This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2816590, IEEE Transactions on Industrial Informatics

Offloading in Internet of Vehicles: A Fog-enabled Real-time Traffic Management System

Xiaojie Wang, Zhaolong Ning and Lei Wang

***Abstract*—Fog computing has been merged with Internet of Vehicle (IoV) systems to provide computational resources for end users, by which low latency can be guaranteed. In this work, we put forward a feasible solution that enables offloading for real-time traffic management in fog-based IoV systems, aiming to minimize the average response time for events reported by vehicles. First, we construct a distributed city-wide traffic management system, in which vehicles close to RSUs can be utilized as fog nodes. Then, we model parked and moving vehicle-based fog nodes according to queueing theory, and draw the conclusion that moving vehicle-based fog nodes can be modeled as an $M/M/1$ queue. An approximate approach is developed to solve the offloading optimization problem by decomposing it into two subproblems and scheduling traffic flows among different fog nodes. Performance analyses based on a real-world taxi-trajectory dataset are conducted to illustrate the superiority of our method.**

***Index Terms*—Internet of Vehicle, offloading, fog computing, traffic management, real-time processing.**

I. Introduction

Great attentions have been drawn for Internet of Things (IoTs) over the last ten years, not only in academic fields but also in industry areas. IoTs consist of ubiquitous things in everyday life, e.g., smartphones, laptops, tablets, TVs and vehicles. The most attractive characteristic of IoTs is to form a heterogeneous network framework by integrating ubiquitous networks [1]. With the development of sensing, computing and networking tools as well as technologies, bulk of data are of rapid expansion in large-scale urban areas, including real-time traffic information, vehicular mobility information and social relationships [2], [3]. As a key branch of IoTs, Internet of Vehicle (IoV) has become a new research field for the development of industrial applications in smart cities, e.g., traffic management and road safety [4], [5]. A lot of countries have focused on the establishment of IoV systems, e.g., ERTICO-ITS in Europe. In industry, worldwide automakers have developed testbed systems based on Vehicle-to-Vehicle (V2V) communications, such as Volvo, BMW and Toyota.

Because the increasing growth of vehicles has caused air pollution and traffic congestions on the road continuously [6], it is deserved to study efficient traffic management schemes by taking timely actions to manage road traffic, with the purpose of achieving green transportations and alleviating kinds of traffic problems [7], [8]. Numerous researches and projects have been conducted to solve this problem by reducing the response time of traffic management server, most of which are based on the centralized data management, i.e., a centralized server is responsible for data processing [9]. However, the generated information by vehicles is always of local relevance, i.e., the data sensed by vehicles have their own lifetime and tempo-spatial scopes. For instance, the information about a traffic jam may be only valid for half an hour, and can merely draw attentions of vehicles that are moving towards the area where the traffic jam occurs [10]. Therefore, the design of decentralized traffic management systems is advocated. To construct such a system, we mainly face several challenges as follows:

- Traditional traffic management schemes are mainly based on centralized control mechanisms, posing heavy loads on the traffic management server and causing a large response delay. Therefore, how to construct a decentralized system model to manage the traffic in urban areas needs to be investigated.
- Although fog computing is promising to offload loads for the traffic management server, the installation and placement of fog nodes without requiring additional network costs are challenging. In addition, how to schedule and balance traffic loads among fog nodes also deserves to be well studied.
- With limited computational resources and low latency requirement for real-time applications in large-scale vehicular networks, how to offload network traffic in an optimized manner has to be developed.

In this paper, we design an oFfloading algOrithm for Real-time Traffic management in fog-based IoV systems, named FORT, with the purpose of minimizing the average response time of the traffic management server for messages. Specifically, we integrate fog computing with a decentralized traffic management system. The whole city is divided into several regions, and traffic events can be managed in the corresponding region. We first establish a three-layer system model, containing the cloud layer, cloudlet layer and fog layer. The cloud is far from vehicles, while a cloudlet and a number of fog nodes coexist in each region, managing messages reported by passing-by vehicles. Parked and moving vehicles can be utilized as fog nodes, with the ability of providing computational resources and capacities. Then, we model the traffic management system as a queueing system, and regard the cloudlet as a processing server based on an $M/M/n$ queue. We further model parked and moving vehicle-based fog nodes according to queueing theory. Since the formulated offloading optimization problem is NP-hard, we put forward

X. Wang, Z. Ning and L. Wang (corresponding author) are with the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software, Dalian University of Technology, Dalian, China. Email: lei.wang@dlut.edu.cn.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2816590, IEEE Transactions on Industrial Informatics

an approximate approach to divide the original problem into two subproblems and solve them iteratively.

To our best knowledge, this work is an early effort to provide a systematic design of offloading for real-time traffic management in fog-based IoV systems so far. We hope this pioneering work can throw light on how to utilize both parked and moving vehicles as fog infrastructures to extend the facility of clouds. Our contributions can be summarized as follows:

- We establish a three-layer system model, partitioning the whole city into several regions for distributed management. The cloudlet and fog layers provide potential computational capacities and resources for message processing without requiring additional network costs.
- We model parked and moving vehicle-based fog nodes according to queueing theory, and estimate the average response time based on their processing abilities. Specifically, we draw the conclusion that moving vehicle-based fog nodes can be viewed as a server with an $M/M/1$ queue. To our best knowledge, this is the first work to provide a detailed design about how to utilize parked and moving vehicles as fog nodes.
- A mathematical framework is cast to investigate the offloading problem by message flow scheduling among the cloudlet and fog nodes. In addition, an approximate approach with time complexity of $O(m^4)$ is proposed to solve the formulated optimization problem, and two subsystems are formed to reach the objective gradually.
- Performance evaluation based on a real-world taxi-trajectory dataset is provided to demonstrate the effectiveness of the designed method. Our method outstands the existing ones in terms of average system response time and required computational resources.

The rest of this paper is structured as follows: in Section II, we review the related work; we present the system model and formulate the studied problem in Section III; in Section IV, we design an approximate approach to minimize the average response time for the traffic management system, followed by performance evaluation in Section V; finally, we conclude this work in Section VI.

II. Related Work

In this section, we review the state-of-the-art for traffic management, fog computing in IoV systems and real-time resource management in fog computing.

A. Traffic Management

With the unprecedented development of vehicles and traffic flows, many researchers have focused on efficient traffic management schemes to alleviate traffic problems in urban areas [11]. A real-time path planning algorithm is proposed to reduce travel costs [12]. Stochastic Lyapunov optimization technique is employed to improve the overall spatial utilization of a road network by avoiding vehicles from traffic jams. RISA, a road information sharing architecture, is designed in vehicular networks [13]. It has a distributed network structure, based on which event information detected by vehicles can be aggregated and disseminated timely. A real-time traffic monitoring scheme based on emerging vehicular communication systems is put forward in [14]. The data collection is conducted based on cluster-based V2X communication patterns, and reliable traffic monitor can be achieved. However, most of these schemes are based on the centralized data management. Instead, we focus on a decentralized approach to manage traffic information in a city-wide area.

B. Fog Computing in IoV Systems

As a complementation of cloud computing, fog computing focuses on migrating computational resources to network edges. It has the advantages of reducing incoming traffic toward clouds and decreasing response delays of network requests [15]. Based on the study of computation transfer strategies with V2V and Vehicle-to-Infrastructure (V2I) communication modes, an efficient predictive scheme is proposed, by which tasks are offloaded to fog nodes via direct or predictive relay transmission mode [16]. A layered network architecture is established to enhance computational capabilities of vehicular networks [17]. An incentive scheme is proposed to maximize utilities of edge server owners and cloud server operators [18]. Nevertheless, most of these researches assume that fog nodes are either fixed or move along a fixed path, while generally ignoring the highly dynamic nature of vehicular networks.

The idea of vehicular fog computing is proposed in [19], which utilizes parked and moving vehicles as computational and communicational infrastructures. A quantitative analysis of the fog-enabled network capacity is carried out, and characteristics as well as relationships among communication connectivity, capability and mobility of vehicles are unveiled. However, it is merely a feasibility analysis.

C. Real-time Resource Management in Fog Computing

Real-time resource management is important in fog-based vehicular networks, because the transmission delay is the main challenge for the deployment of large-scale traffic management systems [20]. Moving vehicles can be utilized to enhance the processing ability of cloud computing for end users [21]. When the cloud is overloaded, potential free resources in vehicles can be scheduled to relieve computational resource consumption, which can largely reduce the response delay. An offloading algorithm enabling cooperative fog computing is proposed in [15]. Intra and inter-fog resource management schemes are designed by considering network latency and efficiency. In order to make a trade-off between delay and energy consumption, an optimization problem is formulated for the allocation of computational and communicational resources [22]. An energy-aware offloading algorithm is formulated. In addition, an iterative search scheme is designed to find the optimal solution.

In this work, we propose a fog-enabled offloading algorithm for real-time traffic management in IoV systems, with the purpose of minimizing the system response time. We first establish a three-layer system model. Then, we illustrate how to utilize parked and moving vehicles as fog nodes. Especially,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2816590, IEEE Transactions on Industrial Informatics

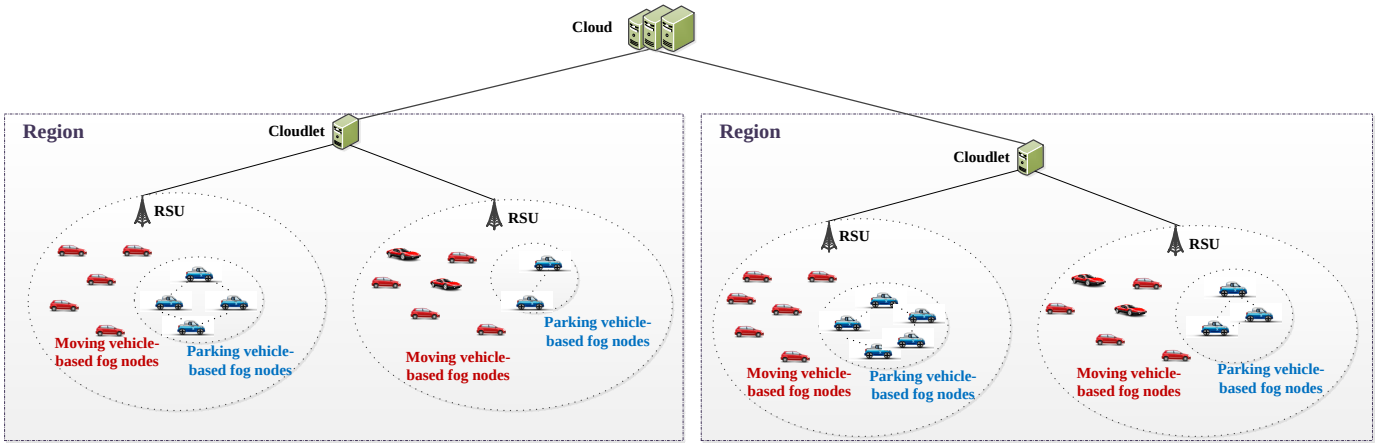


Fig. 1: Three-layer system model.

we draw the conclusion that moving vehicle-based fog nodes can be viewed as an $M/M/1$ queue. To our best knowledge, this is the first work to provide a detailed design about how to utilize parked and moving vehicles as fog nodes. After that, a traffic scheduling scheme is proposed to solve the offloading optimization problem.

III. System Model and Problem Formulation

In this section, we first present the three-layer system model. After that, the cloudlet model, fog model and offloading model for the traffic management system are described respectively. Then, we formulate the offloading problem in fog-based IoV systems.

A. Three-layer System Model

The system model is shown in Fig. 1, containing the cloud layer, the cloudlet layer and the fog layer.

1) Cloud layer: The cloud layer is always far from vehicles and formed by Traffic Management Server (TMS) and Trust Third Authorities (TTAs). In traditional traffic management systems(e.g., [9]), TMS is response for processing messages and informing individuals in the traffic management office to take actions. TTAs manage users' rewards and guarantee the fairness of the whole system. Specially, all the messages uploaded by vehicles in a city-wide area will be validated and processed by TMS, resulting in traffic overload. In this work, TMS is merely response for result reception and reward allocation. Messages uploaded by vehicles will be offloaded to other system components for processing.

2) Cloudlet layer: Since contents generated by vehicles are always of local interests, the information of road traffic conditions within a specific region are only meaningful to vehicles inside or around. Therefore, decentralized data management is advocated in IoV systems, which can manage the locally relevant information in special areas and reduce burdens for the centralized server. First, we divide a city into several regions. In the center of each region, there is a cloudlet to manage uploaded messages generated by vehicles inside. There are also several Road Side Units (RSUs) along roads within each region, acting as schedulers or access points to schedule uploaded messages for passing-by vehicles.

3) Fog layer: Fog nodes are formed by vehicles within the communication ranges of RSUs. Similar to [19], we assume that both parked and moving vehicles can be utilized to form fog nodes. If fog nodes are available, RSUs can directly upload messages to fog nodes (instead of cloudlets) for processing to shorten the response time.

The advantages of our three-layer system model compared with the traditional traffic management scheme are as follows: a) our model is based on a decentralized network structure, and the data process can be managed independently in each region; b) we use the cloudlet and fog nodes to offload traffic for TMS, which largely reduce its burdens; c) our model can largely reduce response delay, because fog nodes and the cloudlet are close to terminals.

In our system, vehicles can upload sensed events (e.g., traffic jams, car accidents and road surface damages) to a nearby RSU along their routes for traffic management. When an RSU receives a message, it will send the message to the cloudlet or fog nodes for processing by extracting the exact information of reported events. Then, the extracted information will be uploaded to TMS for further actions. Specially, the cloudlet and fog nodes are main processors in our system to handle messages reported by vehicles. Therefore, our main focus is to balance message loads among the cloudlet and fog nodes to minimize the response delay for the traffic management system.

Through analyzing real-world trajectories of taxies for more than one month in the city of Shanghai (China), it is a fact that the flow of vehicles arriving at an RSU follows a Poisson process with arrival rate $\lambda_i^{vehicle}$([23] and [24]). Therefore, the message uploading process can be viewed as a subprocess of the vehicular flow. Correspondingly, we consider that the messagee flow arriving at each RSU r_i follows a Poisson process, with arrival rate λ_i. In addition, we analyze network performance within one region as an example, and the analysis model is easy to be extended to all regions in the city-wide network. We consider that there exists a cloudlet c, a group of RSUs $R = \{r_1, ..., r_u\}$, a set of parked vehicle-based fog

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2816590, IEEE Transactions on Industrial Informatics

nodes $F^p = \{v_1^p, ..., v_l^p\}$, and a stream of moving vehicle-based fog nodes $F^m = \{v_1^m, v_2^m ... v_n^m\}$ within region G.

B. Cloudlet Model

The objective of our work is to reduce the response time of TMS by offloading and balancing loads among fog nodes and the cloudlet. The total response time can be obtained by $t_{tol} = t_{up} + t_{wat} + t_{pro} + t_{down}$, where t_{up} is the uploading time for a message from an RSU to a process server. The symbol t_{wat} is the waiting time for a message at the process server. The process time of a server for the message is denoted by t_{pro}, and t_{down} is the travel time for a feedback message to an RSU. For simplification, we can regard that $t_{up} = t_{down}$. Therefore, we utilize $t_{stol} = 2t_{up} + t_{wat} + t_{pro}$ to represent system response time for message processing.

We model the system as a queuing network, and the cloudlet can be modeled as an $M/M/b$ queue, with b homogeneous servers and fixed service rate μ_s. The service rate for a cloudlet is $\rho^c = \lambda^c / b\mu_s$. The symbol λ^c represents the flow waiting for the cloudlet to process, so that $\rho^c < 1$. Then, the waiting time t_{wat}^c can be computed by $t_{wat}^c = t_{que}^c + t_{ser}^c$, where t_{que}^c is the queueing time for a message, and t_{ser}^c is the service time. According to queueing theory [25], average queueing time $E(t_{que})$ for a message can be obtained by:

$$\begin{aligned} E(t_{que}) &= f(b, \rho) \\ &= \left[\sum_{k=0}^{b-1} \frac{\left(\frac{b}{k}\right)! \left(1-\rho^2\right)}{(b\rho)^{b-k} \rho} + \frac{1-\rho}{\rho} \right]^{-1}. \end{aligned} \tag{1}$$

The average service time is $E(t_{ser}^c) = \frac{1}{\mu_s}$. A network delay $d_{r_i \to cloudlet}$ is incurred by message transfer from RSU r_i to the cloudlet. Then, average message response time t_{stol}^c can be obtained by:

$$\begin{aligned} E(t_{stol}^c) &= E(t_{que}^c) + E(t_{ser}^c) + 2 \times t_{up}^c \\ &= f(b, \rho^c) + \frac{1}{\mu_s} + 2 \times d_{r_i \to cloudlet}. \end{aligned} \tag{2}$$

C. Fog Model

As described in Section III-A, two kinds of vehicles can be utilized for fog nodes, i.e., parked and moving vehicles. In this subsection, we mainly describe how to employ them as fog nodes.

1) Parked vehicle-based fog model: We consider that there are s time slots for 24 hours in a day. During each time slot, the total number of parked vehicles has no significant change. In addition, a parked vehicle is regarded as a fog node and has one server with a fixed service rate μ_s. Considering that there are l ($l > 0$) vehicles, we can model these parked vehicle-based fog nodes as an $M/M/l$ queue. The service rate of the parked vehicle-based fog nodes is computed by $\rho_i^{vf} = \lambda_i^p / l\mu_s$, where λ_i^p is the flow that waits for parked vehicle-based fog nodes to process, and $\rho_i^{vf} < 1$ holds. The average response time for one message is:

$$\begin{aligned} E(t_{stol}^p) &= E(t_{que}^p) + E(t_{ser}^p) + 2 \times t_{up}^p \\ &= f\left(l, \rho_i^{vf}\right) + \frac{1}{\mu_s} + 2 \times d_{r_i \to pfog}. \end{aligned} \tag{3}$$

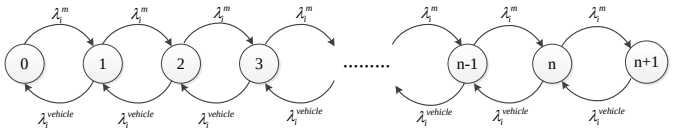


Fig. 2: Markov state for moving vehicle-based fog model.

2) Moving vehicle-based fog model: For the moving vehicles, we first provide the following Lemma:

Lemma 1. *Given the flow of vehicles arriving at RSU r_i with arrival rate $\lambda_i^{vehicle}$, and the flow of messages arriving at RSU r_i with arrival rate λ_i $\left(\lambda_i < \lambda_i^{vehicle}\right)$, the moving vehicle-based fog nodes can be modeled as an $M/M/1$ queueing system.*

Since the message flow arriving at RSU r_i is with arrival rate λ_i, message flows in the waiting queue of moving vehicle-based fog nodes can be viewed as a subprocess of the message flow arriving at RSU r_i, with arrival rate λ_i^m $(\lambda_i^m \leq \lambda_i)$. Correspondingly, $\lambda_i^m < \lambda_i^{vehicle}$ holds. In addition, we assume that all the moving vehicle-based fog nodes have the same resources and computational capacities. When a moving vehicle comes into the wireless communication range of an RSU, it picks up a message at the front of the waiting queue, and can finish message processing before leaving the communication range. The message flow in the waiting queue of moving vehicle-based fog nodes is a stochastic process, denoted as $\{X_n, n \geq 0\}$. We assume that it has status: $i_0, i_1, ..., i_{n+1}$, indicating the total number of messages in the waiting queue, and $P\{X_0 = i_0, X_1 = i_1, ..., X_n = i_n\} > 0$ holds. We can observe that the number of messages in the waiting queue at time $t+1$ has no relationship with the status at the time before t, that is $P\{X_{n+1} = i_{n+1} \mid X_0 = i_0, ..., X_n = i_n\} = P\{X_{n+1} \mid X_n = i_n\}$. As a result, the message flow in the waiting queue of moving vehicle-based fog nodes can be modeled as a Markov chain. The state space diagram for this chain is shown in Fig. 2, and the transition rate matrix is:

$$\begin{pmatrix} -\lambda_i^m & \lambda_i^m & & \\ \lambda_i^{vehicle} & -\left(\lambda_i^{vehicle} + \lambda_i^m\right) & \lambda_i^m & \cdots \\ & \lambda_i^{vehicle} & -\left(\lambda_i^{vehicle} + \lambda_i^m\right) & \\ & & \lambda_i^{vehicle} & \\ \cdots & & & \end{pmatrix}. \tag{4}$$

From the above analysis, we can observe that the Markov state and transition rate matrix for moving vehicle-based fog model is the same as the $M/M/1$ queueing system, where the transition rate matrix of $M/M/1$ queueing system is:

$$\begin{pmatrix} -\lambda & \lambda & & & \\ \mu & -(\mu+\lambda) & \lambda & & \\ & \mu & -(\mu+\lambda) & \lambda & \\ & & \mu & -(\mu+\lambda) & \lambda \\ & \cdots & & & \end{pmatrix}. \tag{5}$$

Therefore, we can regard the model of moving vehicle-based fog nodes as an $M/M/1$ queueing system. That is, the message flow entering the moving vehicle-based fog system follows a Poisson process with arrival rate λ_i^m. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2816590, IEEE Transactions on Industrial Informatics

moving vehicle flow can be viewed as a fixed server with process rate $\lambda_i^{vehicle}$, and the service rate is computed by $\rho_i^{mf} = \lambda_i^m / \lambda_i^{vehicle}$.

THEOREM 1. *Based on **Lemma 1**, we can observe that the average response time for moving vehicle-based fog nodes is directly related to message arrival rate λ_i^m and moving-vehicle arrival rate $\lambda_i^{vehicle}$, which can be computed by:*

$$\begin{aligned} E(t_{stol}^m) &= E(t_{que}^m) + E\left(t_{ser}^m\right) + 2 \times t_{up}^m \\ &= \frac{\rho_i^{mf}}{\lambda_i^{vehicle}\left(1-\rho_i^{mf}\right)} + \frac{1}{\lambda_i^{vehicle}} \\ &+ 2 \times d_{r_i \to mfog}. \\ &= \frac{1}{\lambda_i^{vehicle} - \lambda_i^m} + 2 \times d_{r_i \to mfog}. \end{aligned} \tag{6}$$

D. Offloading Model

In order to balance message flows in fog and cloudlet layers, we consider that all the RSUs are reachable from each other, which is similar to the assumption in [26]. That is, an RSU can redirect part of its flows to other RSUs, since message flows arriving at different RSUs may be significantly different. We define $g\left(i,k\right)$ as the amount of message flows redirected from RSUs r_i to r_k. Then, the following constrains based on $g\left(i,k\right)$ should be satisfied:

$$g\left(i,k\right) = \begin{cases} -g\left(k,i\right), & i \neq k \\ 0, & \text{otherwise.} \end{cases}, \tag{7}$$

$$\sum_{i=1}^{u}\sum_{k=1}^{u} g\left(i,k\right) = 0, \tag{8}$$

$$\sum_{k=1}^{u} \max\left\{g\left(i,k\right), 0\right\} \leq \lambda_i, \tag{9}$$

where $i,k \in \{1,2,...,u\}$. For simplicity and without loss of generality, we assume that the size of offloaded messages is equal, and the delay incurred by transferring a message between a pair of RSUs via the network is identical. We use d_{r_i,r_k} to represent the message transformation delay from RSUs r_i to r_k. If the redirected message flow satisfies $g\left(i,k\right) < 0$ for r_i, delay $-g\left(i,k\right) \times d_{r_i,r_k}$ exists. Therefore, the total delay $t_{i,offload}^j$ caused by incoming message flows from other RSUs in time slot j is illustrated as follows:

$$t_{i,offload}^j = \sum_{k=1}^{u} |\max\left\{g\left(i,k\right), 0\right\} \times d_{r_i,r_k}|. \tag{10}$$

In addition, the final incoming message flow at RSU r_i can be computed by:

$$\overline{\lambda}_i = \lambda_i - \sum_{i=1}^{u} g\left(i,k\right). \tag{11}$$

E. Problem Formulation

When a vehicle detects or comes across an accident on its route, it can record this event in forms of texts, pictures or even short videos. Then, the record is packaged into a message to wait for uploading toward TMS. After TMS gets accurate information of the event, it will immediately take actions and broadcast feedback messages through RSUs. Traditional TMS is responsible for processing all the messages in the system, which may cause a large delay and excessive resource consumption. In order to reduce the response time of TMS, we offload network traffic from TMS to cloudlets and fog nodes.

Specifically, we divide the whole city map into several regions. In each region, a cloudlet is available. Parked and moving vehicles close to an RSU can be potential fog nodes. The response time for a message m_i in time slot j can be calculated by:

$$\begin{aligned} E\left(t_{i,stol}^j\right) &= \alpha E(t_{stol}^c) + \beta E(t_{stol}^p) + \gamma E(t_{stol}^m) \\ &+ t_{i,offload}, \end{aligned} \tag{12}$$

where $\alpha + \beta + \gamma = 1$, and

$$\alpha = \begin{cases} 1, & \text{if the message is processed by the cloudlet,} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

$$\beta = \begin{cases} 1, & \text{if the message is processed by the parked} \\ & \text{vehicle-based fog nodes,} \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

$$\gamma = \begin{cases} 1, & \text{if the message is processed by the moving} \\ & \text{vehicle-based fog nodes,} \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

Therefore, the offloading problem in IoV systems is defined as follows: given system model parameters $(G, \mu_s, d_{r_i \to cloudlet}, d_{r_i \to pfog}, d_{r_i \to mfog}, d_{r_i,r_k}\lambda_1, ...\lambda_m, l, \lambda_i^{vehicle})$, the offloading problem is to find available λ_i^c, λ_i^p, λ_i^m and b such that the average response time $E(t_{stol})$ can be minimized, i.e.,

$$\min_{\lambda_i^c, \lambda_i^p, \lambda_i^m, b} \frac{1}{su} \sum_{j=1}^{s}\sum_{i=1}^{u} E\left(t_{i,stol}^j\right), \tag{16}$$

$$\textbf{s.t.} \begin{cases} Equations(7) - (9), \\ \lambda_i^c + \lambda_i^p + \lambda_i^m \leq \overline{\lambda}_i, \\ \lambda^c = \sum_{i=1}^{k} \lambda_i^c, \\ 0 < \frac{\lambda^c}{b\mu_s}, \frac{\lambda_i^p}{l\mu_s}, \frac{\lambda_i^m}{\lambda_i^{vehicle}} < 1, \\ \alpha + \beta + \gamma = 1, \alpha, \beta, \gamma \in \{0,1\}. \end{cases} \tag{17}$$

The main notations in this study are listed in Table I.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2816590, IEEE Transactions on Industrial Informatics

TABLE I: Important notations

Notation	Description
$\lambda_i^{vehicle}$	Arrival rate of vehicle flow arriving at RSU r_i
λ_i	Arrival rate of message flow arriving at RSU r_i
$\lambda^c, \lambda_i^p, \lambda_i^m$	Arrival rates of message flows arriving at the cloudlet, parked and moving vehicle-based fog nodes, respectively
$t_{up}^c, t_{up}^p, t_{up}^m$	Uploading time for a message from RSU r_i to the cloudlet, parked and moving-vehicle fog nodes, respectively
ρ^c, ρ_i^{vf}	Service rates for the cloudlet and parked vehicle-based fog nodes, respectively
$E\left(t_{stol}^c\right), E\left(t_{stol}^p\right), E\left(t_{stol}^m\right)$	Response time for a message processed by the cloudlet, parked and moving vehicle-based fog nodes, respectively
$d_{r_i \to cloudlet}$	Network delay from RSU r_i to the cloudlet
$d_{r_i \to pfog}$	Network delay from RSU r_i to parked vehicle-based fog nodes
$d_{r_i \to mfog}$	Network delay from RSU r_i to moving vehicle-based fog nodes
$g(i,k)$	Amount of message flows redirected from RSUs r_i to r_k
d_{r_i,r_k}	Transformation delay for a message from RSUs r_i to r_k
$t_{i,offload}$	Total delay caused by incoming message flows from other RSUs
$\bar{t}^{fog}$	Average response time for a message in a region
V_u, V_o	The sets of unloaded and overloaded fog units, respectively
ϕ_i, ϕ_k	The amount of outcoming and incoming message flows for each fog unit, respectively
ϕ_{in}, ϕ_{out}	Total incoming message flows to unloaded fog units and output message flows from overloaded fog units, respectively

IV. Offloading Algorithm for Response Time Minimization

It is noteworthy that the formulated problem in Section III-E is affected by different factors, and variables in different subproblem are tightly coupled with each other. Thus, the tradeoff between the response time and the message flow allocation is complex. To address this issue, we develop an approximate approach including two subsystem optimization methods to make the offloading problem solvable. After that, the whole offloading algorithm is presented.

A. Sub-optimization Problem Formulation

Based on Equations (16)-(17), we can observe that the offloading problem is a Mixed Integer Nonlinear Programming (MINLP) problem, which is difficult to solve. Since the objective is to minimize the average response time for all time slots, it is equivalent to minimizing the average response time in each time slot. Since the status of parked vehicles in each time slot has no significant change, it is promising to solve the problem in each time slot and approach to the global objective further. Correspondingly, the offloading problem in time slot j is defined as follows:

$$\min_{\lambda_i^c, \lambda_i^p, \lambda_i^m, b} \frac{1}{u} \sum_{i=1}^{u} E\left(t_{i,stol}^j\right), \tag{18}$$

s.t. Equation group (17).

Obviously, the cloudlet, parked and moving vehicle-based fog nodes have significant impacts on the system performance. Since the infrastructure-based cloudlet is generally fixed, the performance mainly depends on its processing ability and the arrival rate of incoming message flows. However, parked and moving vehicle-based fog nodes are not stable, and their locations may change as time goes by. Especially, the locations of moving vehicle-based fog nodes vary a lot even during a short period, making the network topology change dynamically. Therefore, the processing ability of fog nodes is the main factor to affect the average response time. Correspondingly, we deal with the offloading problem by two subsystems, i.e., the cloudlet and fog node-based subsystems. In the fog node-based system, we first compute the expected minimum response time for these two kinds of fog nodes. Then, we redirect part of message flows from overloaded fog units to unloaded ones, making the average system response time approach to the expected minimum response time. Based on the configuration of fog nodes, we finally determine the number of servers required on the cloudlet and the arrival rate of incoming message flow.

B. Delay Minimization for Fog Nodes

First, a set of fog nodes, including parked and moving vehicle-based fog nodes at each RSU, is defined as a fog unit, i.e., there are u fog units in a region. We first compute the expected minimum response time $\bar{t}^{fog}$ in a region by solving a concave cost network flow problem. Then, the specific flow of message streams redirected from overloaded fog units to underloaded ones can be decided. As a result, the average response time can approach to the expected minimum response time.

1) Average response time: In order to obtain the expected minimum response time, we first ignore the transmission delay caused by message flow redirection, and only consider the response delay for each fog unit. Then, we regard the total message flow arriving in the system as $\sum_{i=1}^{u} \lambda_i$. Since we need to assign the total message flows to each kind of fog nodes in all fog units, the following objective should be satisfied:

$$\min \sum_{i=1}^{u} \left(E(t_{stol}^p) + E(t_{stol}^m)\right), \tag{19}$$

$$\textbf{s.t.} \begin{cases} \sum_{i=1}^{u} (\lambda_i^p + \lambda_i^m) \le \sum_{i=1}^{u} \lambda_i, \\ 0 < \lambda_i^p < l\mu_s, \\ 0 < \lambda_i^m < \lambda_i^{vehicle}. \end{cases} \tag{20}$$

Because the item $E(t_{stol}^p) + E(t_{stol}^m)$ in Equation (19) is a general nonlinear function (neither convex nor concave) with variables λ_i^p and λ_i^m $(i = 1, 2, ..., u)$, we first employ the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2816590, IEEE Transactions on Industrial Informatics

Algorithm 1 Pseudo-code of brand-and-bound algorithm based delay minimization scheme

Input: G, $costs$, $capacity$
Output: $flows^*$, $minDelay^*$

```
Initial U = N = L = ∅, i = 0
Add a super source node s and a super destination node t in G to form G'
Update the costs for borders in G'
Conduct a bipartite graph M based on G'
while i <= n do
    if U = 0 then
        i = i + 1
    else
        Find u* ∈ U and N, L
        u = u*, v = 2u, delete u from U, L+ and N+
    end if
    Set x_v = 0, and find the minimum cost flow based on Algorithm 2
    Compute LB(L) and UB(L)
    if LB(L) >= UB then
        Continue;
    end if
    if UB(L) < UB then
        UB = UB(L)
    end if
end while
```

method in [27], which can convert the general nolinear cost function to the concave nolinear function. Then, it becomes a minimum concave-cost network flow problem [28]. We apply a well-studied brand-and-bound algorithm in [29], which is suitable for minimum cost flow computing with concave cost functions. After obtaining λ_i^p and λ_i^m, we can obtain expected minimum response time $\bar{t}^{fog}$. The pseudo-code of brand-and-bound algorithm based delay minimization scheme is presented in Algorithm 1.

2) Redirected flow: We first acquire the amount of incoming and output message flows for all fog units. Then, we form the redirected problem for minimizing the average response time as a typical linear minimum cost network flow problem, which can be solved by existing methods, e.g., Edmonds-Karp algorithm [30], in polynomial time.

We define processing ratio σ as the ratio of message flows processed by fog nodes compared to the total flows in the system, i.e.,

$$\sigma = \frac{\sum_{i=1}^{u}(\lambda_i^p + \lambda_i^m)}{\sum_{i=1}^{u}\lambda_i}. \tag{21}$$

In addition, we divide all the fog units into two separate sets. One is overloaded set $V_o = \{i|\lambda_i^p + \lambda_i^m > \lambda_i\}$, and the other is unloaded set $V_u = \{k|\lambda_k^p + \lambda_k^m \leq \lambda_k\}$.

For each overloaded fog unit $i \in V_o$, the amount of message flows that can be offloaded to other fog units is defined as ϕ_i. It should be determined, so as to make the average response time approach to the expected minimum response time. For example, the message flow from overloaded fog unit i to unloaded fog unit k is shown in Fig. 3. Based on the definition of σ, we can obtain $\phi_i = \lambda_i \times \sigma - \lambda_i^p - \lambda_i^m$, where $\phi_i > 0$. For each unloaded fog unit $k \in V_u$, the amount of message flows allowing to arrive at k is denoted as ϕ_k and determined by $\phi_k = \lambda_k^p + \lambda_k^m - \lambda_k \times \sigma$, where $\phi_k > 0$. Then, the total amount of incoming message flows to unloaded fog units is $\phi_{in} = \sum_{k \in V_u} \phi_k$, while the output message flow to overloaded fog units is $\phi_{out} = \sum_{i \in V_o} \phi_i$.

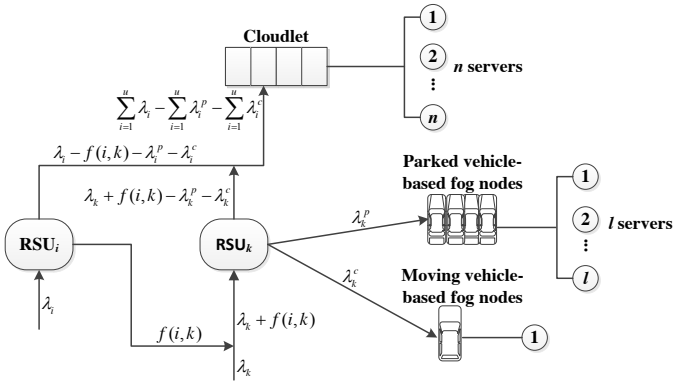


Fig. 3: Message flow from RSU i to RSU k.

Therefore, given the values of ϕ_i $(i \in V_o)$ and ϕ_k $(k \in V_u)$, the redirection problem becomes to minimize the transmission delay by redirecting message flows from overloaded fog units to unloaded fog units, where the objective is:

$$\min \sum_{i \in V_o} \sum_{k \in V_u} g(i,k) \times d_{r_i, r_k}, \tag{22}$$

$$\textbf{s.t.} \begin{cases} \sum_{i \in V_o} g(i,k) = \phi_k, k \in V_u, \\ \sum_{k \in V_u} g(i,k) = \phi_i, i \in V_o, \\ g(i,k) \geq 0. \end{cases} \tag{23}$$

This is a typical linear minimum cost network flow problem, and can be solved by the Edmonds-Karp algorithm. The pseudo-code of the Edmonds-Karp algorithm based delay minimization scheme is presented in Algorithm 2. Then, the minimum transmission delay for redirecting message flows from the overloaded set to the unloaded set can be obtained. By the integration of minimum transmission delay and the expected minimum response time, the optimization objective can be reached.

C. Delay Minimization for Cloudlet

Since vehicle-based fog nodes are not stable enough to cope with message flows and their locations change as time goes by, cloudlets are necessary in the system. Cloudlets act as supplemental components to process message flows that cannot be handled by fog nodes, with the advantage of maintaining system performance stability. If there are sufficient fog nodes, the cloudlet is in idle. Therefore, it is desirable to minimize the number of servers on the cloudlet to manage message flows

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2816590, IEEE Transactions on Industrial Informatics

Algorithm 2 Pseudo-code of Edmonds-Karp algorithm based delay minimization scheme

Input: G, $costs$, $capacity$
Output: $flows^*$, $minDelay^*$

```
Add a super source node s and a super destination node t in G to form G'
while i ∈ V_o do
    capacity[s][i] = φ_i
    costs[s][i] = 0
end while
while k ∈ V_u do
    capacity[k][t] = φ_k
    costs[k][t] = 0
end while
for each edge e[i][j] in G' do
    flows[i][j]=0;
end for
while find a shortest route p from s to t in G' do
    m = min(capacity[i][j]|e[i][j] ∈ p)
    for each edge e[i][j] in p do
        if m <= capacity[i][j] then
            capacity[j][i] = capacity[j][i] + m
            flow[i][j] = flow[i][j] + m
        end if
    end for
    minDelay = flows × cost
end while
```

unhandled by fog nodes, not only reducing the installation cost but also improving the resource utilization ratio.

In time slot s_j, message flows to be handled by the cloudlet in a region can be calculated by $\lambda^c = \sum_{i=1}^{u} \lambda_i - \sum_{i=1}^{u} \lambda_i^m - \sum_{i=1}^{u} \lambda_i^p$. If $\lambda^c \leq 0$, it means that fog nodes are capable to handle all message flows in time slot s_j. Otherwise, the cloudlet has to handle incoming message flows. As the number of parked and moving vehicles changes in different time slots, it is necessary to obtain the number of servers on the cloudlet that can manage message flows during the whole communication process. Then, we deduce the minimum number of servers on cloudlet b^j by substituting parameter λ^c into Equation (2) and making $E(t_{stol}^c) = \bar{t}^{fog}$. For the whole communication process, we define the minimum number of servers on the cloudlet as $b = \max\{b_j | j = \{1, 2, ..., s\}\}$. Hence, we can set up servers on the cloudlet, which can achieve a high resource utilization ratio and with a low installation cost.

D. Offloading Algorithm

We design the offloading algorithm in IoV-based traffic management systems based on optimization methods. Our principle is to utilize fog nodes first, since they are closer to RSUs than the cloudlet. If message flows cannot be processed by fog nodes, RSUs will direct rest flows to the cloudlet. The number of parked vehicles does not have significant changes in each time slot and the flow of moving vehicles follows a Poisson process with arrival rate $\lambda_i^{vehicle}$ at RSU r_i. In addition, the parked and moving vehicle-based fog nodes at RSU r_i form a fog unit. We first employ Equations (19)-(20) to calculate expected minimum response time $\bar{t}_i^{fog}$ of fog nodes within a region. It is a minimum concave-cost network flow problem, and can be solved by a brand-and-bound algorithm. After obtaining $\bar{t}^{fog}$, we can determine the redirected message flows from overloaded fog units to unloaded ones by solving the optimization problem according to Equations (22)-(23). It is a typical linear minimum cost network flow problem, and can be solved by the Edmonds-Karp algorithm. Finally, we can obtain message flows λ_i^p and λ_i^m arriving at parked and moving vehicle-based fog nodes respectively in each fog unit. Message flows transmitted to the cloudlet can be computed based on λ_i^p and λ_i^m, so that the desired number of servers on the cloudlet can be acquired. If we install servers on the cloudlet according to the acquired number, the minimum install cost and maximum resource utilization can be achieved.

THEOREM 2. *The time complexity of the proposed FORT algorithm is $O(m^4)$, where m is the number of borders in the network graph for the brand-and-bound algorithm and $m < n$ holds.*

Proof: The time complexity of FORT mainly depends on the two referred optimization algorithms, i.e., the Edmonds-Karp and the brand-and-bound based delay minimization scheme. For the first one, the time complexity is $O(nm^2)$ [30], where n is the number of RSUs in the network, and m is the number of edges. For the second one, there are at most $m-n+1$ branches for computation. In each branch, Edmonds-Karp algorithm is conducted to find the current maximum flow in a bipartite graph. Therefore, the time complexity of our presented FORT is $O((m+n) \times (2m)^2 \times (m-n+1))$, where $m+n$ is the number of nodes and $2m$ is the number of arcs in the bipartite graph, i.e., the total time complexity is $O(m^4)$.

V. Performance Evaluation

In order to validate the network performance of FORT, we conduct extensive simulations based on the map of Shanghai (China) and real-world traces of taxies in Shanghai (China) by Monte Carlo method.

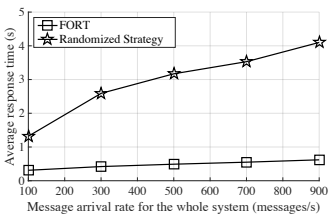
A. Simulation Setup

TABLE II: Selected GPS locations in Shanghai

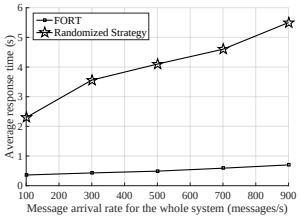
Jingan district		Hongkou district	
ID	Locations	ID	Locations
1	31.228587,121.455745	1	31.304725,121.473037
2	31.230982,121.456374	2	31.294622,121.470535
3	31.234170,121.431700	3	31.276524,121.491138
4	31.223078,121.446153	4	31.276615,121.477732
		5	31.273372,121.487506
		6	31.263758,121.486227

To demonstrate the feasibility of FORT, we consider a realistic scenario by using the map of Shanghai and the real traces of taxies collected from April 1, 2015 to April 30, 2015, including the recorded information of more than 1000 taxies in Shanghai. The data set contains the corresponding information, e.g., GPS locations, record time, speed and directions. Specifically, we divide Shanghai into seven regions according

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2816590, IEEE Transactions on Industrial Informatics

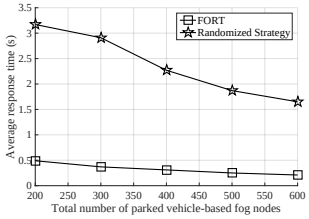


(a) Jingan district

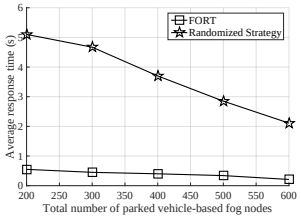


(b) Hongkou district

Fig. 4: Average response time with different message arrival rates for the whole system.



(a) Jingan district



(b) Hongkou district

Fig. 5: Average response time based on the total number of parked vehicle-based fog nodes.

to its administrative divisions [31]. Within each region, it has a cloudlet and several RSUs. Taking Hongkou and Jingan districts as examples, we select several centers of sub-districts to place RSUs, as illustrated in Table II. In addition, we use Google map to measure distances between each two locations. For each location, we do statistical analysis on the arrival rate of moving vehicles in the range of 500m for every 10 minutes. Similar to [32], we assume that the network delay between each pair of RSUs is determined by the normal distribution of $0.1 \leq N\left(0.15, 0.05\right) \leq 0.2$.

Since our work is a priori attempt to provide a detailed design of fog computing in real-time traffic management systems, related algorithms are not available for comparison to our best knowledge. Correspondingly, we consider the following situation, which is named as "Randomized Strategy". Details are illustrated as follows: when the message flow arrives at an RSU, it randomly directs messages to moving and parked vehicle-based fog nodes for processing. The objective of this strategy is to maximize workloads processed by parked and moving vehicle-based fog nodes, i.e.,

$$\min \sum_{i=1}^{u} \left(\lambda_i^p + \lambda_i^m\right), \tag{24}$$

$$\textbf{s.t.} \begin{cases} \lambda_i^p + \lambda_i^m \leq \lambda_i, \\ 0 < \lambda_i^p < l\mu_s, \\ 0 < \lambda_i^m < \lambda_i^{vehicle}. \end{cases} \tag{25}$$

It is an Integral Linear Programming (ILP) problem. The resident message flows not processed by fog nodes will be uploaded to the cloudlet for processing.

B. Simulation Results

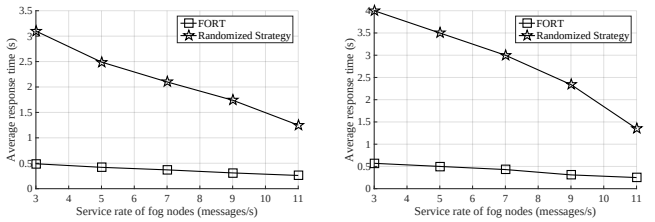
Fig. 4 illustrates the performance of average response time with different message arrival rates for the whole system. Herein, the message arrival rate for the whole system is defined as $\sum_{i=1}^{u} \lambda_i$, based on which we randomly generate the message arrival rate for each RSU. We notice that the average response time of Randomized Strategy is higher than that of FORT. When the message arrival rate increases, the average response time for Randomized Strategy sharply raises, while that of FORT gently goes up. For example, when the message arrival rate is about 300 messages/s, the average response time of Randomized Strategy is 2.59s while that of FORT is 0.42s in Fig. 4(a). The reason is that, FORT obtains the minimum average response time by two subsystems. When the message arrival rate becomes large, the system can balance message loads among these RSUs. However, the Randomized Strategy intends to maximize loads of fog nodes, without a load balancing method to minimize the average response time. Similar results can be obtained in Fig. 4(b).

The performances of average response time based on the total number of parked vehicle-based fog nodes are illustrated in Fig. 5. We define the total number of parked vehicle-based fog nodes by $\sum_{i=1}^{u} l$. After specifying the total number of parked vehicle-based fog nodes, we randomly generate the parked vehicle-based fog nodes at each RSU, making the total number equal to $\sum_{i=1}^{u} l$. From Fig. 5(a), we can find that the performance on average response time of FORT is better than that of Randomized Strategy, and they both drop when the number of parked vehicle-based fog nodes becomes large. This is because more parked vehicle-based fog nodes accompany with more powerful processing abilities. Therefore, more messages can be handled simultaneously, which can largely shorten the average response time.

The results of average response time based on the distinct service rate of fog nodes are presented in Fig. 6. When the service rate of fog nodes increases, the processing ability of parked vehicle-based fog nodes becomes strong. As a result, fog nodes can process more messages in parallel. In Fig. 6(a), it is obvious that the average response time of Randomized Strategy and FORT drops when the service rate increases. For instance, when the service rate of fog nodes is 5 messages/s in Fig. 6(a), the average response time of Randomized Strategy and FORT are 2.49 and 0.42s, respectively. When the service rate of fog nodes is 9 messages/s, the average response time for Randomized Strategy drops 30%, while that of FORT decreases 26%. Similar results can also be obtained in Hongkou district, as shown in Fig. 6(b). The performance of FORT is better than that of Randomized Strategy. This is because it makes the average response time of the system approach to the minimum average response time, while Randomized Strategy focuses on maximizing workloads of fog nodes.

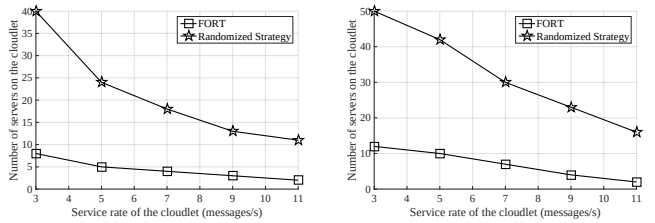
Fig. 7 illustrates the number of servers on the cloudlet with different service rates of the cloudlet. It shows that the number of servers on the cloudlet drops when the service rate increases. For example, when the service rate is 5 messages/s, the number of required servers on the cloudlet for FORT is 4,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2816590, IEEE Transactions on Industrial Informatics



(a) Jingan district (b) Hongkou district

Fig. 6: Average response time with different service rates of fog nodes.

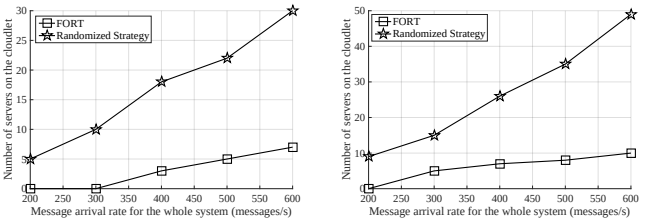


(a) Jingan district (b) Hongkou district

Fig. 7: The number of servers on the cloudlet with different service rates of cloudlet.

while that of Randomized Strategy is 22. When the service rate is 9 messages/s, the number of required servers on the cloudlet for FORT is 2, while that of Randomized Strategy is 13 in Fig. 7(a). In addition, the required number of servers on the cloudlet for FORT is much smaller than that of Randomized Strategy. The reason is that, FORT mainly leverages fog nodes to process messages and balance loads among fog units, so that only the messages not handled by fog nodes will be uploaded to the cloudlet for processing. Our method can largely reduce the incoming message flows for the cloudlet. Nevertheless, Randomized Strategy uploads messages to the cloudlet when they cannot be processed in their fog units.

The number of servers on the cloudlet with different message arrival rates is illustrated in Fig. 8. We can observe that the number of required servers on the cloudlet of Randomized Strategy is much larger than that of FORT. For example, when the message arrival rate is 300 messages/s, the required server number of Randomized Strategy is 10, while that of FORT is 0 in Fig. 8(a). When the message arrival rate is 700 messages/s, the required number of servers of Randomized Strategy is 22, while that of FORT is merely 5. The reason is that the system cannot process more messages, if the message arrival rate becomes larger. For FORT, a load balancing algorithm is conducted to balance workloads among all the fog units. This means fog nodes are able to process more messages. However, Randomized Strategy only processes message flows in their own fog units. If a fog unit is overloaded, it will upload messages to the cloudlet for processing, which aggravates burdens of the cloudlet.



(a) Jingan district (b) Hongkou district

Fig. 8: The number of servers on the cloudlet with different message arrival rates.

VI. Conclusion

In this paper, we propose a feasible solution that enables offloading for real-time traffic management in fog-based IoV systems, with the purpose of minimizing the average system response time. We first model parked and moving vehicle-based fog nodes by queueing theory, and then mathematically formulate an optimization problem for the fog-enabled offloading problem. Specially, we draw the conclusion that moving vehicle-based fog nodes can be modeled as an $M/M/1$ queue. Then, an offloading optimization problem is formulated. An approximate approach is developed to solve the formulated problem by scheduling the message flow allocation among different fog nodes. At last, real-world traces of taxies in Shanghai are utilized to demonstrate the superiority and effectiveness of our presented FORT method. In our future work, we will consider how to utilize vehicles outside the communication ranges of RSUs as fog nodes to offload loads for TMS.

Acknowledgments

The work is supported by National Natural Science Foundation of China with No. 61733002 and No. 61502075, the Fundamental Research Funds for the Central University with No. DUT17LAB16, No. DUT2017TB02 and No. DUT17RC(4)49, Tianjin Key Laboratory of Advanced Networking (TANK), School of Computer Science and Technology, Tianjin University, Tianjin, China, 300350.

References

[1] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[2] Z. Ning, X. Hu, Z. Chen, M. Zhou, B. Hu, J. Cheng, and M. S. Obaidat, "A cooperative quality-aware service access system for social Internet of vehicles," *IEEE Internet of Things Journal, Doi: 10.1109/JIOT.2017.2764259*, 2017.

[3] J. He, Y. Ni, L. Cai, J. Pan, and C. Chen, "Optimal dropbox deployment algorithm for data dissemination in vehicular networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 632–645, 2018.

[4] C. Zhu, L. Shu, V. C. M. Leung, S. Guo, Y. Zhang, and L. T. Yang, "Secure multimedia big data in trust-assisted sensor-cloud for smart city," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 24–30, 2017.

[5] W. Li, C. Zhu, V. C. M. Leung, L. T. Yang, and Y. Ma, "Performance comparison of cognitive radio sensor networks for industrial IoT with different deployment patterns," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1456–1466, 2017.

[6] Z. Ning, X. Wang, X. Kong, and W. Hou, "A social-aware group formation framework for information diffusion in narrow-band Internet of things," *IEEE Internet of Things Journal, Doi: 10.1109/JIOT.2017.2777480*, 2017.

[7] C. Zhu, V. C. M. Leung, L. Shu, and C. H. Ngai, "Green Internet of things for smart world," *IEEE Access*, vol. 3, pp. 2151–2162, 2015.

[8] C. Zhu, J. J. P. C. Rodrigues, V. C. M. Leung, L. Shu, and L. T. Yang, "Trust-based communication for industrial Internet of things," *IEEE Communications Magazine*, 2018.

[9] W. Hou, Z. Ning, and L. Guo, "Green survivable collaborative edge computing in smart cities," *IEEE Transactions on Industrial Informatics, DOI: 10.1109/TII.2018.2797922*, 2018.

[10] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog computing for Internet of things applications: Challenges and solutions," *IEEE Communications Surveys & Tutorials, Doi:10.1109/COMST.2017.2762345*, 2017.

[11] Z. Cao, S. Jiang, J. Zhang, and H. Guo, "A unified framework for vehicle rerouting and traffic light control to reduce traffic congestion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1958–1973, 2017.

[12] M. Wang, H. Shan, R. Lu, R. Zhang, X. Shen, and F. Bai, "Real-time path planning based on hybrid-VANET-enhanced transportation system," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 5, pp. 1664–1678, 2015.

[13] J. Ahn, Y. Wang, B. Yu, F. Bai, and B. Krishnamachari, "RISA: Distributed road information sharing architecture," in *IEEE INFOCOM*, pp. 1494–1502, 2012.

[14] R. Aissaoui, H. Menouar, A. Dhraief, F. Filali, A. Belghith, and A. Abu-Dayya, "Advanced real-time traffic monitoring system based on V2X communications," in *IEEE International Conference on Communications*, pp. 2713–2718, 2014.

[15] W. Zhang, Z. Zhang, and H.-C. Chao, "Cooperative fog computing for dealing with big data in the Internet of vehicles: Architecture and hierarchical resource management," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 60–67, 2017.

[16] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading," *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 36–44, 2017.

[17] X. Chen and L. Wang, "Exploring fog computing-based adaptive vehicular data scheduling policies through a compositional formal method-PEPA," *IEEE Communications Letters*, vol. 21, no. 4, pp. 745–748, 2017.

[18] Y. Liu, C. Xu, Y. Zhan, Z. Liu, J. Guan, and H. Zhang, "Incentive mechanism for computation offloading using edge computing: A stackelberg game approach," *Computer Networks*, vol. 129, no. 2, pp. 399–409, 2017.

[19] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.

[20] Z. Ning, F. Xia, N. Ullah, X. Kong, and X. Hu, "Vehicular social networks: Enabling smart mobility," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 16–55, 2017.

[21] H. Zhang, Q. Zhang, and X. Du, "Toward vehicle-assisted cloud computing for smartphones," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5610–5618, 2015.

[22] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, "Energy-latency trade-off for energy-aware offloading in mobile edge computing networks," *IEEE Internet of Things Journal, Doi:10.1109/JIOT.2017.2786343*, 2017.

[23] J. He, L. Cai, P. Cheng, and J. Pan, "Delay minimization for data dissemination in large-scale VANETs with buses and taxis," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 1939–1950, 2016.

[24] H. Zhu, M. Li, and L. Fu, "Impact of traffic influxes: Revealing exponential intercontact time in urban VANETs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, pp. 1258–1266, 2011.

[25] L. Klennrock, "Queueing systems volume 1: Theory," *New York*, 1975.

[26] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Reinforcement learning for resource provisioning in the vehicular cloud," *IEEE Wireless Communications*, vol. 23, no. 4, pp. 128–135, 2016.

[27] B. W. Lamar, "A method for solving network flow problems with general nonlinear arc costs," in *Network Optimization Problems: Algorithms, Applications and Complexity*, pp. 147–167, World Scientific, 1993.

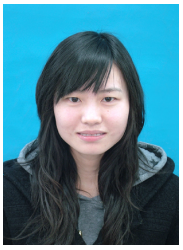
[28] G. M. Guisewite and P. M. Pardalos, "Minimum concave-cost network flow problems: Applications, complexity, and algorithms," *Annals of Operations Research*, vol. 25, no. 1, pp. 75–99, 1990.

[29] M. Florian and P. Robillard, "An implicit enumeration algorithm for the concave cost network flow problem," *Management Science*, vol. 18, no. 3, pp. 184–193, 1971.

[30] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of the ACM*, vol. 19, no. 2, pp. 248–264, 1972.

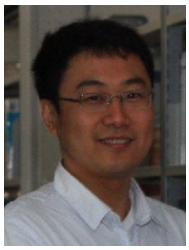
[31] "Administrative divisions of shanghai." https://en.wikipedia.org/wiki/List_of_administrative_divisions_of_Shanghai.

[32] M. Jia, W. Liang, Z. Xu, and M. Huang, "Cloudlet load balancing in wireless metropolitan area networks," in *IEEE INFOCOM*, pp. 1–9, 2016.



Xiaojie Wang received the M.S. degree from Northeastern University, China, in 2011. From 2011 to 2015, she was a software engineer in NeuSoft Corporation, China. Currently, she is pursuing the Ph.D. degree with the School of Software, Dalian University of Technology, Dalian, China. Her research interests are vehicular network and fog computing.

Zhaolong Ning (M'14) received the M.S. and PhD degrees from Northeastern University, Shenyang, China. He was a Research Fellow at Kyushu University, Japan. He is an associate professor in the School of Software, Dalian University of Technology, China. He has been serving as an Associate Editor of IEEE Access, the lead guest editors of The Computer Journal, IEEE Access, etc. His research interests include fog computing, vehicular network, and network optimization.



Lei Wang (M'01) is currently a full professor of the School of Software, Dalian University of Technology, China. He received his B.S., M.S. and Ph.D. from Tianjin University, China, in 1995, 1998, and 2001, respectively. He was a Member of Technical Staff with Bell Labs Research China (2001-2004), a senior researcher with Samsung, South Korea (2004-2006), a research scientist with Seoul National University (2006-2007), and a research associate with Washington State University, Vancouver, WA, USA (2007-2008). His research interests involve wireless ad hoc network, sensor network, social network and network security. He has published 90+ papers and his papers have received 1200+ citations.