figures 2 and 3 show the average delay for each application to the vehicle under different schemes and the average response time for different types of applications under each scheme. As can be seen from the figures, the MT-WBO scheme achieves a lower response time than the other three schemes, which is 88ms less than the conventional load distribution scheme. In particular, the conventional scheme always assigns application requests to the nearest edge node, regardless of the workload in each edge node, so it produces the lowest network delay. However, the edge node is prone

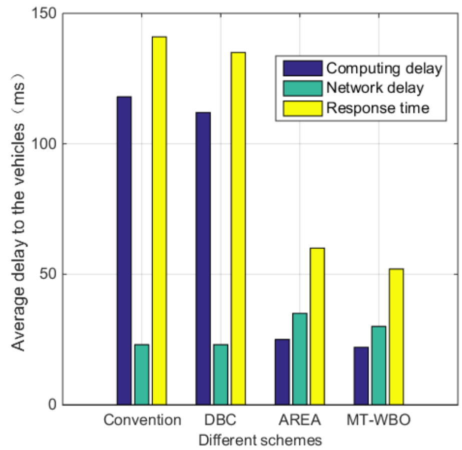


FIGURE 2. Average performance of each application for different schemes.

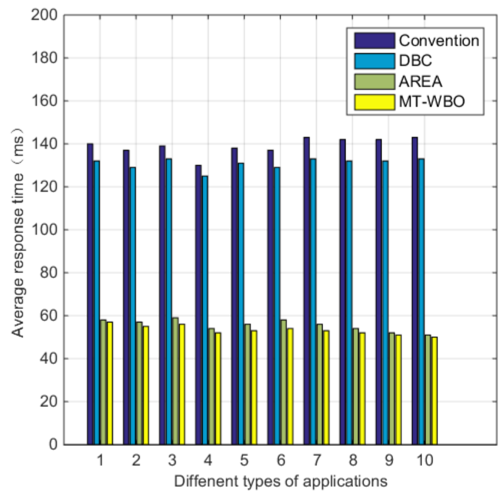


FIGURE 3. Average response time for different types of applications.

to overload, resulting in increased computing delay and the longest response time. The response time is the longest. The DBC scheme uses a density-based strategy to assign applications to the nearest edge node until the workload of each edge node exceeds the average workload in the edge node, but does not consider the diversity of applications in each edge node. Both the conventional scheme and the DBC scheme result in lower network latency and higher computational latency than the MT-WBO scheme. The AREA scheme tends to allocate applications with smaller computational sizes to lighter edge nodes, so its computational latency has been significantly reduced. However, because it still uses conventional data transmission, it is not suitable for in-vehicle application-intensive situations, so the network delay is relatively large. The MT-WBO scheme takes into account the two factors of network delay and computation delay, considers that different types of applications in each edge node correspond to different workloads, and has a more reasonable application priority decision strategy, according

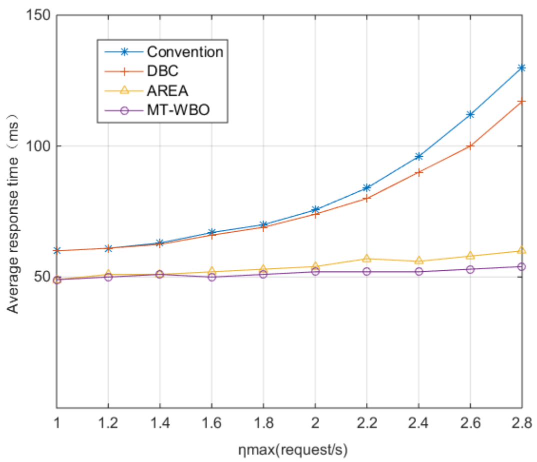


FIGURE 4. Average response time as λ_{max} changes.

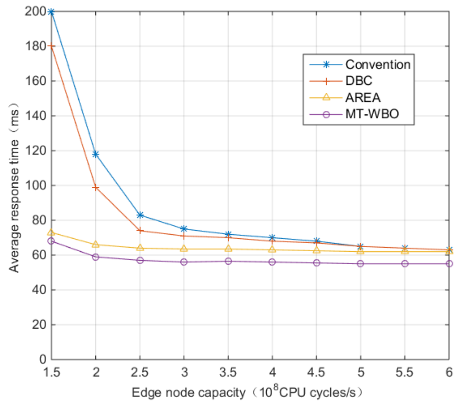


FIGURE 5. Average response time corresponding to the capacity change of each edge node.

to the corresponding workload. Different types of VMs optimize allocation of computing resources, so their computational latency is also significantly reduced. At the same time, the MT-WBO scheme uses MPTCP for task offloading and VM migration, so its network latency is not significantly increased, and the average response time of each application is significantly shortened.

We further analyze how the workloads of application influence the performance of the three schemes. The value of λ_{max} reflects the workload of application, i.e. increasing λ_{max} will increase the workload of application. As shown in Figure 4, as λ_{max} increases, the average response time of the three schemes increases gradually. However, compared with the other two schemes, MT-WBO has a relatively low average response time and a slow growth rate. When the workload of the application is heavy, MT-WBO can always offload the application with the longest response time to the alternate edge node through multipath transmission, thereby iteratively minimizing the maximum response time of the application. Otherwise, MT-WBO also optimally allocates the computing resources of each edge node to different types of applications based on its workload and its corresponding computing size, thereby further reducing the computing delay.

In addition, we also verified the effect of edge node capacity on the average response time. Figure 5 shows the response times of the four schemes as the edge node capacity increases. It can be seen from the figure that when the capacity of the edge node is less than 2.5*108 CPU rev / sec, the average response time of the MT-WBO scheme and the AREA scheme is much shorter than the conventional scheme and the DBC scheme. This is because when the capacity of the edge nodes is small, the conventional scheme and the DBC scheme cannot balance the workload between edge nodes of different type application requests, resulting in high computing delay. The MT-WBO scheme and the AREA scheme can minimize the computing delay, and the MT-WBO scheme can also reduce unnecessary network delays, so the response time is the shortest. However, when the capacity of the edge node is very high, the computing delay is no longer the dominant factor of the average response time, so the average response time of the conventional AREA scheme and that of the DBC scheme are close to the response time of the MT-WBO scheme.

VI. CONCLUSION

This paper proposes a multipath transmission workload balancing optimization scheme based on MEC. This scheme is applied in the edge computing architecture, using a combination of edge nodes and cloud computing data centers. First of all, considering the response time of different types of vehicles, dynamically assigning different types of application requests to each edge node, and then optimally allocating the computing resources of each edge node to the corresponding VM to reduce the average response time of the vehicle application. Simulation experiments show that the scheme has performance gain compared with other schemes in reducing response delay. In future research, we plan to perform a corresponding proportion of load distribution on the load of nearby edge nodes and surrounding vehicles, resulting in higher resource utilization and less delay.

REFERENCES

[1] M. M. Rathore, A. Ahmad, A. Paul, and S. Rho, "Urban planning and building smart cities based on the Internet of Things using big data analytics," *Comput. Netw.*, vol. 101, pp. 63–80, Jun. 2016.

[2] K. Zhang, Y. Mao, and S. Leng, "Optimal delay constrained offloading for vehicular edge computing networks," in *Proc. IEEE Int. Conf. Commun.*, vol. 98, Jul. 2017, pp. 93–100.

[3] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive offloading," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 36–44, Jun. 2017.

[4] K. Wang, H. Yin, W. Quan, and G. Min, "Enabling collaborative edge computing for software defined vehicular networks," *IEEE Netw.*, vol. 32, no. 5, pp. 112–117, Sep. 2018.

[5] X. Sun and N. Ansari, "Latency aware workload offloading in the cloudlet network," *IEEE Commun. Lett.*, vol. 21, no. 7, pp. 1481–1484, Jul. 2017.

[6] A. Mahmood, B. Butler, and B. Jennings, "Towards efficient network resource management in SDN-based heterogeneous vehicular networks," in *Proc. IEEE Comput. Softw. Appl. Conf.*, vol. 23, Jul. 2018, pp. 3–8.

[7] J. Kim, B. H. Oh, and J. Lee, "Receive buffer based path management for MPTCP in heterogeneous networks," *IEEE Integr. Netw. Service Manage.*, vol. 9, no. 12, pp. 13–18, Jan. 2017.

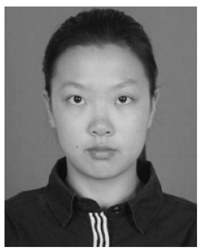
[8] Q. Fan and N. Ansari, "Application aware workload allocation for edge computing-based IoT," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2146–2153, Jun. 2018.
[9] H. Li, "Mobile edge computing: Progress and challenges," in *Proc. 4th IEEE Int. Conf. Mobile Cloud Comput., Services, Eng.*, vol. 13, Oct. 2016, pp. 147–153.
[10] X. Wei, S. Wang, and A. Zhou, "MVR: An architecture for computation offloading in mobile edge computing," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, vol. 27, Sep. 2017, pp. 89–95.
[11] G. Luo, "Cooperative vehicular content distribution in edge computing assisted 5G-VANET," *China Commun.*, vol. 15, no. 7, pp. 1–7, Jul. 2018.
[12] H. Trinh, "Energy-aware mobile edge computing and routing for low-latency visual data processing," *IEEE Trans. Multimedia*, vol. 20, no. 10, pp. 2562–2577, Oct. 2018.
[13] Q. Hu, "Vehicular multi-access edge computing with licensed Sub-6 GHz, IEEE 802.11p and mm-wave," *IEEE Access*, vol. 6, pp. 1995–2004, 2017.
[14] C. F. Liu, M. Bennis, and H. V. Poor, "Latency and reliability-aware task offloading and resource allocation for mobile edge computing," in *Proc. IEEE Globecom Workshops*, Jan. 2017, pp. 21–26.
[15] B. Amit and S. C. Mohd, "Centralized framework for workload distribution in fog computing," in *Proc. IEEE Int. Conf. Converg. Technol.*, vol. 98, Feb. 2018, pp. 1–9.
[16] C. Zhu, "Folo: Latency and quality optimized task allocation in vehicular fog computing," *IEEE Internet Things J.*, to be published.
[17] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Trans. Cloud Comput.*, vol. 5, no. 4, pp. 725–737, Oct. 2017.
[18] W. Wang, Y. Zhao, and M. Tornatore, "Virtual machine placement and workload assignment for mobile edge computing," in *Proc. IEEE Int. Conf. Cloud Netw.*, Dec. 2017, pp. 95–101.
[19] Q. Fan and N. Ansari, "Workload allocation in hierarchical cloudlet networks," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 823–7820, Apr. 2018.

ZHAO HAITAO received the M.S. and Ph.D. degrees (Hons.) in signal and information processing from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2008 and 2011, respectively, where he is currently an Associate Professor. His current research interests include wireless multimedia modeling, capacity prediction, and wireless network coding.

DING YI received the bachelor's degree from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2017, where he went on further education, majoring in communication and information system.

ZHANG MENGKANG received the bachelor's degree from the Ocean University of China, Qingdao, China, in 2016. He went on further education at the Nanjing University of Posts and Telecommunications, majoring in electronic and communications engineering.

WANG QIN received the B.S. degree in communications engineering and the Ph.D. degree in communication and information system from the Nanjing University of Posts and Telecommunications (NJUPT), China, in 2011 and 2016, respectively, where she is currently doing the Postdoctoral Research in electronic science and technology. From 2015 to 2016, she was a Visiting Scholar with the Department of Computer Science, San Diego State University, USA. She is also an Assistant Professor with the College of Engineering and Computing Sciences, New York Institute of Technology (NYIT), Nanjing Campus. Her research interests include multimedia communications, multimedia pricing, resource allocation in 5G, and the Internet of Things. She serves as a Co-Editor of the *IoT Journal* (Elsevier), the Track Co-Chair of MIPR 2019, a Technical Program Committee Member for many international conferences, such as ICC 2018, ICNC 2018, and CCNC 2017, and a Reviewer for many journals, such as the IEEE Transactions on Wireless Communications and the IEEE Transactions on Vehicular Technology.



SHI XINYUE is currently pursuing the degree in communication engineering with the Nanjing University of Posts and Telecommunications.

ZHU HONGBO received the B.S. degree from the School of Telecommunication Engineering, Nanjing University of Posts and Telecommunications (NUPT), Nanjing, China, in 1982, and the Ph.D. degree from the School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, in 1996. He is currently the Vice President of NUPT, where he is also a Professor and a Supervisor of Doctoral degree candidate in communications networks. His current research interests include the Internet of Things and next generation network.