Vehicular Fog Computing: Enabling Real-Time Traffic Management for Smart Cities

Zhaolong Ning, Jun Huang, and Xiaojie Wang

Abstract

Fog computing extends the facility of cloud computing from the center to edge networks. Although fog computing has the advantages of location awareness and low latency, the rising requirements of ubiquitous connectivity and ultra-low latency challenge real-time traffic management for smart cities. As an integration of fog computing and vehicular networks, vehicular fog computing (VFC) is promising to achieve real-time and location-aware network responses. Since the concept and use case of VFC are in the initial phase, this article first constructs a three-layer VFC model to enable distributed traffic management in order to minimize the response time of citywide events collected and reported by vehicles. Furthermore, the VFC-enabled offloading scheme is formulated as an optimization problem by leveraging moving and parked vehicles as fog nodes. A real-world taxi-trajectory-based performance analysis validates our model. Finally, some research challenges and open issues toward VFC-enabled traffic management are summarized and highlighted.

Introduction

The urban vehicular network is viewed as a core component of intelligent transportation systems, covering the regions of traffic safety, localization and navigation, high-efficiency information sharing and spread, and so on. On one hand, over 150 million cars on roads will be connected by 2020. Since each car generates on average 30 TB data a day, it challenges the ever-saturating wireless bandwidth [1]. On the other hand, the increasing number of vehicles on roads is promising to alleviate the traffic burden of cellular networks via intelligent management. However, the traditional ad hoc vehicle-to-vehicle (V2V) communication pattern suffers from intermittent connectivity [2], making the quality of services and ultra-low latency requirements challenging. Although some solutions have been investigated to satisfy the communication and computational requirements of traffic management for smart cities, they are far from enough. For example, the bandwidth of cellular networks is limited and primarily controlled by network operators. The deployment of roadside units (RSUs) is costly, and it is impossible to fully cover all roads. Furthermore, mobile cloud computing is time consuming and expensive for real-time traffic uploading in vehicular networks. Therefore, real-time traffic management for smart cities calls for a novel platform.

Mobile traffic will reach 360 exabytes in 2020, which is eight times as large as that in 2015.[1] As an integration of vehicular networks and cloud computing, vehicular cloud computing (VCC) is aimed at employing network resources efficiently so that vehicles are not only resource consumers but also resource providers. The ever increasing services and applications in vehicular networks call for huge computing resources and real-time feedback, challenging the resource-limited vehicles and centralized traffic management mechanisms, especially during traffic peak times in a citywide area.

By facilitating the communication, computing, and networking close to end terminals, fog computing is flexible and highly efficient for optimizing network resources from a local viewpoint. The OpenFog consortium has released its reference architecture to construct an interoperable and scalable fog computing platform in 2017. As the traffic of vehicular services becomes overwhelming, fog nodes are likely to be swamped. One natural question is how to scale up the computing resource of fog nodes. Vehicular fog computing (VFC) makes use of vehicular resources to promote the computational capability as well as further lower the delay of fog computing. By VFC, unexploited computing resources of vehicles can be supplemented to act as components of fog nodes, such as vehicles in parking areas or shopping malls. Table 1 makes a comparison between VCC and VFC.

Moving vehicles have been leveraged to promote the processing ability of cloud computing for end terminals [3]. If the cloud is overloaded, available resources in vehicles can be scheduled for resource-consuming computing to reduce the process delay. The concept of VFC was presented in [4], utilizing both parked and moving vehicles as computational and communication infrastructures. A quantitative analysis of fog capacity was carried out, together with the relationship among the communication connectivity and capability, and vehicle mobility. However, it was merely a feasibility analysis. Security challenges in VFC were discussed, and a fog-assisted traffic control

[1] http://www.cisco.com/c/en/us/solutions/collateral/service- provider/visual-networking-indexvni/mobile-white-paper-c11-520862.html.

Zhaolong Ning and Xiaojie Wang are with Dalian University of Technology and Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province; Zhaolong Ning is also with Chongqing University of Posts and Telecommunications and Chongqing Key Laboratory of Mobile Communications Technology; Jun Huang is with Chongqing University of Posts and Telecommunications.

Digital Object Identifier: 10.1109/MWC.2019.1700441

1536-1284/19/$25.00 © 2019 IEEE

Features	VCC	VFC
Individual communication	Bandwidth constrained	Real-time load balancing
Burden on core network	High	Low
Computational capability	High	Low
Deployment cost	High	Low
Decision making	Centralized and remote	Distributed and local
Resource optimization	Global	Local
Latency	High	Low
Mobility management	Easy	Hard
Reliability	High	Low

TABLE 1. Comparison between VCC and VFC.

system was presented as a use case [1]. Different from the existing works, this article first constructs a three-layer VFC model to manage citywide traffic in a distributed manner. After that, we investigate a VFC-enabled offloading scheme for real-time traffic management. Specifically, the fog layer, by utilizing both parked and moving vehicles, is designed to offer computing resources for information processing. The effectiveness of our model is demonstrated by a real-world taxi-trajectory-analysis-based evaluation.

The rest of this article is organized as follows. The following section illustrates the three-layer architecture of VFC. After that, we put forward the VFC-enabled real-time traffic management scheme and demonstrate its effectiveness. Some research challenges and open issues for VFC are then discussed before concluding this article.

Architecture of VFC

Vehicles are viewed as the infrastructures in VFC-enabled architectures, whose objective is to take advantage of both fog service providers and vehicular communications. Figure 1 illustrates the three-layer VFC architecture, containing the cloud layer, cloudlet layer, and fog layer.

Cloud Layer: The cloud layer is generally constituted by a traffic management server (TMS) and a trusted third authority (TTA). It performs city-level monitoring and centralized remote control. Generally, the TMS is in charge of processing messages and informing traffic managers to take actions. If all the messages uploaded by vehicles are processed at the TMS side, it would be overloaded. Therefore, the TMS is merely in charge of result reception and reward allocation in our work. Individual rewards and network fairness are managed by the TTA.

Cloudlet Layer: The cloudlet layer receives data reported by vehicles, and processes the collected data before delivering them to the cloud layer. Since the contents generated by vehicles are always of local interest, the uploaded information related to road conditions merely interests vehicles inside or around a specific region. In our distributed traffic management scheme, we first separate a city into different regions. In the center of each region, a cloudlet is responsible for the management of uploaded messages by vehicles. This layer includes various network devices, including gateway, routers, and access points. Some RSUs may also exist to schedule message uploading and manage the messages generated by vehicles.

Fog Layer: The fog layer consists of vehicles/devices in the wireless communication range of RSUs. This layer is significant for VFC due to the growing sensing, computing, communication, and storage abilities of vehicles/onboard equipment. Some data generated by vehicles can be utilized for vehicle-level network decision making, and others can be uploaded to the fog layer for processing. We utilize both parked and moving vehicles near RSUs to form fog nodes for VFC, and the information of sensed events can be uploaded to RSUs. After that, RSUs decide whether the uploaded traffic is handled by the cloudlet or fog nodes. Through analyzing the real-world trajectory of taxies in Shanghai, the traffic flow of vehicles arriving at an RSU is demonstrated to follow a Possion process [5]. We assume that message uploading flows toward RSU r_i follow a Possion process with arrival rate λ_i. Within one region, a cloudlet, a group of RSUs, and a collection of vehicle-based fog nodes coexist.

VFC-Enabled Real-Time Traffic Management

Our designed VFC-enabled architecture concentrates on traffic management and road safety. The sensed events, such as traffic jams, car accidents, and road surface damage, can be uploaded by vehicles to a nearby RSU along their travel routes. After that, the uploaded messages are directed to the cloudlet or fog nodes for processing before uploading to a TMS. Then the TMS broadcasts the feedback message to vehicles through RSUs. Vehicles within the communication ranges of RSUs can be leveraged as fog nodes to process messages directly (instead of cloudlets), by which the response time can be largely shortened. However, cloudlets are necessary due to the dynamic network statuses of vehicle-based fog nodes. If fog nodes are unable to process message flows, cloudlets will manage them. Otherwise, they are in idle. Therefore, the objective of our work is to minimize the response delay for traffic management by load balancing among the cloudlet and fog nodes. This section illustrates the components of expected response time for message uploading, which is shown in Fig. 2.

Problem Formulation

The whole city map is divided into several regions. Within each region, the cloudlet and fog nodes (including both parked and moving vehicle-based fog nodes) coexist. The expected response time for one message can be expressed by

$$E(t) = \alpha E\left(t^C\right) + \beta E\left(t^P\right) + \gamma E\left(t^M\right) + \frac{t_{i,off}}{N}, \qquad (1)$$

where $E(t^C)$, $E(t^P)$, and $E(t^M)$ are the average response time of the cloudlet, and parked and moving vehicle-based fog nodes, respectively. $t_{i,off}$ is the delay generated by input messages from other RSUs. The total number of RSUs is N. Binary variables α, β, and γ denote whether the message is processed by the cloudlet, or parked or moving vehicle-based fog nodes, respectively. The summation of these three binary variables equals 1 during one time slot.

The objective of our VFC-enabled real-time traffic management is to minimize the required time in Eq. 1 by properly assigning traffic flows to the cloudlet, and parked and moving vehicle-based fog nodes. The required response time for the cloudlet and fog nodes is illustrated as follows.

Response time for the cloudlet: The required response time for the cloudlet includes four parts: the consumed time for a message uploading from an RSU to a process server, message waiting time, message processing time, and message forwarding time back to the RSU. For simplicity, we consider the time of message uploading and forwarding back to the RSU as the same value. The average message response time for the cloudlet is the summation of the queueing time, service time, and travel time: $E(t^C) = E(t^C_{que}) + E(t^C_{ser}) + 2t^C_{up}$.

The investigated network can be modeled as a queueing network [6], and the waiting queue of a cloudlet can be viewed as an *M/M/b* queue, including *b* servers. The service rate for a cloudlet is $\rho^C = \lambda^C/b\mu_s$, where λ^C is the flow to be processed by the cloudlet and μ_s is the service rate. The expected service time is: $E(t^C) = 1/\mu_s$.

Response time for fog nodes: Fog nodes are formed by parked and moving vehicles. For the parked-vehicle-based fog model, the total number of parked vehicles is stable during each time slot. For *l* vehicles, the parked-vehicle-based fog node can be modeled as an *M/M/l* queueing system. The corresponding service rate can be computed by $\rho^P = \lambda^P/\mu_s$, where λ^P is the flow to be processed by the parked vehicles. The expected message response time for parked-vehicle-based fog nodes is the summation of the corresponding queueing time, service time, and travel time, that is, $E(t^P) = f(b, \rho^P) + 1/\mu_s + 2d_{ri\rightarrow pfog}$.

For moving-vehicle-based fog nodes, we can demonstrate the corresponding model follows an *M/M/1* queueing system (i.e., the message flowing into the system follows a Poisson procedure with arrival rate λ^m). Since the moving-vehicle-based fog nodes can be regarded as a static server with rate λ, the corresponding service rate is $\rho^m = \lambda^m/\lambda$. Therefore, the average response time for moving-vehicle-based fog nodes is $E(t^M) = \rho_i^m/[\lambda(\rho_i^m)] + 1/\lambda + 2d_{ri\rightarrow mfog}$.

Since message flows at distinct RSUs vary with time, flow redirection among RSUs is necessary. Define $g(i, k)$ as the redirected message flow from RSUs r_i to r_k. The final input message flow at RSU r_i (i.e., $\overline{\lambda_i}$) is to take out the redirected message flow from flow λ_i. Therefore, message offloading for real-time traffic management is to properly assign message flows to the cloudlet and parked- and moving-vehicle-based fog nodes so that the expected response time can be minimized:,

$$\min \frac{1}{su}\sum_{j=1}^{s}\sum_{j=1}^{u}E\left(t_i^j\right)$$
$$\textbf{s.t. } \lambda_i^c + \lambda_i^p + \lambda_i^m \le \overline{\lambda_i}, \quad (2)$$

where *s* and *u* are the total amount of time slots and fog units, respectively. One fog unit is a set of fog nodes at each RSU. The flows arriving at RSU r_i to be processed by the cloudlet and parked- and moving-vehicle-based fog nodes are defined as λ_i^c, λ_i^p, and λ_i^m, respectively.

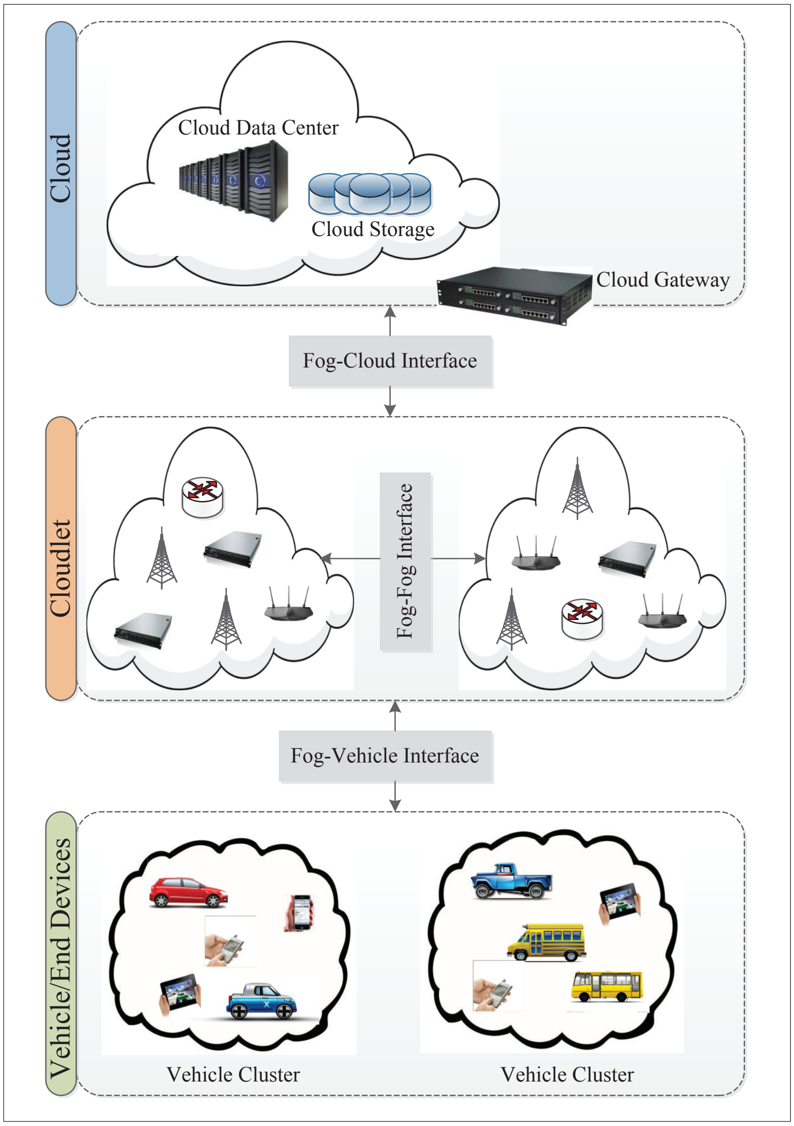


FIGURE 1. Three-layer VFC architecture.

VFC-Enabled Offloading Algorithm

Since the formulated VFC-enabled offloading is a mixed integer nonlinear programming problem, we transfer the objective from the overall response time minimization to the response time minimization during each time slot; that is, the objective in Eq. 2 is substituted by

$$\min \sum_{i=1}^{u}E\left(t_i^j\right)/u.$$

It is obvious that the network performance is influenced by the cloudlet, and parked- and moving-vehicle-based fog nodes. For the cloudlet, it is generally statistic, and the offloading ability depends on its processing ability. However, the locations of both parked- and moving-vehicle-based fog nodes change with time, challenging the estimation of average response time.

We solve the formulated offloading problem by the following steps. First, we calculate the minimum response time for the parked- and moving-vehicle-based fog nodes. After that, the message flows among different RSUs are redirected to approach the obtained average

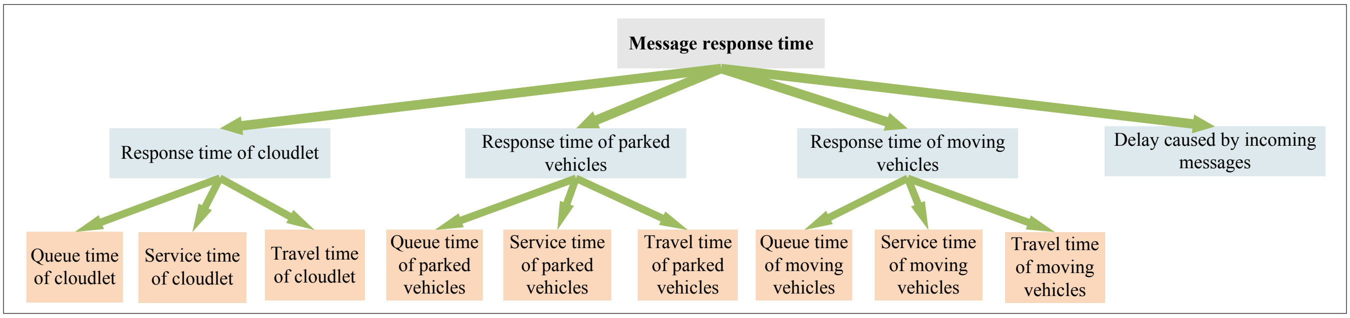


FIGURE 2. Expected message response time in our VFC-enabled framework.

response time. At last, the assignment of cloudlet and input message flows for processing can be determined.

Delay minimization for fog nodes: We first compute the average delay for fog nodes by assigning message flows to the parked- and moving-vehicle-based nodes in the fog unit. Since the total expected delay of parked- and moving-vehicle-based fog nodes is neither a convex nor a concave nonlinear function, this problem can be transferred to a minimum concave-cost network flow problem [6] and solved by the well-studied brand-and-bound algorithm. The average response time for fog nodes can be obtained after acquiring the traffic flows for these two kinds of fog nodes.

After that, we determine the number of input and output message flows for each fog unit. Define as the percentage of message flows handled by fog nodes, the fog units can be separated into two non-overlapped sets, the overloaded and unloaded sets (V_o and V_u). For an overloaded fog unit $i \in V_o$, the amount of message flows to be offloaded to other fog units can be calculated by $\phi_i = \lambda_i \times \delta - \lambda_i^p - \lambda_i^m$, and the total number of input message flows to the fog units is the summation of ϕ_i. For the unloaded fog unit $k \in V_u$, the amount of message flows arriving at unit k can be computed by $\phi_k = \lambda_k^p - \lambda_k^m - \lambda_k \times \delta$. Similarly, the total number of output message flows to the fog units is the summation of ϕ_k.

In order to minimize system transmission delay, the investigated problem becomes redirecting messages from the overloaded fog units to the unloaded ones, that is,

$$\min \sum\nolimits_{i \in V_o} \sum\nolimits_{k \in V_{u_i}} g(i,k) \times d_{r_i, r_k}$$
$$\textbf{s.t.} \sum\nolimits_{i \in V_o} g(i,k) = \phi_k, \sum\nolimits_{k \in V_u} g(i,k) = \phi_i. \quad (3)$$

It is a typical linear minimum cost network flow problem, and can be solved by some existing methods (e.g., the Edmonds-Karp algorithm). The optimal objective can be obtained by integrating the transmission delay with average response time.

Delay minimization for the cloudlet: Due to the dynamic network status of vehicle-based fog nodes, the cloudlet is indispensable in the traffic management system, acting as a complementary part of fog nodes for message flow processing. The objective of cloudlet deployment is to utilize the minimum number of cloudlet servers to cope with the message flows unhandled by the fog nodes. By substituting the flows handled by the cloudlet into Eq. 2 and setting $E(t^c)$ equal to the average response time for a message within a region, the minimum required number of servers on the cloudlet can be obtained.

In our VFC-enabled offloading scheme, fog nodes have high priority to process message flows before redirecting them to the cloudlet. Our scheme contains the following steps:

1. Calculating the average response time of messages by the brand-and-bound algorithm
2. Computing the redirected message flows from the overloaded fog nodes to the unloaded ones by the Edmonds-Karp algorithm
3. Determining the amount of message flows in each fog unit to be processed by the parked and moving vehicle-based fog nodes respectively
4. Confirming the required number of servers on the cloudlet

Performance Evaluation

In order to demonstrate the superiority of the presented VFC-enabled real-time traffic management scheme, this subsection provides some preliminary results according to the real-world city map and traces of taxies in Shanghai, China. Specifically, one administrative division is defined as a region. In each region, a candidate RSU is located in the center of a sub-district. We take Putuo and Huangpu districts as examples. The selected GPS locations are shown in Fig. 3. The traces of over 1000 taxies are collected from the entire month of April 2015, including the GPS locations, directions, speeds, and record times. The statistic of the arrival rate for moving vehicles is compiled every 10 minutes within the range of 500 m of each RSU. By analyzing the arrival rate of moving vehicles in the dataset, we notice that the average number of vehicles is between 100 and 500 per second with an RSU. We set the corresponding number between 200 and 600 per second by considering different proportions of moving and parked vehicles.

To the best of our knowledge, the designed VFC-enabled traffic management scheme is a prior work to provide a feasible solution for distributed citywide traffic management. A randomized strategy is selected for comparison, attempting to maximize the workload processed by both the parked- and moving-vehicle-based fog nodes. The residual flows unable to be processed by fog nodes will be handled by the cloudlet.

The average response time with different message arrival rates in Putuo and Huangpu districts is shown in Fig. 4. It is obvious that as the message arrival rate increases, the average response time of the randomized strategy skyrockets, while our solution increases gently. For example, in Fig. 4b, the average response time of the randomized strategy and our solution are 4.2 s and 0.6 s, respectively. This is because our solution can dynamically balance network loads instead of concentrating on maximizing network loads of fog nodes as in the randomized strategy.

Figure 5 illustrates the trend of average response time when the total number of parked-vehicle-based fog nodes increases. The response time of these two methods decreases as the number of fog nodes increases, because more parked-vehicle-based fog nodes are accompanied with more powerful processing ability. It is straightforward to see that our method is more suitable for various kinds of network situations and is almost independent of the number of fog nodes, demonstrating its scalability well.

Research Challenges and Open Issues

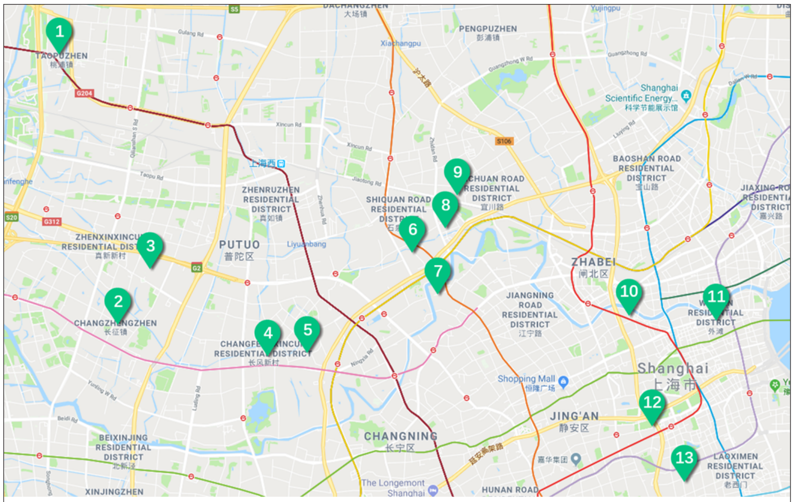
The study of VFC-enabled real-time traffic management is still at the very beginning. Some corresponding research challenges and open issues are discussed in this section.

Traffic Mobility and Predition

Drivers can acquire the information of road conditions and events that have occurred by mobility and traffic prediction to avoid possible congestion and accidents. Since it is vital to integrate parked and moving vehicles efficiently for the implementation of VFC, accurate traffic mobility and prediction is significant for high-efficiency utilization of vehicles' energy and computing resources. Traffic prediction can be conducted by the cooperation of traffic authorities, vehicular fog nodes, and vehicles through data sensing, processing, and sharing. Massive historical and real-time data can be leveraged for mobility prediction by traffic flow mining based on the vehicle's position, direction, and velocity. However, how to cope with network heterogeneity needs to be investigated well [7].

Predictive Offloading

Since the resources of vehicles are rather limited, computing-resource-hungry applications challenge the design of vehicular networks. Although traffic offloading can be conducted by the base station or remote cloud, offloading efficiency will be seriously degraded by the distant deployment of backbone and backhaul networks [8]. In order to provide timely feedback from vehicles, VFC-enabled predictive offloading is promising. In [9], a VFC-enabled offloading framework is investigated to reduce latency and delivery cost. By estimating time consumption, a predictive offloading mechanism is designed to decide whether direct uploading or predictive relay forwarding is suitable for vehicular communications. In order to take advantage of the potential network resource from other connected vehicles and alleviate link congestion, a novel architecture is designed for delay-tolerant traffic offloading without extra deployment of infrastructures and hardware [10]. It is noticed that most of the existing research on VFC-enabled offloading assumes that fog nodes are stationary



Putuo district		Huangpu district	
ID	Locations	ID	Locations
1	31.278978, 121.363603	10	31.239377, 121.465145
2	31.237881, 121.373256	11	31.238309, 121.481430
3	31.246551, 121.379041	12	31.222411, 121.470225
4	31.232627, 121.401455	13	31.214104, 121.475516
5	31.233867, 121.408167		
6	31.249332, 121.426857		
7	31.242759, 121.431407		
8	31.253006, 121.432931		
9	31.258151, 121.435001		

FIGURE 3. Selected GPS locations in Shanghai.

or move along predefined paths, which is unrealistic for traffic management in smart cities.

Security and Privacy in VFC

With the rapid development of vehicles, enabling secure communication and information exchange among vehicles is significant for the realization of network service and application. Although the network storage, structure, and computing facilities of VCC can be extended by VFC, potential security and privacy issues arise due to its unique features. Current relevant schemes mainly concentrate on the security and privacy in fog computing, such as the scalability, privacy-preserving authentication, and forensics. In order to promote road safety and lower traffic accidents caused by poor road conditions, crowdsensing and vehicle-based sensing are advocated. A VFC-enabled privacy-preserving protocol is presented to monitor road surface conditions [11], which is a high-efficiency certificateless aggregate signcryption method. The security and privacy requirements of VFC-enabled crowdsensing are discussed in [12], including the unique infrastructure and various promising applications (e.g., parking navigation and traffic collision reconstruction). Although possible solutions are provided for security assurance and privacy preservation, private information of road events, unauthentic vehicle connections, and heterogeneous road infrastructures still challenge the design of VFC architecture. In addition, when computational tasks are conducted by various vehicles in VFC, privacy preservation is significant [13].

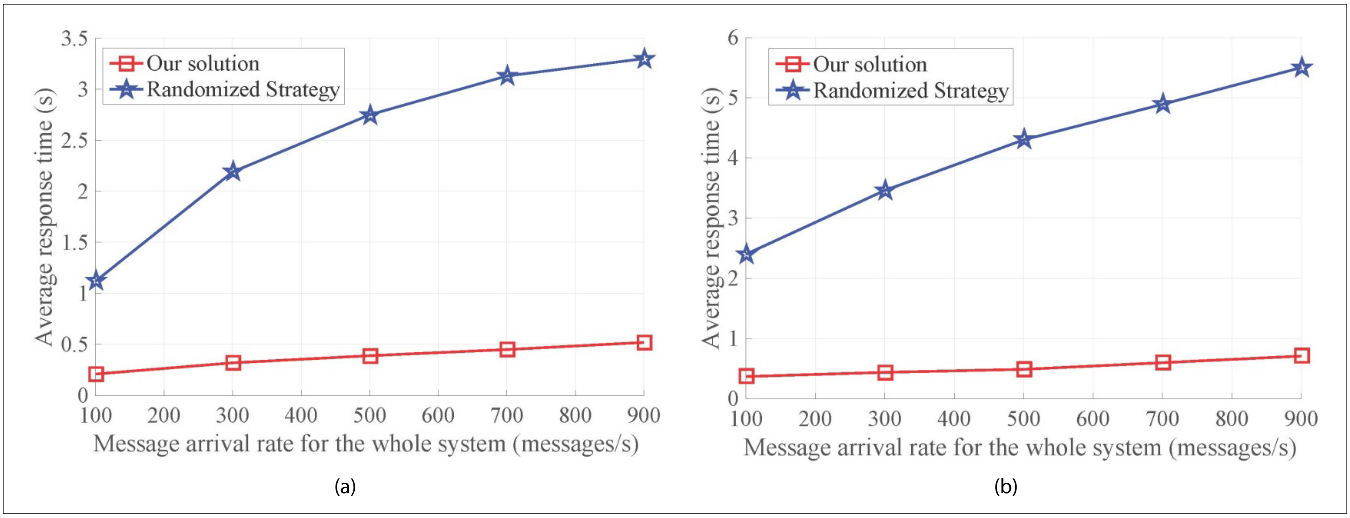


FIGURE 4. Average response time with different message arrival rates in Shanghai: a) Putuo district; b) Huangpu district.

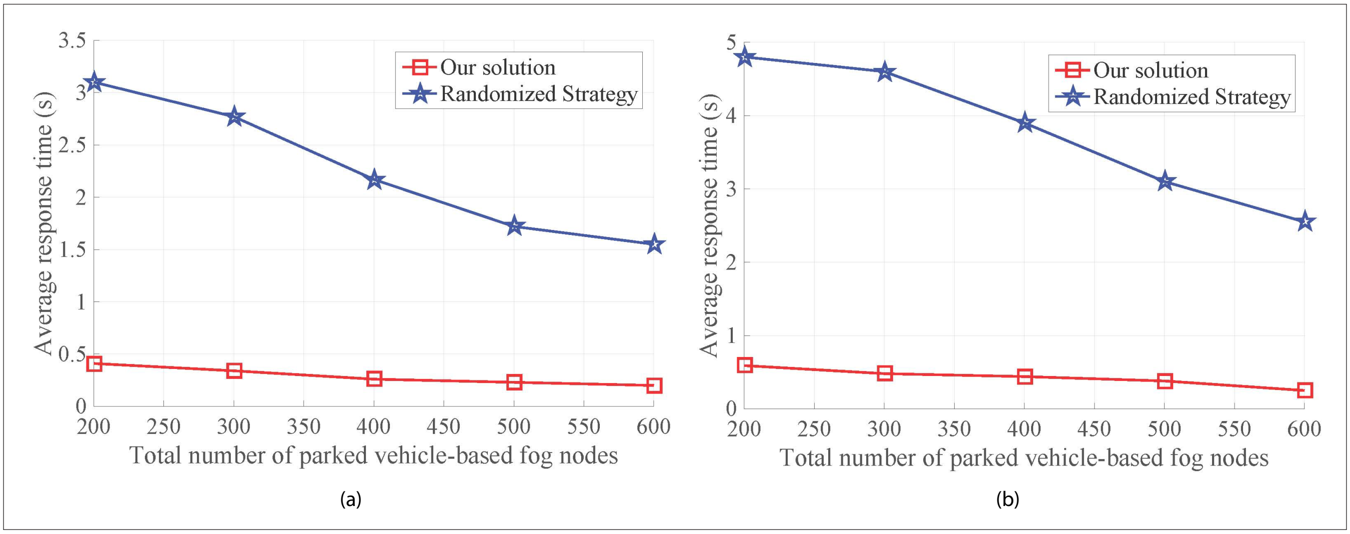


FIGURE 5. Average response time with the number of parked vehicle-based fog nodes in Shanghai: a) Putuo district; b) Huangpu district.

Interaction Between Fog and Cloud

VCC and VFC are promising alternatives to provide ubiquitous Internet connections and respond to huge traffic demands for citywide intelligent transportation systems. VCC manages network resources in a centralized manner for high-efficiency management, while VFC makes use of edge computing and storage to reduce latency and improve resource utilization. Since the design philosophies of VCC and VFC are complementary, their interactions require comprehensive study. A fog-to-cloud computing system is hopeful for grouping cars in a parking lot and constructing roadside clouds by controlling traffic lights dynamically [14]. With the objective of providing network services, such as network function virtualization, on top of a telecom operator's infrastructure, a secure and distributed fog-cloud computing architecture is presented [15]. Services deployed on fog nodes can provide timely and flexible network responses. However, the business model in fog computing is still open, not to mention VFC. Cooperative uploading and offloading by parked and moving vehicles can be a new trend of shared economy, calling for a reliable credit and payment system.

Conclusion

In order to manage traffic in a distributed and real-time manner, this article presents a VFC-enabled traffic management scheme for smart cities. Specifically, a three-layer VFC architecture is constructed to dynamically cooperate with each other for network load balancing. The advantages of our designed architecture are:

1. It is based on a decentralized network structure, so data processing can be managed within each region.
2. The integration of the cloudlet and fog nodes can effectively offload network traffic and largely alleviate network burdens.
3. Response delay can be largely reduced because the cloudlet and fog nodes are close to terminals.

To the best of our knowledge, it is fresh work to provide a VFC-enabled systematic design for distributed citywide traffic management. Performance evaluations demonstrate that our solution can provide prompt response under various network circumstances. Since the study of VFC-enabled real-time traffic management is still in its infancy, some possible research challenges and future directions are highlighted to provide a roadmap for the VFC ecosystem.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China (61502075 and 61671093), by the China Postdoctoral Science Foundation under grant 2018T110210, and by the Fundamental Research Funds for the Central Universities under Grant DUT17RC(4)49.

References

[1] C. Huang *et al.*, "Vehicular Fog Computing: Architecture, Use Case, and Security and Forensic Challenges," *IEEE Commun. Mag.*, vol. 55, no. 11, Nov. 2017, pp. 105–11.

[2] X. Du and F. Lin, "Maintaining Differentiated Coverage in Heterogeneous Sensor Networks," *EURASIP J. Wireless Commun. and Networking*, vol. 5, no. 4, 2005, pp. 565–72.

[3] Z. Ning, *et al.* "A Cooperative Quality Aware Service Access System for Social Internet of Vehicles," *IEEE Internet of Things J.*, vol. 5, no. 4, 2018, pp. 2506–17.

[4] X. Hou *et al.*, "Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures," *IEEE Trans. Vehic. Tech.*, vol. 65, no. 6, 2016, pp. 3860–73.

[5] J. He *et al.*, "Delay Minimization for Data Dissemination in Large-Scale VANETs with Buses and Taxis," *IEEE Trans. Mobile Computing*, vol. 15, no. 8, 2016, pp. 1939–50.

[6] L. Klennrock, *Queueing Systems Volume 1: Theory*, New York, 1975.

[7] X. Du *et al.*, "A Routing-Driven Elliptic Curve Cryptography Based Key Management Scheme for Heterogeneous Sensor Networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, 2009, pp. 1223–29.

[8] Z. Ning *et al.*, "A Social-Aware Group Formation Framework for Information Diffusion in Narrowband Internet of Things," *IEEE Internet of Things J.*, vol. 5, no. 3, 2018, 1527–38.

[9] K. Zhang *et al.*, "Mobile-Edge Computing for Vehicular Networks: A Promising Network Paradigm with Predictive Off-Loading," *IEEE Vehic. Tech. Mag.*, vol. 12, no. 2, 2017, pp. 36–44.

[10] P. Si *et al.*, "DaVe: Offloading Delay-Tolerant Data Traffic to Connected Vehicle Networks," *IEEE Trans. Vehic. Tech.*, vol. 65, no. 6, 2016, pp. 3941–53.

[11] S. Basudan *et al.*, "A Privacy-Preserving Vehicular Crowdsensing Based Road Surface Condition Monitoring System Using Fog Computing," *IEEE Internet of Things J.*, vol. 4, no. 3, 2017, pp. 772–82.

[12] J. Ni *et al.*, "Securing Fog Computing for Internet of Things Applications: Challenges and Solutions," *IEEE Commun. Surveys & Tutorials*, vol. 20, no. 1, 2018, pp. 601–28.

[13] X. Du *et al.*, "An Effective Key Management Scheme for Heterogeneous Sensor Networks," *Ad Hoc Networks*, vol. 5, no. 1, 2007, pp. 24–34.

[14] X. Masip-Bruin *et al.*, "Foggy Clouds and Cloudy Fogs: A Real Need for Coordinated Management of Fog-to-Cloud Computing Systems," *IEEE Wireless Commun.*, vol. 23, no. 5, 2016, pp. 120–28.

[15] R. Vilalta *et al.*, "TelcoFog: A Unified Flexible Fog and Cloud Computing Architecture for 5G Networks," *IEEE Commun. Mag.*, vol. 55, no. 8, Aug. 2017, pp. 36–43.

Biographies

Zhaolong Ning (zhaolongning@dlut.edu.cn) received his M.S. and Ph.D. degrees from Northeastern University, Shenyang, China. He was a research fellow at Kyushu University, Fukuoka, Japan. He is an associate professor in the School of Software, Dalian University of Technology, China. His research interests include vehicular networks, fog computing, and network optimization.

Jun Huang (xiaoniuadmin@gmail.com) received his Ph.D. in 2012 from Beijing University of Posts and Telecommunications, China. He was a visiting student at Waseda University, Japan, a research scientist at the South Dakota School of Mines, and a research fellow at the University of Texas at Dallas. He is now a full professor at the Institute of Electronic Information and Networking, Chongqing University of Posts and Telecommunications. He has over 80 publications, and several of them are in prestigious journals or conference proceedings. His research interests include IoT and D2D/M2M.

Xiaojie Wang (wangxj1988@mail.dlut.edu.cn) received her Master's degree from Northeastern University, China, in 2011. From 2011 to 2015, she was a software engineer at NeuSoft Corporation. Currently, she is pursuing a Ph.D. degree at Dalian University of Technology. Her research interests are edge computing, vehicular networks, and social computing.

Services deployed on fog nodes can provide timely and flexible network responses. However, the business model in fog computing is still open, not to mention VFC. Cooperative uploading and offloading by parked and moving vehicles can be a new trend of shared economy, calling for a reliable credit and payment system.