Received March 12, 2020, accepted March 20, 2020, date of publication March 26, 2020, date of current version April 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2983440

A Delay-Tolerant Data Transmission Scheme for Internet of Vehicles Based on Software Defined Cloud-Fog Networks

BIN XIA[1], FANYU KONG[2], JUN ZHOU[3], XIAOSONG TANG[4], AND HONG GONG[1]

[1]Department of Management, Chengyi University College, Jimei University, Xiamen 361021, China
[2]Chongqing Engineering Technology Research Center for Development Information Management, Chongqing Technology and Business University, Chongqing 400067, China
[3]Chongqing Business Vocational College, Chongqing 401331, China
[4]Department of Rial and Civil Engineering, Chongqing Vocational College of Public Transportation, Chongqing 402247, China

Corresponding author: Fanyu Kong (cqkfy2002@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 71702015, in part by the China Postdoctoral Science Foundation under Grant 2017M611810, in part by the Fundamental Research Funds for the Central Universities under Grant 2018CDQYJZ0032, in part by the Social Science Planning Major Application Project in Chongqing under Grant 2017ZDYY51, in part by the Chongqing Engineering Technology Research Center for Development Information Management Open Foundation under Grant gczxkf201706, and in part by the Research Platform Open Project in Chongqing Technology and Business University (CTBU) under Grant 1456041, Grant KFJJ2017058, and Grant KFJJ2017061.

ABSTRACT The low-latency advantages of fog computing can be applied to solve high transmission latency problems of many network architectures in Internet of Vehicles. Therefore, this paper studies the application of fog computing in Internet of Vehicles. Considering that the fog network equipment deployed in Internet of Vehicles is relatively scattered, a new network architecture is proposed, which integrates cloud computing, fog computing and software defined network and other technologies. The proposed framework uses software defined network to centrally control fog network and obtains equipment performance of fog network. Furthermore, the optimal load balancing strategy is developed by communication overhead and other information. Based on time delay modeling of fog network, we study the time delay modeling of cloud-fog network and the energy consumption modeling of fog network. In addition, this paper models the selection process of data transmission network and data calculation execution server of delay-tolerant data as a partially observable Markov decision process optimization strategy in software defined Internet of Vehicles. By observing the state of system, current storage makes optimal decisions on data transmission and selection of computing nodes, thereby minimizing system overhead. Simulation results show that the proposed scheme can effectively reduce transmission delay and system overhead, improve data calculation efficiency.

INDEX TERMS Delay-tolerant data transmission, software defined network, fog computing, the Internet of Vehicles, load balancing strategy, partially observable Markov decision process.

I. INTRODUCTION

With the increasing number of Internet of Vehicles (IoV) users, it is necessary to build a powerful data center that supports IoV application services. At present, cloud computing technologies are mainly used to provide services for Internet of Vehicles. Cloud computing is defined as a computing paradigm in which computing resources are provided on pay-as-you-go basis and users can access the Internet anytime, anywhere. In recent years, cloud computing technologies are applied for data storage, data processing and data analysis in IoV. At the same time, some applications of IoV are deployed to cloud computing data centers to provide users with related services [1]–[3]. However, the explosive growth in the number of mobile terminals such as vehicles increases the burden on cloud servers in IoV. Besides, cloud computing data centers are far away from end users, which results in higher service processing delays. For latency-sensitive applications in IoV, this is an urgent issue [4], [5]. For example, the ambulance needs to acquire surrounding traffic condition

The associate editor coordinating the review of this manuscript and approving it for publication was Honghao Gao.

information in real time to assist driving, so that it can reach the rescue site timely in case of traffic congestion. And in order to ensure the safe driving of cars, real-time information of collision warning is needed to assist driving necessarily.

In order to reduce the burden of cloud server and the task processing delay, this paper proposes a new network architecture, which integrates cloud computing, fog computing, Software Defined Network (SDN) and other technologies. As an emerging network paradigm, SDN is one of the most popular research fields in IT. The characteristic of SDN is to control networks in a systematic, centralized and programmable manner by decoupling data planes and control planes. This feature makes SDN an important technology to solve the problems of difficult expansion and control of IoV architectures. Fog computing extends computing from cloud data centers to the edge of network. It provides computing, storage and network services between terminal equipment and cloud data centers. Some experts say that fog computing is a cloud closer to end users, which provides computing and services with low latency. However, fog computing networks usually consist of multiple network equipment with weak computing power. It is difficult for a single fog equipment to process large amounts of data efficiently [6], [7]. Thus, it is necessary to form a fog computing network for distributed computing using multiple fog equipment, and to balance network loads and reduce latency by load balancing strategies.

In recent years, with the development of cloud computing, academia has begun to study load balancing strategies with cloud computing. Although fog computing is called a cloud computing supplement, it is a cloud close to the ground. However, due to the heterogeneity of fog computing network, the load balancing strategy of cloud computing cannot be directly applied to fog computing. Although the cloud computing center is often far away from users, it can also be used as a distributed computing node with strong computing capacity and high transmission delay to handle some tasks. Therefore, this paper integrates cloud computing, SDN and fog computing to form a software defined cloud-fog network architecture for IoV. Moreover, in order to save energy consumption and reduce delay of business processing effectively, this paper studies delay optimization model under the constraints of energy consumption in fog network, and proposes a strategy of compromise between delay and energy consumption. The main contributions of this paper are summarized as follows:

1) A new network architecture is proposed, which integrates cloud computing, fog computing, SDN and other technologies. The proposed architecture uses SDN to centrally control fog network, obtains the equipment performance, communication overhead and other information of fog network to formulate the optimal load balancing strategy.
2) Considering the selection process of delay-tolerant data transmission networks and data calculation execution servers in software defined IoV, we model them as Partially Observable Markov Decision Process (POMDP) optimization strategy, which optimizes and reduces business processing delay effectively.

II. RELATED WORKS

After the concept of fog computing was proposed, many scholars began to study fog computing architectures. They merged the fog computing architecture with some existing network architectures and used the advantages of fog computing to make up for the shortcomings of existing network architectures. Ref [7] proposed a more concise fog computing architecture and introduced the application architecture of fog computing in smart grids. Ref [8] proposed an IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing, it can schedule and manage resources efficiently and in real time according to specific situation. Ref [9] combined SDN with fog computing and applied it to Vehicular Ad hoc Networks (VANETs), then proposed VANET architecture that combines fog computing with SDN. The architecture utilized SDN to increase flexibility, scalability, programmability and global awareness. Besides, it provided delay-sensitive and location-aware services for VANET by fog computing that satisfies the needs of VANETs solutions in the future. Fog computing complements cloud computing, the combination of cloud and fog network can provide users with better services. Ref [10] proposed a cloud-fog combination architecture to provide users with high-quality cloud gaming experience, which uses equipment around the user, such as idle computers, as fog computing nodes. And it rendered the game screen and send it to the user, the cloud server performed intensive game state calculations and send updated information to fog nodes. Experiments verified that this architecture can more effectively reduce game response delay and network bandwidth consumption compared to cloud computing or local cloud, thereby reducing game operating costs.

In order to study the fog network deeply, many scholars modeled the fog network and proposed feasible optimization strategies. Ref [11] introduced fog computing architecture into 5G cellular networks to reduce the latency of 5G networks. In order to meet the requirements of delay-sensitive applications, they studied the modeling of minimizing blocking probability under total delay constraints. Besides, they considered choosing three strategies such as random strategy, lowest latency strategy, maximum available capacity strategy to distribute load, and compared with the three strategies. Ref [12] proposed a software-defined embedded system supported by fog computing for traditional embedded systems that are limited by functionality, flexibility and scalability. The system performs modeling analysis on task execution time, including calculation time, I/O time and transmission time. Thus, a low-complexity three-phase algorithm to minimize service response time is proposed. In addition, energy consumption, as an important parameter of fog networks, is widely concerned and studied. Ref [13] theoretically modeled the fog computing architecture and studied the mathematical modeling of service delay and energy consumption of fog computing and traditional cloud computing architectures. In the IoT application scenario, they compared the fog computing architecture with the traditional cloud computing architecture in terms of service latency and energy consumption. It is clarified that fog computing has advantages over cloud computing in both reducing service latency and saving energy consumption. Ref [14] studied the cloud-fog combination architecture and divided cloud-fog system into four subsystems. It modeled the delay and energy consumption of subsystem and the entire cloud and fog system respectively, and studied the trade-off between energy consumption and delay. Under the constraints of delay constraints, the load distribution problem that achieves optimization goal of minimizing system energy consumption is analyzed, and the corresponding optimization problem is solved using corresponding algorithms.

At present, user requests are increasing in the Internet, especially in IoT. For example, there are many user requests at the same time and processing too many applications will generate huge energy consumption in the IoV. Therefore, in order to save energy consumption and minimize delay, it is particularly important to study the problem of delay optimization under the constraint of energy consumption. Ref [15] introduced two new wireless communication access protocols for small data packet transmission in IoT networks. As a result, higher data transmission throughput and lower latency performance are obtained. Ref [16] proposed a small data transmission scheme in a Narrow Band (NB) IoT system. It enables equipment in idle state to transmit small data packets without the network resources controlling connection establishment process. Ref [17] studied the coexistence and coordination of Internet of Everything (IoE) fog computing and cloud computing by jointly optimizing offloading decisions, calculating resource allocation and transmitting power. They proposed an Energy-efficient Computation Offloading and Resource Allocation (ECORA) solution in order to minimize system costs. Thus, compared with existing scheme, this scheme can reduce the system cost effectively [18].

However, despite lots of research work on data transmission and data calculation of IoT networks, there are still two problems that can be easily ignored in current research. On the one hand, data transmission on the current network can be divided into delay-sensitive data transmission and delay-tolerant data transmission. For delay-tolerable data transmission, a certain increase in transmission delay does not seriously affect the transmission performance of such data. However, if delay-sensitive data and delay-tolerant data are not distinguished during transmission, it will inevitably cause an increase in network resource load and resource waste. On the other hand, traditional data computing generally adopts cloud computing mode. However, the deployment of data storage and cloud computing servers is often far away. Frequent data transmission also causes excessive network load and unnecessary waste of network resources.

III. SYSTEM MODEL

A. SOFTWARE DEFINED CLOUD-FOG NETWORK ARCHITECTURE

IoV consists of vehicles, roadside infrastructure, and provides vehicle-to-vehicle, vehicle-to-infrastructure and vehicle-to-base station communications. Vehicles are usually moving at high speed in IoV. This means that vehicles have higher requirements for service processing time, especially for delay-sensitive services, such as, the measurement of the distance between vehicles, ambulance real-time monitoring of road conditions and the query of route maps. Although traditional cloud computing architectures can support these services, the cloud center server is far from vehicle terminals, the service processing delay is high. In order to solve these problems, we introduced fog computing into the cloud-based IoV architecture to satisfy the requirements of low latency. In addition, flexible centralized control using SDN controllers is also essential. Because SDN controller can obtain the global load information of all fog equipment and cloud center servers as well as equipment performance and communication parameters [19]. The proposed network architecture is shown in Figure 1.

The network architecture is divided into four layers: cloud computing layer, SDN control layer, fog computing layer and user layer.

The user layer consists of vehicles equipped with On Board Unit (OBU). The OBU includes processing unit, sensors, positioning system (for example, Global Positioning System (GPS)), radio transceiver that transmits Wireless Access in Vehicular Environments (WAVE) signals, and Wi MAX / 3G / 4G LTE radio signals that communicate with base stations transceiver.

The fog computing layer consists of Base Station (BS) and Road Side Unit (RSU) with computing and storage capabilities. In SDCFN architecture, both BSs and RSUs act as fog computing nodes and run the Open Flow protocol to communicate with SDN controller. The fog node can not only obtain required data and services from cloud server by active caching. Moreover, it can obtain traffic condition information from its neighboring fog nodes and store application-oriented information transmitted through it. In addition, the fog node regularly uploads the running status of collected vehicles and other dynamic information to the cloud for global information sharing. However, because massive connected vehicles generate a large number of processing services, it is difficult for a single BS or RSU processing task to satisfy the requirements of low latency. Thus, it is necessary to perform distributed computing and apply load balancing strategy to balance load and reduce latency.

The SDN control layer consists of SDN controllers. In this architecture, SDN controllers, fog nodes and cloud computing centers send flow tables to fog network through the Open Flow protocol. It sets the data forwarding rules and controls fog network centrally and globally. The SDN controller can collect global information of cloud-fog network, including

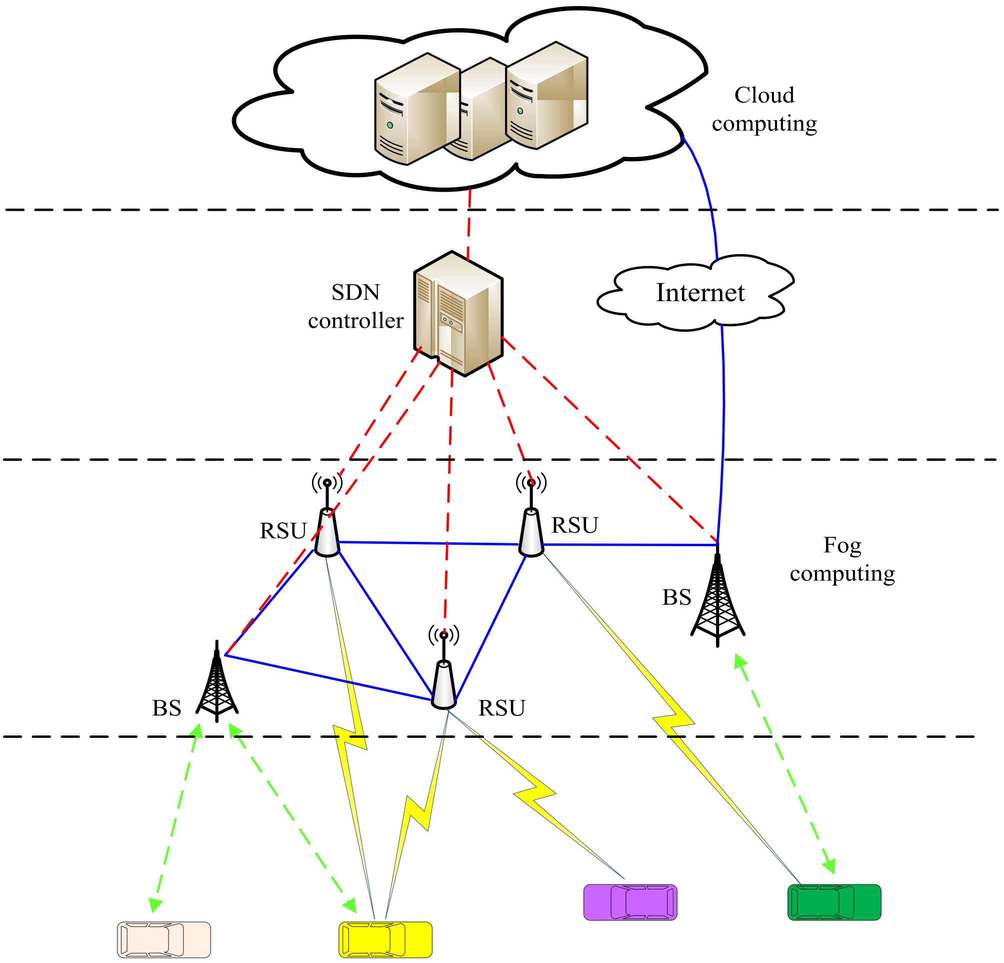

FIGURE 1. Proposed software defined cloud-fog network architecture.

the load, the processing speed of equipment and the communication delay. Therefore, SDN controller can formulate the optimal load balancing strategy for cloud-fog network. At the same time, SDN controller provides an open programming interface to support network routing and resource management, which provides users with software programming access to control cloud-fog network.

The cloud computing layer consists of high-performance server clusters. It stores and analyzes data and service requests from terminal devices, and provides various comprehensive services. In addition, this paper proposes to treat cloud computing as a distributed computing node as a whole, and then forms a hybrid cloud-fog distributed computing network.

B. THEORETICAL MODEL

In IoV, the BS and RSU that make up fog computing network are scattered and have weak computing power. Therefore, how to allocate computing tasks according to the performance and communication overhead of equipment, balance the load of computing nodes in cloud-fog network and reduce the delay of business processing become particularly important, especially for delay-sensitive services in IoV. This paper studies the time delay modeling of cloud-fog networks. Considering that there are many types of equipment in IoV, the amount of requested data is large and the equipment consumes more energy. Besides, the reduction of time delay will lead to increased energy consumption. Thus, the time delay modeling under energy consumption constraints of fog network is studied to achieve a compromise between delay and energy consumption in this paper.

This paper considers the software defined cloud-fog network consisting of k fog computing equipment in IoV scenario, its network equipment topology is shown in Figure 2.

Based on graph theory, this paper abstracts network nodes in network topology Figure 2 into a weighted undirected graph $G = (V, E)$, as shown in Figure 3. Specifically, $V = \{z_1, z_2, \cdots z_i, \cdots z_k, S, C\}$ is the vertex set, vertex $z_i \in V$ is the fog computing equipment, k is the number of fog computing equipment. S indicates the SDN controller and C indicates the cloud server cluster. $E = \left\{e_{z_1,z_2}, \cdots, e_{z_i,z_j}, \cdots e_{z_{k-1},z_k}, e_{z_3,c}, e_{z_{3,c}}\right\}$ is the edge set, edge

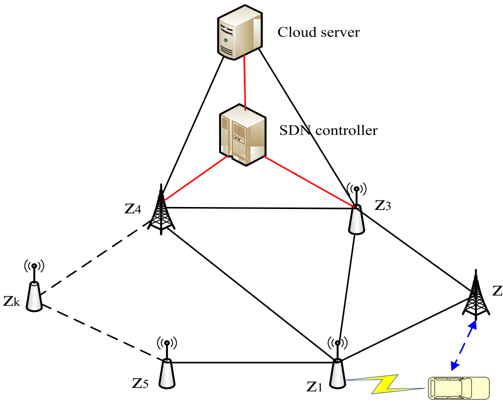

FIGURE 2. **Topology diagram of computing equipment with software defined cloud-fog network architecture.**

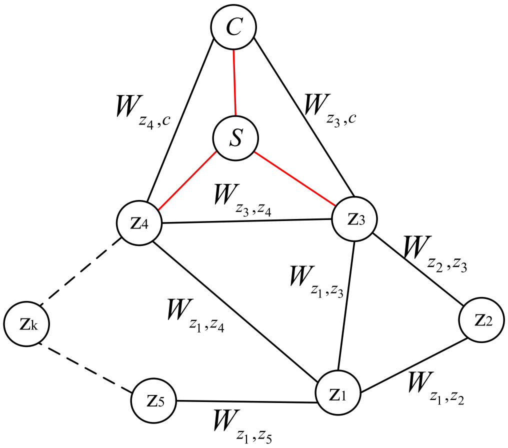

FIGURE 3. **Undirected diagram of software defined cloud-fog network architecture.**

e_{z_i,z_j} represents the communication link between fog computing nodes z_i and z_j. The weight τ_{z_i,z_j} on edges represents the communication delay between fog computing nodes $\{z_i, z_j\}$.

The computing power of each fog computing node z_i in Fig. 3 is assumed to be λ_{z_i}. In order to make full use of the high computing performance of cloud computing center to improve the overall business processing performance of cloud fog network, this paper considers the entire cloud computing center as a distributed computing node. Its computing power is assumed to be λ_c.

(1) Delay model of software defined cloud-fog network architecture

During task execution, users submit the request of task D to fog computing equipment connected to it every time. And it divides received user request task into several subtasks that are satisfied, and assigns it to cloud computing nodes including itself for processing. Therefore, the total delay processed by entire computing task in SDCFN can be expressed as:

$$t(\delta_i, \delta_c) = \max\left\{\frac{\delta_i D}{\lambda_{z_i}} + W_{z_i,z_j}, \frac{\delta_c D}{\lambda_c} + W_{z_i,c}\right\}$$

$$\times i, j = 1, 2, \cdots k \tag{1}$$

In the formula: $\delta_i D / \lambda_{z_i}$ represents the calculation delay of computing equipment z_i processing subtask d_i. W_{z_i,z_j} and $W_{z_i,c}$ are the communication overhead between fog equipment $\{z_i, z_j\}$ and the communication overhead from fog equipment z_i to cloud center C. Similarly, $\delta_c D / \lambda_c$ represents the computation delay of cloud computing center C processing sub-task d_c. The communication delay is modeled considering only transmission delay and the propagation delays are collectively referred to as other delays. The specific communication overhead expression is as follows:

$$W_{z_i,z_j} = \frac{d_{z_i,z_j}}{\mu_{z_i,z_j}} + \sigma_{z_i,z_j} \tag{2}$$

$$W_{z_i,c} = \frac{d_{z_i,c}}{\mu_{z_i,c}} + \sigma_{z_i,c} \tag{3}$$

$d_{z_i,z_j} / \mu_{z_i,z_j}$ represents the transmission delay caused by fog equipment z_j transmitting data d_{z_i,z_j} to fog equipment z_i. μ_{z_i,z_j} is the data transmission rate of $\{z_i, z_j\}$ communication links. Similarly, $d_{z_i,c} / \mu_{z_i,c}$ indicates the transmission delay caused by fog equipment z_i transmitting data $d_{z_i,c}$ to cloud computing center C, and $\mu_{z_i,c}$ is the data transmission rate of $\{z_i, C\}$ communication link. σ_{z_i,z_j}, $\sigma_{z_i,c}$ represent other delays of data transmission on $\{z_i, z_j\}$ and $\{z_i, C\}$ communication links, respectively.

Because the processing time of distributed computing total task is equal to the maximum computing delay of all subtasks. Therefore, in order to achieve the goal of minimum processing delay, we must find an optimal set of $\{\delta_i, \delta_c\}$ to minimize the objective function $t(\delta_i, \delta_c)$. In summary, the total task processing delay is modeled as follows:

$$\min\left\{\max\left[\frac{\delta_i D}{\lambda_{z_i}} + \left(\frac{d_{z_i,z_j}}{\mu_{z_i,z_j}} + \sigma_{z_i,z_j}\right), \frac{\delta_c D}{\lambda_c} + \left(\frac{d_{z_i,c}}{\mu_{z_i,c}} + \sigma_{z_i,c}\right)\right]\right\}$$

$$\times i, j = 1, 2, \cdots k$$

$$s.t. \sum_{i=1}^{k} \delta_i + \delta_c = 1 \tag{4}$$

(2) Delay modeling under the constraints of fog network energy consumption

Based on time delay modeling, the total energy consumption during execution of task D is modeled and analyzed. Considering the particularity of cloud computing center, this paper only models the energy consumption of fog network, and studies the delay model under the constraints of energy consumption Taking the sub-task as a unit, in distributed computing process, the energy consumption E_i generated by each sub-task i during allocation execution process can be expressed as:

$$E_i = E_{trans,i} + E_{proc,i} \tag{5}$$

where $E_{trans,i}$ represents the transmission energy consumption of subtask i during transmission from source computing node to target computing node. $E_{proc,i}$ represents the processing energy consumed by target computing node executing subtask i. In fact, the equipment itself generates some inherent energy consumption. However, because this paper mainly studies the relationship between distributed processing tasks and energy consumption of system, the inherent energy consumption of equipment is ignored during modeling process.

Assume that subtask i is allocated to equipment z_i for execution by the device z_j. The transmission energy consumption and processing energy consumption are:

$$E_{trans,i} = \theta_{z_i,z_j}, d_{z_i,z_j} \tag{6}$$

$$E_{proc,i} = \theta_{z_i,}, d_i \tag{7}$$

In the formula, θ_{z_i,z_j} is the energy consumption for transmitting single-bit data and $\theta_{z_i,}$ is the energy consumption for processing single-bit data of equipment z_i. Then the total energy consumption generated by entire task D during execution. That is, the total energy consumption E_{sys} generated by software defined cloud-fog system processing task D is:

$$E_{sys} \sum_{z_i,z_j \in V} \theta_{z_i,z_j}, d_{z_i,z_j} + \sum_{i=1}^{k} \theta_{z_i,d_i} \tag{8}$$

The total energy consumption generated during task execution can be minimized by optimizing the clock frequency of computing device chip. Essentially, it means that CPU can reduce power consumption by executing applications slowly. However, task must be completed within a specified deadline for delay-sensitive applications. Otherwise, it will destroy the user's experience. Therefore, for software defined cloud-fog system, processing delay should be minimized under the condition that overall energy consumption of system is limited. That is, under the constraint that total energy consumption of system is less than energy consumption threshold $E_{\max}$, an optimal load balancing scheme is obtained. And then obtain an optimal set of δ_i such that the objective function t is minimized. In summary, the delay modeled by energy constraints of fog network is as follows:

$$\begin{aligned}
&\min \left\{ \max \left[\frac{\delta_i D}{\lambda_{z_i}} + \left(\frac{d_{z_i,z_j}}{\mu_{z_i,z_j}} + \sigma_{z_i,z_j} \right) \right] \right\} \\
&s.t. E_{sys} \sum_{z_i,z_j \in V} \theta_{z_i,z_j}, d_{z_i,z_j} + \sum_{i=1}^{k} \theta_{z_i,d_i} \leq E_{\max} \\
&\sum_{i=1}^{k} \delta_i = 1 i, j = 1, 2, \cdots k
\end{aligned} \tag{9}$$

C. VEHICLE MOVEMENT MODEL

For vehicle movement, $\gamma(k)$ is defined as the set of vehicle movement directions with time-tolerant data. In this paper, only the movement of roads or vehicles (or the data contained in it) along south-north and east-west directions is considered. Therefore, five possible cases are included: northward (N), southward (S), eastward (E), westward (W), and stationary RSU direction (F), which is $\gamma(k) \in \{N, S, E, W, F\}$.

In this paper, Manhattan mobile model is selected as the mobile model of vehicles in the proposed network architecture [20]. Suppose the density of vehicles and road intersections are ρ_{veh} and ρ_{int}, respectively, and the speed of vehicle is v. The time for vehicles to move randomly is t_{move} and satisfies the uniform distribution $U(0, 1/\rho_{int} v)$. Thus, the probability density function of vehicle's random movement time can be written as

$$f_{T_{move}}(t_{move}) = \begin{cases} \rho_{int} v & t_{move} \in [0, 1/\rho_{int} v] \\ 0, & others \end{cases} \tag{10}$$

For vehicles arriving at a road intersection, let P_{wait} denote the probability that vehicles needs to wait, and the time that vehicles waits randomly at intersection is denoted t_{wait}. If random waiting time satisfies the uniform distribution $U(0, T_{wait})$, the probability density function can be expressed as

$$f_{T_{wait}}(t_{wait}) = \begin{cases} 1/T_{wait}, & t_{move} \in [0, T_{wait}] \\ 0, & others \end{cases} \tag{11}$$

Consequently, the probability of vehicles moving and stopping can be expressed as

$$P_{move} = \frac{1/\rho_{int} v}{1/\rho_{int} v + (T_{wait} P_{wait}/2)} = \frac{2}{2 + T_{wait} P_{wait} \rho_{int} v} \tag{12}$$

$$p_{shop} = 1 - P_{move} = \frac{T_{wait} P_{wait} \rho_{int} v}{2 + T_{wait} P_{wait} \rho_{int} v} \tag{13}$$

where: $1/\rho_{int} v$ is the time that vehicles move from one intersection to next intersection; $T_{wait}/2$ is the average waiting time at intersection; $1/\rho_{int} v + T_{wait} P_{wait}/2$ is the total time that vehicles move and wait.

In addition, for vehicle A driving on the road, data transmission as its neighbor can be divided into two types: If vehicle B and vehicle A travel in the same direction and stay within the mutual communication range. Vehicle B can be a neighbor of vehicle A and establish a direct communication link; If vehicle A is stationary at intersection, other vehicles waiting at intersection or roadside RSU are within the communication range of vehicle A. Then these vehicles and RSU can be neighbors of vehicle A and can establish a wireless communication link with vehicle A. Except for these two cases, vehicle A will not be able to choose a neighbor to establish a wireless communication link with its neighbor in IoV.

IV. OPTIMIZATION OF NETWORK AND COMPUTING RESOURCES BASED ON POMDP

A. POMDP MODELING

POMDP can be regarded as a special Markov model because it emphasizes "partial observability" in the state transition process [21], [22]. In software defined IoV, it is necessary to pass state observations because the delay tolerable data transmission is not successfully obtained. Therefore, the transmission network and computing node selection problem with the minimum system overhead can be modeled as a POMDP optimization problem. Through the state observation information including action and historical state information, optimal action decision in each time slot will be selected, so that system obtains the lowest system overhead.

Action space A is considered as a joint action space, which contains network transmission selection and computation node selection for delay-tolerant data. At time t_k, the current storage needs to select the network for data transmission and the server for data calculation processing. Besides, $a(k) \in A$ represents the mixed action at this time, and is specifically expressed as

$$a(k) = [a_t(k), a_c(k)] \tag{14}$$

where $a_t(k)$ and $a_c(k)$ are the network transmission selection and data computing node selection of delay-tolerant data, respectively. Specifically, $a_t(k)$ is expressed as

$$a_t(k) = \begin{cases} 0, & \mathit{Data\ stored\ in\ OBU} \\ 1, & \mathit{Data\ transmission\ through\ IOT\ network} \\ \dot{\gamma} & \mathit{Data\ transfer\ to\ adjacent\ fog} \\ & \mathit{computing\ equipment}\ e' \end{cases} \tag{15}$$

In the formula: e' is the adjacent fog computing equipment that can be selected and transmitted in current storage. $\dot{\gamma} \in \{N, S, E, W, F\}$ and $a_c(k)$ can be expressed as

$$a_c(k) = \begin{cases} 0, & \mathit{Data\ is\ calculated\ locally} \\ 1, & \mathit{Data\ is\ calculated\ on\ Fog\ computing\ server} \end{cases} \tag{16}$$

The system state space set is defined as $\boldsymbol{S}$ at t_k and system state $s(k) \in \boldsymbol{S}$ includes current storage moving direction $\gamma_{cur}(k)$, the current storage distance to destination $d(k)$, and the data successful transmission index $\kappa(k)$. Hence, the system state is expressed as

$$s(k) = [\gamma_{cur}(k), d(k), \kappa(k)] \tag{17}$$

In addition, the data successful transmission index $\kappa(k)$ can be expressed as

$$\kappa(k) = [\kappa_N(k), \kappa_S(k), \kappa_E(k), \kappa_W(k), \kappa_F(k)] \tag{18}$$

each element in the formula is index of successful transmission in corresponding direction. For example, $\kappa(k)$ can be expressed as

$$K_N(k) = \begin{cases} 0, & \gamma_{e'}(k) = N \\ 1, & \mathit{others} \end{cases} \tag{19}$$

In the actual scenario, current storage can obtain the information of $\gamma_{cur}(k)$ and $d(k)$. However, the successfully transmitted information $\kappa(k)$ is difficult to obtain directly and accurately. Therefore, the successful data transmission index $\kappa(k)$ needs to be accurately obtained based on observation information.

For the successful transmission index $\kappa(k)$, its observation space set is defined as $\boldsymbol{O}$. At time t_k, the corresponding observation state is expressed as

$$o(k) = [o_N(k), o_S(k), o_E(k), o_W(k), o_F(k)] \tag{20}$$

each element in the formula is the observation state of corresponding direction. Take for example $o_N(k)$, which can be expressed as

$$o_N(k) = \begin{cases} 0, \mathit{no}\ e'\ \mathit{and}\ \gamma_{e'}(k) = N \\ 1,\ \mathit{others} \end{cases} \tag{21}$$

At the same time, the probability of observation state is defined as conditional probability of system state when it moves from $s(k) = i$ to $s(k+1) = j$ under action $a(k)$, expressed as

$$O^a_{o|s}(i,j) = \frac{P_{\kappa|o,\gamma_{cur}}\{i,j\} P_{o|,\gamma_{cur}}\{i,j\}}{\sum\limits_{\dot{o}_{j\in O}} P_{\kappa|o,\gamma_{cur}}\{i,j\} P_{o|,\gamma_{cur}}\{i,j\}} \tag{22}$$

By introducing this probability, the characteristics of "observation" in POMDP are fully highlighted. In this way, in software defined IoV, whether the transmission of tolerable time delay data cannot be accurately obtained and the state observation is needed. In equation (22) $P_{\kappa|o,\gamma_{cur}}\{i,j\}$ and $P_{o|,\gamma_{cur}}\{i,j\}$ can be expressed as

$$P_{\kappa|o,\gamma_{cur}}\{i,j\} = \prod_{\dot{\gamma}\in\{N,E,S,W,F\}} P_{\kappa\dot{\gamma}|o,\gamma_{cur}}\{i,j\} \tag{23}$$

$$P_{o|\gamma_{cur}}\{i,j\} = \prod_{\dot{\gamma}\in\{N,E,S,W,F\}} P_{o\dot{\gamma}|\gamma_{cur}}\{i,j\} \tag{24}$$

At this time, in order to calculate formulas (23) and (24), there are two cases that need to be discussed separately, and the details are as follows:

When $\gamma_{cur}(k) \neq F$, $P_{o\dot{\gamma}|\gamma_{cur}}\{i,j\}$ can be expressed as

$$P_{o\mathrm{F}|\gamma_{cur}}\{i,j\} = \begin{cases} 1, & o_F(k) = 0 \\ 0, & o_F(k) = 1 \end{cases} \tag{25}$$

and

$$P_{o\dot{\gamma}|\gamma_{cur}}\{i,j\} = \begin{cases} \left(1 - \dfrac{P_{move}M_{trange}}{M_{road}}\right)^{\rho_{veh}} & o_{o\dot{\gamma}}(k) = 0 \\ 1 - \left(1 - \dfrac{P_{move}M_{range}}{M_{trad}}\right)^{\rho_{veh}} & o_{o\dot{\gamma}}(k) = 1 \end{cases} \tag{26}$$

And $P_{\kappa|o,\gamma_{cur}}\{i,j\}$ can be expressed as

$$P_{\kappa\boldsymbol{F}|o,\gamma_{cur}}\{i,j\} \begin{cases} 1, & \kappa_F(k) = 0 \\ 0, & \kappa_{\boldsymbol{F}}(k) = 1 \end{cases} \tag{27}$$

and

$$P_{\kappa\dot{\gamma}|o,\gamma_{cur}}\{i,j\} = \begin{cases} 1, & o_{\dot{\gamma}}(k) = 0\ \mathit{and}\ \kappa_{\dot{\gamma}}(k+1) = 0 \\ 0, & o_{\dot{\gamma}}(k) = 0\ \mathit{and}\ \kappa_{\dot{\gamma}}(k+1) = 1 \\ P_{sue} & o_{\dot{\gamma}}(k) = 1\ \mathit{and}\ \kappa_{\dot{\gamma}}(k+1) = 0 \\ 1 - P_{sue} & o_{\dot{\gamma}}(k) = 1\ \mathit{and}\ \kappa_{\dot{\gamma}}(k+1) = 1 \end{cases} \tag{28}$$

where P_{suc} is defined as the probability that data is successfully transmitted from vehicle A to its neighbors, which can be expressed as

$$P_{suc} = \begin{cases} \frac{\rho_{int} v D_{vehA}}{H_*} - \frac{\rho_{int} v T_{wait}}{2}, \\ \qquad \frac{D_{vehA}}{H_*} \leq \min\left[t_{more} + t_{wait}, \delta t_k\right] \\ 0, \\ \qquad \frac{D_{vehA}}{H_*} > \min\left[t_{more} + t_{wait}, \delta t_k\right] \end{cases} \tag{29}$$

where D_{vehA} is the capacity of transmitted data packet; $H_* \in \{H_{veh}, H_{loT}\}$.

When $\gamma_{cur}(k) = F$, $P_{o\dot{\gamma}|\gamma_{cur}}\{i,j\}$ can be expressed as

$$P_{o\mathrm{F}|\gamma_{cur}}\{i,j\} = \begin{cases} 1 - P_{int\_RSU}, & o_F(k) = 0 \\ P_{int\_RSU}, & o_F(k) = 1 \end{cases} \tag{30}$$

and

$$P_{o\dot{\gamma}|\gamma_{cur}}\{i,j\} = \begin{cases} \left(1 - \frac{P_{move} M_{range}}{M_{road}}\right)^{\rho_{veh}} & o_{o\dot{\gamma}}(k) = 0 \\ 1 - \left(1 - \frac{P_{move} M_{range}}{M_{road}}\right)^{\rho_{veh}} & o_{o\dot{\gamma}}(k) = 1 \end{cases} \tag{31}$$

And $P_{\kappa|o,\gamma_{cur}}\{i,j\}$ can be expressed as

$$P_{\kappa\dot{\gamma}|o,\gamma_{cur}}\{i,j\} \begin{cases} 1, & \kappa_{o\dot{\gamma}}(k+1) = 0 \\ 0, & \kappa_{o\dot{\gamma}}(k+1) = 1 \end{cases} \tag{32}$$

The one-step transition probability of the system state is defined as $P_s^a(i,j)$, which can be expressed as

$$P_s^a(i,j) = \sum_{o\in 0}\left[P_{k|o,\gamma_{cur}}(i,j)\, P_{\gamma_{cur}|}(i,j) \times \prod_{\dot{\gamma}\in\{N,S,E,W,F\}} P_{o\dot{\gamma}|\gamma_{cur}}(i,j)\right] \tag{33}$$

$P_{k|o,\gamma_{cur}}(i,j)$ and $P_{o\dot{\gamma}|\gamma_{cur}}(i,j)$ have been given calculation methods before, and $P_{\gamma_{cur}|a}(i,j)$ needs to be given according to different actions. The specific calculation process is as follows:

When $a_t(k) \in \{0,\}$, $a_c(k) \in \{0, 1\}$. Under this condition, the delay-tolerable data will remain in current storage. Computing tasks can be performed locally or on fog computing equipment. At this time, the probability $P_{\gamma_{cur}|a}(i,j)$ can be expressed as

$$P_{\gamma_{cur}|a}(i,j) = \begin{cases} 1, & \gamma_{cur}(k) = F,\ \gamma_{cur}(k+1) = F \\ P(\dot{\gamma}_i|\dot{\gamma}_i) & \gamma_{cur}(k) =, \dot{\gamma}_i\ \gamma_{cur}(k+1) = \dot{\gamma}_i \\ 0, & others \end{cases} \tag{34}$$

When $a_t(k) \in \{1\}$, $a_c(k) \in \{1\}$. Under this condition, the delay-tolerant data will be transmitted by IOT facility, and the computing task can optionally be performed on fog computing equipment. At this time, the probability $P_{\gamma_{cur}|a}(i,j)$ can be expressed as

$$P_{\gamma_{cur}|a}(i,j) = \begin{cases} 0 & \gamma_{cur}(k+1) = \dot{\gamma}_i \\ 1, & others \end{cases} \tag{35}$$

When $a_t(k) \in \{N, S, E, W, F\}$, $a_c(k) \in \{0, 1\}$. Under this condition, the delay-tolerant data will be transmitted by IoV, and the computing tasks can be selected to be performed locally or on fog computing equipment. At this time, the probability $P_{\gamma_{cur}|a}(i,j)$ can be expressed as

$$P_{\gamma_{cur}|a}(i,j) = \begin{cases} P_{suc}P(\dot{\gamma}_i|\dot{\gamma}_i) & \gamma_{cur}(k) = \dot{\gamma}_i, \gamma_{cur}(k+1) = \dot{\gamma}_i \\ 0, & others \end{cases} \tag{36}$$

In software defined IoV, current storage can use the proposed method to select optimal decision to obtain the minimum system overhead. That is expressed as the minimum network overhead and the minimum data calculation execution time. For vehicle A, its system overhead $E_{vehA}(k)$ can be expressed as

$$E_{\mathrm{veh}A}(k) = \begin{cases} \eta t_{\mathrm{veh}A}(k) + \zeta C_{\mathrm{veh}A}(k), & a_t(k) \in \{0, \dot{\gamma}\} \\ \eta t_{\mathrm{veh}A}(k) + \zeta C_{IoT}(k), & a_t(k) \in \{1,\} \end{cases} \tag{37}$$

In the formula: η and ζ are weight factors and satisfy $0 \leq \eta$, $\zeta \leq 1$ and $\eta + \zeta = 1$. C_{vehA} and C_{IoT} are the network overheads when IoV and IOT networks are selected for current storage respectively.

The total expected overhead for system over the entire time period can be expressed as

$$E = \arg\min_{\{a_t,a_c\}}\left(\sum_{k=0}^{K}\sum_{vehA=1}^{N_{vehA}} \sigma^{K-k} E_{vehA}(k)\right) \tag{38}$$

where σ is the discount factor, which satisfies $0 \leq \sigma \leq 1$. Each user in system achieves the optimization objective of formula (38) by selecting the optimal transmission decision of network transmission and calculation server in each time period. At the same time, the discount factor σ is used to ensure the convergence of proposed optimization algorithm and to ensure that the optimal solution of algorithm can be obtained.

B. OPTIMIZATION PROBLEM SOLVING

Through the above discussion, the optimal strategy modeling based on POMDP is given. For function solving problems, dynamic value iterations and Bayesian-based information state updates are used. Given value function $\varphi_k(\pi(k))$, it is defined as the minimum system overhead at time slot $\delta t(k)$ and is specifically expressed as

$$\varphi_k(\pi(k)) = \min_{a(k)\in A}\left\{\sum_{i\in s}\sum_{j\in s}\pi_i^k P_s^a(i,j)\sum_{o(k)\in s} O_{o|s}^a(i,j)\right.$$

$$\times\left[Q_{vehA}(k)+\varphi_{k+1}(\pi(k+1))\right]\Bigg\} \tag{39}$$

where $\pi(k)$ is the information space and can be expressed as

$$\pi(k)=\left\{\pi_1^k,\pi_2^k,\cdots\pi_i^k,\pi_j^k\cdots\pi_s^k\right\} \tag{40}$$

For π_i^k, it can be obtained by Bayesian update rules. Specifically, it is expressed as

$$\pi_j^{k+1}=\frac{\sum\limits_i \pi_i^k P_s^a(i,j)\,O_{o|s}^a(i,j)}{\sum\limits_i\sum\limits_j \pi_i^k P_s^a(i,j)\,O_{o|s}^a(i,j)} \tag{41}$$

By Bayesian update iterative calculation, the information state probability in POMDP is solved. According to equation (39), the state transition probability and observation probability at each moment can be mapped in information state. The information state probability based on state transition and state observation information at each moment can be obtained by iterative calculation. This provides a basis for the selection of optimization decisions.

For proposed POMDP algorithm, its computational complexity depends on the matrix dimensions of action space and state transition space modeled by POMDP. In network model proposed in this paper, the matrix dimensions formed by state transition space and action space are $5\times5\times D$ and 7×2 respectively, where D is the discrete levels of state $d(k)$. In actual process, optimal selection process needs to perform calculation of the above spatial dimensions and function iteration. At the same time, computational complexity may increase with the increase of action space and state dimension space. Unlike existing network models, SDN controller is introduced into system model presented in this paper. These complicated calculation processes will be executed offline, which will greatly reduce system's real-time calculation volume. When the values of all parameters are given, optimal action decision of system with the least overhead will be calculated and determined offline. At the same time, if parameter changes, the value function will be recalculated offline and dynamically updated. That is, once all parameters are determined, the optimal strategy is obtained and stored in controller. Online, system will no longer perform complex calculations repeatedly. Besides, current storage only selects the corresponding optimal action based on current state.

V. SIMULATION

A. EXPERIMENTAL SETTINGS

In order to verify that proposed scheme can reduce the delay more effectively when processing the IoV data. And it can effectively optimize the delay under the constraints of energy consumption in fog network to achieve a compromise between delay and energy consumption. Simulations in this section realize the delay optimization under the constraint of energy consumption. The proposed scheme is compared with delay performance of cloud computing, fog computing network and single fog node.

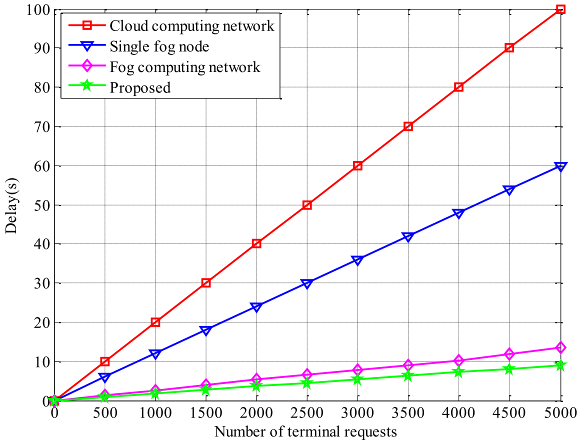

FIGURE 4. Comparison of task processing delay between the proposed scheme and other three architectures.

Experimental platform is MATLAB, the experimental computer CPU is Intel i5-4590 and its memory is 8GB. The computing power and communication delay of cloud computing equipment in the experiment were set according to literature [23], and fog equipment energy consumption parameter settings refer to reference [13], the vehicle networking data in the experiment are all simulated settings. Take the collision detection and early warning service in IoV as an example. The amount of data required to perform a single collision warning mission is approximately 500Kb. The processing of a single request task executes about 10 millions of instructions and the size of each instruction is 64 bits. In addition, the weighting factor is set to 0.5 and the on-board CPU performance is 0.5GHz. The network contains a total number of vehicles set to 1000, vehicle moving speed v is set to $8\text{m}*\text{s}^{-1}$ and the distance between adjacent intersections is 500 meters. The intersection density ρ_{veh} and ρ_{int} is 0.002m^{-1} and vehicle intersection waiting probability is 0.8. The waiting time $T_{wait}/2$ at the intersection of vehicle is 150s and the performance of on-board CPU is 0.5GHz.

The number of nodes in fog layer is 10, default equipment z_1 is the node that receives tasks and assigns tasks. Specifically, the relevant parameters of each fog computing node are shown in Table 1. The communication delay only considers the communication delay between equipment receiving tasks and tasks distributing equipment. And tasks initialization uses random allocation.

B. COMPARISON OF LATENCY PERFORMANCE BETWEEN THE PROPOSED SCHEME AND THREE ARCHITECTURES INCLUDING CLOUD

Under the condition of unlimited energy consumption, this paper simulates delay performance of the proposed scheme network, and compare it with cloud computing network, fog computing network and single fog node. Among them, the selected single fog node is node z_1 that receives tasks. The results are shown in Figure 4.

TABLE 1. Parameters of fog computing nodes.

Parameter type	C	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8	z_9	z_{10}
computing power (MIPS)	12000	900	1100	1000	1300	800	1400	700	1100	1200	1000
uplink broadband (MIPS)	2	83	87	78	84	87	79	85	82	85	88
downlink broadband (MIPS)	1.9	101	103	102	98	103	101	104	102	103	100
other one-way delay (ms)	10	1.0	1.1	1.2	1.2	1.0	1.2	1.3	1.4	1.1	1.2
transmission energy consumption (nJ/bit)	-	23	25	20	22.5	18.5	19.5	17.5	24	18	21
calculate energy consumption (nJ/bit)	-	12.5	6.25	9.25	13.5	15	13.75	17.5	18.75	11.25	16.25

Simulation results in Figure 4 show that when the number of user requests is less than 500, the number of tasks to be processed is small. Although transmission delay of Cloud is high, the processing speed of cloud computing servers is much faster than that of Fog and the proposed solution. Therefore, delay performance gap between the four is not obvious. However, the latency of Cloud and a single fog node is significantly higher than that of Fog and the proposed scheme with the increasing number of task requests. Because the cloud server is far from end users and the bandwidth is limited, more users requesting data to be transmitted to Cloud for calculation will result in a higher transmission delay, which increases the total task processing delay significantly. While a single fog node processing task does not cause transmission delay, but the computing power is too weak. In addition, it can be seen from Figure 4 that the use of cloud computing center as a distributed computing node enhances the overall computing performance of proposed solution. Consequently, when the number of user requests is greater than 3000, the delay of proposed scheme is slightly lower than Fog, which is very advantageous for dealing with delay-sensitive services. In summary, the proposed scheme can more effectively support delay-sensitive services in IoV, which improves service quality and enhances user experience.

C. IMPACT OF UPLINK BANDWIDTH ON THE PROPOSED SCHEME

In fact, due to link bandwidth competition, the bandwidth obtained when fog nodes transmit data to cloud data center changes in real time. Therefore, this paper studies the impact of uplink bandwidth of fog-to-cloud transmission data on the task processing delay of proposed scheme. The proposed scheme uses POMDP strategy and the number of user requests processed is 5000. Besides, the results are shown in Figure 5.

The results show that with continuous increase of uplink bandwidth, the delay of cloud computing network continues to decrease, and the delay of proposed scheme remains slightly unchanged after a slight decrease. When the uplink bandwidth is less than 10Mbps, the delay of proposed scheme is slightly reduced because of increased bandwidth. The service processing delay of cloud computing network is reduced, and the total delay of proposed solution is reduced. After that, the delay remains unchanged because processing delay of cloud computing network continues to decrease, which makes the node with the largest processing delay in proposed solution change from a cloud computing network to a node in og computing network. However, because the performance parameters of fog computing network are unchanged, the total delay of proposed scheme remains unchanged. In summary, when the uplink bandwidth is limited, the delay performance advantage of proposed scheme is obvious. To improve the latency performance of cloud computing networks, the increase in uplink bandwidth is essential.

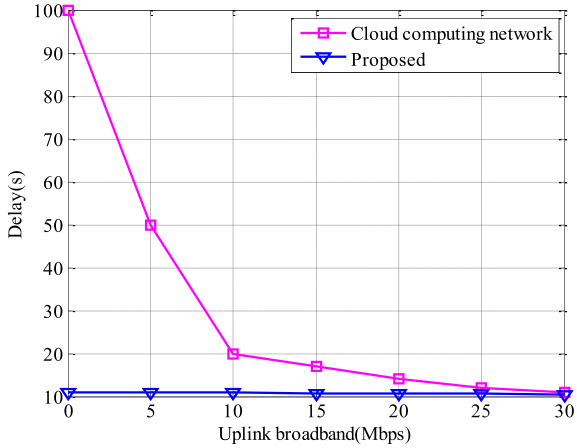

FIGURE 5. Impact of uplink bandwidth on the proposed scheme.

D. COMPARISON OF DELAY PERFORMANCE BETWEEN THE PROPOSED SCHEME AND SOME ADVANCED SCHEMES

In order to test the performance of proposed scheme in reducing service processing delay. Under the constraints of

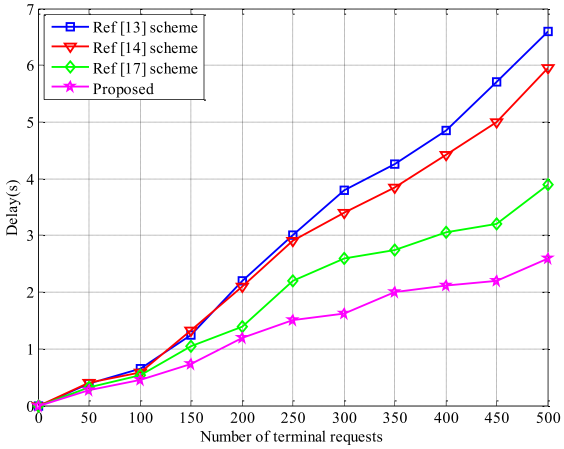

FIGURE 6. Comparison of delay performance between the proposed scheme and some advanced schemes.

fog network energy consumption, we compare the proposed scheme with three load balancing algorithms such as Ref [13], Ref [14] and Ref [17]. The simulation results are shown in Figure 6.

Figure 6 shows that when the amount of user request tasks to be processed is small, such as when the number of requests is less than 100, the delay differences between the four algorithms are not obvious and the resulting delays are similar. However, the advantages of proposed solution slowly emerge as the number of requests continues to increase. When the number of requests is higher than 30, it is obviously better than the Ref [13] scheme, the Ref [14] scheme and the Ref [17] scheme. When the number of user requests is 100, the task processing delays of Ref [13] scheme, Ref [14] scheme, Ref [17] scheme and the proposed scheme are 6.68s, 5.95s, 3.89s and 2.54s respectively. Compared with other three comparison schemes, the delay performance of proposed scheme has obvious advantages. It is fully explained that proposed scheme is applied to solve the high transmission delay problem existing in many network architectures in IoV. It can reduce latency and improve user experience more effectively than using several other schemes.

VI. CONCLUSION

At present, user requests are increasing in the Internet, especially in IoT. For example, there are many user requests at the same time, and processing too many applications generates huge energy consumption in IoV. Therefore, in order to save energy consumption and minimize delay, it is particularly important for us to study the problem of delay optimization under the constraint of energy consumption. The cloud computing layer consists of high-performance server clusters that stores and analyzes data, service requests from terminal equipment. Besides, it provides a variety of integrated services. In addition, this paper proposes to treat cloud computing as a distributed computing node as a whole and forms a hybrid cloud-fog distributed computing network. The transmission process of delay-tolerant data in software defined IoV and the selection of computing nodes can be modeled as partially observable Markov decision process. Moreover, optimizing it can obtain the minimum system overhead, including the minimum network overhead and the shortest data calculation processing time. Although the cloud computing center is often far away from users, it can also be used as distributed computing nodes with strong computing capacity and high transmission delay to handle some tasks. Thus, this paper integrates cloud computing, SDN and fog computing to form a software defined cloud-fog network architecture for IoV. In addition, in order to save energy consumption and reduce service processing delay effectively, this paper studies delay optimization model under the constraints of fog network energy consumption. And then the strategy of compromise between delay and energy consumption is proposed.

This paper conducts a preliminary study on the network architecture of fog computing applied to IoT and how to optimize the latency of fog computing networks using load balancing strategies. However, there are still some deficiencies in our work. Firstly, the research on fog computing networks is still in the theoretical stage. The realization and application of specific fog network platform will be the focus of future research. Secondly, the load balancing strategy of fog network studied in this paper is mainly a centralized strategy. For improve the survivability and reliability of fog network in the future, we can consider studying distributed load balancing strategy suitable for fog network. Furthermore, this paper studies how to use intelligent strategies to optimize the latency of fog networks and other load balancing strategies with better performance need further study. In addition, reducing the time complexity of intelligent optimization algorithms is also the focus of next research.

REFERENCES

[1] H. Gao, Y. Duan, L. Shao, and X. Sun, "Transformation-based processing of typed resources for multimedia sources in the IoT environment," *Wireless Netw.*, 2019, doi: 10.1007/s11276-019-02200-6.

[2] H. Gao, W. Huang, and X. Yang, "Applying probabilistic model checking to path planning in an intelligent transportation system using mobility trajectories and their statistical data," *Intell. Automat. Soft Comput.*, vol. 25, no. 3, pp. 547–559, 2019.

[3] X. Wang, Y. Ting-Ting, and H. Shuang-Shuang, "Parallel Internet of vehicles: The ACP-based networked management and control for intelligent vehicles," *Acta Automatica Sinica*, vol. 44, no. 8, pp. 1391–1404, 2018.

[4] P.-W. Tsai, C.-W. Tsai, C.-W. Hsu, and C.-S. Yang, "Network monitoring in software-defined networking: A review," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3958–3969, Dec. 2018.

[5] O. Lemeshko and O. Yeremenko, "Enhanced method of fast re-routing with load balancing in software-defined networks," *J. Electr. Eng.*, vol. 68, no. 6, pp. 444–454, Nov. 2017.

[6] Q. Wang, S. Guo, J. Liu, and Y. Yang, "Energy-efficient computation offloading and resource allocation in fog computing for Internet of everything," *China Commun.*, vol. 16, no. 3, pp. 32–41, 2019.

[7] Y. Jie, M. Li, and C. Guo, "Game-theoretic online resource allocation scheme on fog computing for mobile multimedia users," *China Commun.*, vol. 16, no. 3, pp. 22–31, 2019.

[8] X. Ma, H. Gao, H. Xu, and M. Bian, "An IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, p. 249, 2019, doi: 10.1186/s13638-019-1557-3.

[9] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicu-laradhoc network with fog computing," in *Proc. IEEE Int. Symp. Integr. Netw. Manage.*, Ottawa, ON, Canada, May 2015, pp. 1202–1207, doi: 10.1109/INM.2015.7140467.

[10] Y. Lin and H. Shen, "Cloud fog: Towards high quality of experience in cloud gaming," in *Proc. 44th Int. Conf. Parallel Process.*, Beijing, China, 2015, pp. 500–509.

[11] K. Intharawijitr, K. Iida, and H. Koga, "Analysis of fog model considering computing and communication latency in 5G cellular networks," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, Mar. 2016, pp. 1–4, doi: 10.1109/PERCOMW.2016.7457059.

[12] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system," *IEEE Trans. Comput.*, vol. 65, no. 12, pp. 3702–3712, Dec. 2016.

[13] S. Sarkar and S. Misra, "Theoretical modelling of fog computing: A green computing paradigm to support IoT applications," *IET Netw.*, vol. 5, no. 2, pp. 23–29, Mar. 2016.

[14] R. Deng, R. Lu, and C. Lai, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.

[15] H. Gao, Y. Xu, Y. Yin, W. Zhang, R. Li, and X. Wang, "Context-aware QoS prediction with neural collaborative filtering for Internet-of-Things services," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2019.2956827.

[16] S.-M. Oh and J. Shin, "An efficient small data transmission scheme in the 3GPP NB-IoT system," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 660–663, Mar. 2017.

[17] Q. Wang, S. Guo, J. Liu, and Y. Yang, "Energy-efficient computation offloading and resource allocation for delay-sensitive mobile edge computing," *Sustain. Comput., Informat. Syst.*, vol. 21, pp. 154–164, Mar. 2019.

[18] H. Gao, W. Huang, Y. Duan, X. Yang, and Q. Zou, "Research on cost-driven services composition in an uncertain environment," *J. Internet Technol.*, vol. 20, no. 3, pp. 755–769, 2019.

[19] S. Yan, A. Aguado, and Y. Ou, "Multi-layer network analytics with SDN-based monitoring framework [Invited]," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 9, no. 2, pp. 271–279, Feb. 2017.

[20] S. Kr. Maakar, Y. Singh, and A. L. Sangal, "Traffic pattern based performance comparison of two proactive MANET routing protocols using manhattan grid mobility model," *Int. J. Comput. Appl.*, vol. 114, no. 14, pp. 26–31, 2015.

[21] L. Tang, R. Liang, and Y. Zhang, "Load balance algorithm based on POMDP load-aware in heterogeneous dense cellular networks," *J. Electron. Inf. Technol.*, vol. 39, no. 9, pp. 2134–2140, 2017.

[22] Y. F. Zhou and N. Chen, "The LAP under facility disruptions during early post-earthquake rescue using PSO-GA hybrid algorithm," *Fresenius Environ. Bull.*, vol. 28, no. 12A, pp. 9906–9914, 2019.

[23] S. Yi, Z. Hao, and Z. Qin, "Fog computing: Platform and applications," in *Proc. 3rd IEEE Work-Shop Hot Topics Web Syst. Technol.*, Washington, DC, USA, Nov. 2015, pp. 73–78, doi: 10.1109/HotWeb.2015.22.

BIN XIA graduated from the Logistical Engineering University of PLA, in 2014. He received the Ph.D. degree in barracks planning and management. He is currently an Associate Professor with the Chengyi University College, Jimei University. His main research interests include traffic engineering, construction engineering, and urban planning.

FANYU KONG was born in Heilongjiang, China. He graduated from Tongji University, in 2008. He received the Ph.D. degree in transportation planning and management. He is currently a Senior Engineer with Chongqing Technology and Business University. His research interests include traffic engineering, transportation management, and logistics system optimization.

JUN ZHOU was born in Jiangxi, China. He graduated from the Logistical Engineering University of PLA, in 2011. He received the Ph.D. degree in logistics informatization. He is currently a Professorship Senior Engineer with the Chongqing Business Vocational College. His research interest includes artificial intelligence.

XIAOSONG TANG was born in Jiangsu, China. He graduated from the Logistical Engineering University of PLA, in 2008. He received the Ph.D. degree in civil engineering. He is currently an Associate Professor with the Chongqing Vocational College of Public Transportation. His research interests include the stability of rock and soil engineering and its numerical analysis.

HONG GONG was born in Fujian, China. She graduated from the University of Technology Sydney, in 2009. She received the master's degree in professional accounting. She is currently a Lecturer with the Chengyi University College, Jimei University. Her research interests include financial accounting and logistics and supply chain management.