IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 16, 2019, accepted July 2, 2019, date of publication July 8, 2019, date of current version August 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927343

Jamcloud: Turning Traffic Jams Into Computation Opportunities–Whose Time Has Come

XUEFENG XIAO[1], XUESHI HOU[2], CHUANMEIZI WANG[2], YONG LI[2], (Senior Member, IEEE), PAN HUI[3], (Fellow, IEEE), AND SHENG CHEN[4,5], (Fellow, IEEE)

[1]School of Economics and Management, Beijing Information Science and Technology University, Beijing 100192, China
[2]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[3]Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong
[4]School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K.
[5]King Abdulaziz University, Jeddah 21589, Saudi Arabia

Corresponding author: Xuefeng Xiao (xfxiao08@163.com)

ABSTRACT Traffic jam has been and will remain a major problem in most cities around the world. We view this situation as a computation opportunity and propose to build the cloud-computing facilities on the top of jammed cars and other vehicles to turn the energy and other resources that otherwise would be wasted into computing power. Specifically, we define the vehicular mobile cloudlet as a group of nearby vehicular mobile devices congested in the traffic jams while connected by short-range communications. Based on local mobile cloudlets of congested vehicles and available remote cloud-computing resources, we propose and evaluate the JamCloud, a system to collect and aggregate the computation capacities of congested vehicles in the city. For this newly-conceived novel cloud system, the fundamental problems are how much computation capacity the mobile cloudlets have and what is the overall achievable performance of the whole JamCloud system. Based on the three realistic large-scale urban vehicular mobility traces, we analyze and model the vehicular mobility patterns as well as the computation capacity in both the mobile cloudlet and system-wide. Specifically, by analyzing the patterns of staying time, resident number, and incoming and outgoing of vehicles in the regions with traffic jams, we model the mobile cloudlet as a periodic non-homogeneous immigration-death process, which predicts its computational capability with accuracy above 90%. Based on the observed strong Poisson features of mobile cloudlets, we further propose a queueing network model to characterize the overall performance of JamCloud with the computing resources of multiple mobile cloudlets and remote clouds. Our study thus reveals the microscopic computation capability of local cloudlets as well as the overall and asymptotic performance of the JamCloud, which provides foundational understanding to design, such systems in practice. With the inevitably growing trend of making vehicles electric, and in particular with the forthcoming 5th generation (5G) mobile communication technology, the time has finally come to turn JamCloud into reality.

INDEX TERMS Mobile cloudlet, vehicular cloud computing, traffic jams, mobility model, computation capacity, electric vehicle, 5th generation mobile communication.

I. INTRODUCTION

A. MOTIVATIONS

The number of vehicles in operation worldwide surpassed the 1.25 billion-unit mark in 2015 [1], and there was a record of 79 million new cars sold worldwide in the year 2017 [2]. Thus, 216,400 extra cars are added to the global traffic jam every day. Traffic jam has been a general problem in most cities, and the situation is becoming worse. For example, China's capital, Beijing, has been struggling to deal with traffic congestion for many years. According to the deputy director of Beijing Transportation Research Centre, the average weekday congestion time in the first half of 2013 was 100 minutes [3]. Another extreme example was a clog stretching for 100 kilometres (62 miles) between Beijing and Jining, and lasted nearly a month. In the U.S. and Europe,

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

the situation may be better. Nevertheless, in 2018, average London driver lost 227 hours due to congestion and the cost of congestion per driver was £1,680, while average Boston driver lost 164 hours and the cost of congestion per driver was 2,291 US dollars, according to INRIX 2018 Global Traffic Scorecard [4].

The past two decades have witnessed a growing trend in making the vehicles on our roads smarter and offering better driving experience [5]. Equipped with an on-board computer, wireless devices, e.g., radio transceivers and sensing devices, and a rechargeable battery, a typical modern car or bus is able to interact with the Internet and associated services. As technology is moving closer and closer to embedded sophisticated resources on individual vehicles, particularly as the society is inevitably moving away from fossil-fuelled cars towards electric cars [6], it is certain that in the very near future even the low-end vehicles will be equipped with on-board wireless communication devices, data collection devices and computers. The initial views of leveraging vehicular networking systems were to keep the drivers informed about potential safety risks and enhance their awareness of traffic condition [7]. However, these applications only require small portion of the huge on-board capacities of vehicles and, therefore, the available resources are not fully utilized.

Moreover, when the cars or buses stuck in the traffic, the engines are running but the vehicles are not moving or moving very slowly so that the fuel and other resources are mostly wasted. However, we view this situation as a computation opportunity, and we can build cloud-computing facilities on the top of these jammed vehicles so as to turn the energy and other valuable resources that would otherwise be wasted into computing power. The on-board computers can form a 'computing fleet' when the vehicles are slowly moving in traffic congestion situations. The fact that vehicles blocked in a traffic jam are moving slowly as well as in a highly predictable manner also helps alleviate the mobility challenges, which otherwise are typically very hard to address in mobile computing environments. Furthermore, since the vehicles are organized in a highly condensed and regular way, device-to-device (D2D), i.e., vehicle-to-vehicle (V2V), communication expenses are likely to be low. Thus the congested vehicles can form a vehicular mobile cloudlet, which is defined as a group of nearby vehicular mobile devices connected by wireless V2V communication enabling to transfer data and exchange computation resources for other computing tasks.

B. OUR CONTRIBUTIONS

Against the above background, In this paper, we propose and evaluate JamCloud, a system based on local mobile cloudlets of congested vehicles and remote cloud computing centers, to collect and aggregate the computation capacities of congested vehicles. The fundamental problems in the design of JamCloud are how much the computation capacity the vehicular mobile cloudlets have and what is the achievable overall performance of the JamCloud system, which are both unknown. Firstly, since the presence of vehicles in close proximity is usually an un-planned event, the pooling of the resources from these vehicles must occur spontaneously by the common recognition of a need for which there are no pre-assigned or dedicated resources available. This agility of action of the mobile cloudlets does not exist in conventional clouds and turns out to be an important defining characteristic of JamCloud. Therefore, how much the computation capacity the cloudlets have needs to be explored, and more importantly what are applications that can benefit from this computation capacity need to be identified. Secondly, the macroscopic system performance of JamCloud depends not only on the dynamics of cloudlets but also on the capacity of remote clouds. Thus the overall achievable performance of the system requires further investigations.

More specifically, we investigate the temporal and spatial dynamic computation capacity of the JamCloud system using three realistic large-scale vehicular mobility traces of Beijing, Shanghai and Nanjing, build models based on the immigration-death process plus queueing network, and define the metrics to describe the system's computation capacity as well as the achievable performance. The novel contributions of our work can be summarized as follows.

- We conceive for the first time the idea of JamCloud to aggregate computation capacity from local congested vehicles and remote cloud computing centers to better satisfy the computation demand of vehicles as well as to exploit the computation capacity of vehicular mobile cloudlets for serving other applications.
- We analyze the temporal and spatial dynamic computation capacity of mobile cloudlets formed by congested vehicles at intersections based on three realistic large-scale vehicular mobility traces, model the system by an immigration-death process, and further based on the proposed model, predict its computation capacity.
- Based on the above model, we propose a queueing network model to describe the mobility patterns of the whole system. By defining appropriate metrics, we carry out theoretical analysis and simulations to evaluate the probability of satisfying computation demand of JamCloud, and reveal the overall and asymptotic performance of the system.

C. JAMCLOUD WHOSE TIME HAS COME

This work has been more than five years in making during which there were some critics for JamCloud concept – primarily it was not apparent which vital application could directly benefit from JamCloud and how realistic to implement it in practice. During this period, we have witnessed the drive to make vehicles on road electric and more importantly the forthcoming 5th generation (5G) mobile communication technology [8]. These new technologies have solved critical implementation issues and provided foundation and infrastructure for realizing JamCloud. More importantly, with the coming 5G, JamCloud has found its ideal, direct and critical application target.

5G will provides Gbps rate, enabling massive connections, with very low network latency. With 5G comes massive opportunities - everything is connected, Internet of things (IoT), intelligent transport system, self-driving vehicle - the list is endless. 5G also has to be very energy efficient, consuming much less power. Hence network densification has to be driven to a new level. A 5G base station (BS) or road side unit (RSU) of a small cell must be able to provide extremely high peak baseband processing capacity to meet peak-time demand. Maintaining a high capability computing unit at a small BS or RSU is not very economy. Furthermore at off-peak time, most of this computing capability is idle as it is not needed. A solution is for BS or RSU to 'outsource' its baseband processing requirements to cloud based on the concept of centralized radio access network (C-RAN) [9]–[11].

A 5G BS or RSU can send conveniently its baseband processing jobs to a JamClond's vehicular cloudlet formed at the nearby intersection. It can easily be seen that the computing capacity of a mobile cloudlet formed by the vehicles congested at the intersection dynamically matches the demand of computing requirements well. At daytime busy hours, the BS or RSU has peak demand for baseband processing, and the computing capacity of the mobile cloudlet also happens at peak. At nighttime, the computing capacity of the mobile cloudlet is low, as there is no heavy traffic jam. But the demand of the BS or RSU is also very low, and there is no need to outsource the demand. The fact that the JamCloud is nearby is not only handy but also very beneficial to low network latency. It can be seen that JamCloud's time has come.

D. RELATED WORK

Together with an explosive growth of the mobile applications and emerging of cloud computing concept, mobile cloud computing (MCC) has been introduced as a potential and enabling technology for mobile services [12]. Traditionally, MCC integrates the remote cloud into the mobile environment to overcome the obstacles related to the performance, environment and security in mobile computing [13]. However, the mobile devices suffer from computing resource limitation, battery restriction and processing time constraint [14]. In addition, because MCC uses the client-server communication model, uploading real-time information on the cloud via mobile Internet is costly and time consuming [15]. Clearly, this traditional MCC cannot reliably offer high capacity and low latency to dedicated computing application.

Therefore, a peer-to-peer communication model for MCC [16]–[19] was proposed in the light of the increasing memory and computational power of mobile devices [20], [21]. In these works, mobile cloudlet is a group of nearby mobile devices connected by short range communications [21]. Thus the idea of computation offloading through mobile cloudlets has been proposed. However, these works are still either limited to low capability devices and static scenario, or restricted to naive distributed computing approaches, and they have serious scalability and reliability issues. By contrast, our work deals with vehicles, such as cars and buses, in traffic jams, which potentially have high computation capability, high-energy capacity, and low mobility, and we focus on harvesting otherwise wasted or idle resources. Furthermore, we propose realistic scenarios and approaches to build scalable mobile cloud computing facilities.

Another new technological shifting is vehicular cloud computing (VCC), which takes advantage of cloud computing to serve the drivers of vehicular ad-hoc networks with a pay-as-you-go model. Thus, the objective of VCC is to provide several computational services at low cost to vehicle drivers [5]. The idea of vehicular cloud was proposed in [22], which is an extension of conventional cloud computing with several new dimensions. However, VCC still suffers from several drawbacks, such as the high cost of the service constrained communications due to high mobility of the vehicles [23], [24]. Different from these works, we focus on a hybrid system including the local mobile cloudlet formed by the congested vehicles to overcome these drawbacks.

Liu *et al.* [25] and Malandrino *et al.* [26] proposed the idea of how parked vehicles can help to improve the connectivity among moving vehicles in sparse traffic areas. The key value of these works is in establishing data exchange between moving and parked vehicles. Utilizing vehicles as infrastructures for communication and computation was extensively investigated in [27], which also includes communication and computation capacity analysis for parked cars as vehicular frog computing (VFC) cloudlet. We may call this type of vehicular cloudlet as ParkedCloud, and a US patent was recently granted for ParkedCloud as networked communication infrastructure [28]. However, in this paper, we focus on slowly moving vehicles in traffic jams. To the best of our knowledge, no existing work investigates an approach or architecture to turn slowly moving vehicular fleets in traffic jams into networked computation clouds.

The rest of this paper is organized as follows. The conceived JamCloud system is described in Section II. The datasets used to investigate JamCloud's capacity and our preprocessing method are presented in Section III. Section IV is dedicated to analyzing and modeling the dynamics of vehicular cloudlets formed by congested vehicles, and simulations for calculating the microscopic computation capacity are included. The metrics and simulations to evaluate the macroscopic performance of JamCloud are shown in Section V. Finally, we conclude the paper in Section VI.

II. JAMCLOUD SYSTEM OVERVIEW

A. SYSTEM DESCRIPTION

A JamCloud contains all vehicles equipped with embedded computers jammed around the intersections of a city, where BSs or RSUs may also be deployed to connect to the mobile network and remote clouds. A sketch of the system is shown is Fig. 1, where the BSs or RSUs are linked to the remote cloud computing centers, network center controller and Internet by backhaul connections indicated in Fig. 1 as wired links.

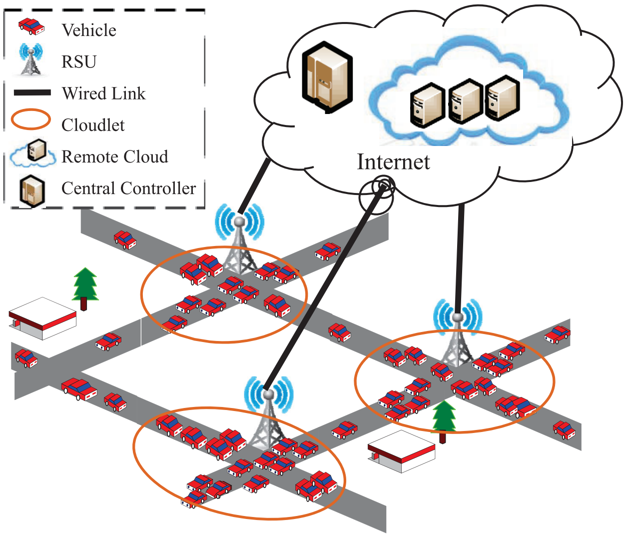


FIGURE 1. JamCloud system overview.

The vehicles congested around an intersection form a local mobile cloudlet. Originally, the purpose of such a mobile cloudlet is to help each other. A vehicle may offload computation to other vehicles or to load computation from others through V2V communications. Moreover, the BSs or RSUs deployed at intersections are able to communicate with remote cloud computing centers through backhaul connections and with the congested vehicles through vehicle to infrastructure (V2I) communications. Therefore, the whole system becomes a hybrid cloud which aggregates computation resources from the mobile cloudlet and remote cloud, and reallocates them to vehicles in order to satisfy the computation demand of every vehicle so as to optimize the system's capacity.

Eclectic vehicle's on-board computer is very powerful, and request for help from other vehicles becomes less likely. As more and more vehicles on road become electric, the aggregated computation capacity of a mobile cloudlet will become under utilized or wasted. It is highly desired to harvest this computation resource by downloading other computation applications to a mobile cloudlet. With the forthcoming 5G, we readily find such a vital computation application to JamCloud. Rather than submitting its baseband signal processing tasks to a remote cloud center [9]–[11], a BS or RSU can simply send them to the mobile cloudlet formed by congested vehicles in a nearby intersection for completion. This not only utilizes the idle computation power of the congested vehicles but also helps to achieve low network latency which is one of the critical metrics of 5G networks.

B. COMPUTATION RESOURCES AGGREGATION AND REALLOCATION

When vehicles stuck in the traffic, usually around intersections, not moving or moving slowly, these vehicles are tightly packed, forming a local 'jam cloud'. Some of these vehicles have extra computing resources while others may need more. Naturally, these vehicles may want to balance the computing resources among themselves. More importantly, the aggregate computing resources of such a local mobile cloud are typically more than what these vehicles need, and it is highly desired to harvest the aggregate computing resources of these vehicles for other computation applications. It can be assumed that an efficient controller is embedded in the system to manage these resources. In fact, owing to network densification, at an intersections around the city, there will be a BS or RSU nearby. The vehicles congested around an intersection can communicate to the BS or RSU deployed there through V2I communications as well as communicate with each other via V2V communications. Thus, these vehicles naturally form a vehicular mobile cloudlet. The information regarding the computation capacity of a vehicle is uploaded to the control system of the cloud, i.e., the BS or RSU, so that the computation resources of vehicles can be aggregated by the system. Besides, the information of computation resources of remote clouds is known to the BS or RSU.

When a vehicle in the JamCloud demands a certain amount of computation resources more than what it has, a request can be submitted to the nearest BS or RSU, and there are two sources that can provide this extra computation demand, the local mobile cloudlet and a remote cloud. Even though a remote cloud has much greater computation capacity than the local cloudlet, the energy and time cost caused by local cloudlet is much lower, and local vehicles can actually share their computation capacity freely. Hence, in the original concept of JamCloud, if the demand can be satisfied by local mobile cloudlet, the reallocation of computation resources will be completed locally. Otherwise, the request will be submitted to the central control system of the cloud, and remote computation centers will receive a command to allocate computation resources to the vehicle. However, as mentioned previously, computation capacity of vehicle has increased considerably, and the aggregate computation resources of a local cloudlet are more than enough to meet all the demands of the vehicles in the local cloudlet. Therefore, there is no need to ask for the resources from remote cloud center. Moreover, there will be substantial computation resources left by a local mobile cloudlet after serving all the local needs. Consequently, the challenge for the system is to find other suitable computation applications that can utilize these otherwise are idle or wasted computation resources.

With the forthcoming 5G, we can readily identify such an application which is ideal for exploiting the aggregate computation resources of a local mobile cloudlet. The BS or RSU nearby can simply outsources its baseband signal processing jobs to the local mobile cloudlet to complete. Intuitively, it can be seen that the dynamics of the aggregate computation capacity of a local mobile cloudlet match the demands of baseband signal processing by BS or RSU well. During rush hours of daytime, for example, the computation capacity of local cloudlet around an intersection is at its peak, while the BS or RSU nearby is also at its peak demand for baseband signal processing. At late nighttime, the local cloudlet has very little capacity, which coincides with the period of low demand for baseband signal processing by the BS or RSU. It is also intriguing to see that some of the computation tasks completed by a local cloudlet may actually relate to the mobile communications made by the vehicles of the cloudlet. An important advantage for BS or RSU to download its baseband signal processing to nearby local cloudlet rather than to submit them to a remote cloud center is that it helps to maintain low network latency.

C. FUNDAMENTAL PROBLEMS

For this conceived JamCloud system, obviously, there exist some unknown fundamental problems. Firstly, what is the computation capacity of a vehicular mobile cloudlet? We do not have a comprehensive understanding to answer the question yet. Our first technical contribution is to address this fundamental issue. In order to provide a quantitative evaluation, we can define the computation capacity as the aggregate floating point operation speed of all the devices in the mobile cloudlet. Thus, the number of vehicles incoming, outgoing and staying around an intersection, namely, the vehicles forming the cloudlet, are all contributing to its aggregate computation capacity. Besides, the staying time of a vehicle is also important to its contribution to this computation capacity. In other words, the statistics of vehicles' mobility patterns are fundamental to answer the first question. Based on our available vehicular mobility traces, we can collect some useful statistics on the mobility patterns of vehicles, and these statistics enable us to model and estimate the computation capacity of a mobile cloudlet.

Next, what is the overall achievable performance of the JamCloud system composed of local cloudlets and remote cloud? Specifically, if the JamCloud is built, how often can the demand of computation in the cloud be satisfied? How do we evaluate the performance of the system? Hence, we need to define appropriate metrics to evaluate the performance of the system and build corresponding suitable models for analysis. Besides, what is the asymptotic performance when the scale of the system tends to infinity? We also need to analyze this problem using the metrics defined for performance evaluation.

In this paper, we aim to provide understanding and insights to these fundamental problems via empirical study based on three large-scale urban vehicular mobility traces of Beijing, Shanghai and Nanjing.

III. DATA TRACE AND PROCESSING

We investigate the temporal and spatial dynamic computation capacity of the JamCloud system using realistic vehicular mobility traces of Beijing, Nanjing and Shanghai. We choose these cities to study because traffic jam is a common scene around these three cities. In this section, we provide a brief description of the vehicular mobility traces and introduce our processing method as well as justify that these traces are sufficient and appropriate for our study.

TABLE 1. Trace summary.

Trace	Beijing	Nanjing	Shanghai
Year	2009	2010	2007
Number of vehicles	27,000	7,200	4,300
Duration (days)	30	90	30

A. DATA TRACES

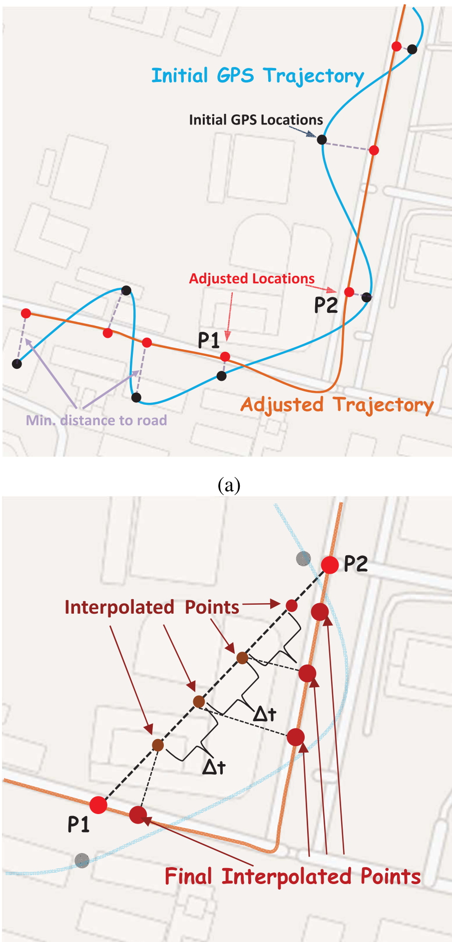
Beijing trace [29] is the largest vehicular mobility trace available to us. To collect the data, we used the mobility track logs obtained from 27,000 participating Beijing taxis carrying GPS receivers with the duration of one month in May 2009. Taxis are more sensitive to urban environments in terms of underlying road topology and traffic control, compared with other vehicular devices, such as buses, since they have broader coverage in space and time. Furthermore, it is practical and relatively easy to collect taxi mobility data as compared to private cars. These were the reasons why we chose this data set. In this data set, the locations and timestamps of the moving taxis were recorded every 15 seconds. The information we specifically used includes: the taxis' ID, the longitude and latitude coordinates of the taxis' location, and timestamps. We obtain about twenty million effective records per day from the whole data trace.

Similarly, Nanjing trace [30] and Shanghai trace [31] are also collected by GPS receivers on taxis. Nanjing trace contains the mobility track logs of 7,200 Nanjing taxis for a period of 60 days in 2010. Shanghai trace contains track logs of 4,300 taxis within the duration from January 31, 2007 to February 26, 2007.

The related information of these three vehicular mobility traces are summarized in Table 1.

B. DATA PREPROCESSING

We now describe the preprocessing of vehicular mobility trace. We obtain the taxis' locations varying with time from the taxis' moving trace, which were measured by GPS devices and the coordinates are longitudes and latitudes whose precisions are 0.00001 degrees. For the convenience of processing data and using Beijing trace as an example, we convert the coordinates to meters with a precision of 1 m and set the origin point $(0, 0)$ at $(40.0°N, 116.4°E)$ near the center of Beijing. Since the location data are measured by the GPS devices, the noise may exist in the collected data due to the inaccuracy of GPS devices. Also since the taxis may not report their locations at the same time slots with the same fixed frequency, we need to process the data trace to obtain the accurate locations of all the taxis in the same time slots and frequency. Thus we first use the city map of Beijing to correct the taxis' locations so that they are all on the city's roads, and this location adjustment is shown in Fig. 2 (a). We then use the method of interpolation to insert the location points at the time slots we need so that all the taxis have location information at every ten-second interval, which becomes the time measurement sampling interval in our data processing. We now explain how we carry out this interpolation, as illustrated in Fig. 2 (b).



(a)

(b)

FIGURE 2. Illustration of vehicular GPS data preprocessing including: (a) location adjustment, and (b) time frequency adjustment.

Consider that we only have the location information (x_1, y_1) and (x_2, y_2) of a taxi at time points t_1 and t_2, respectively, but we do not have any information of the taxi between t_1 and t_2. If $t_2 - t_1 \leq 10$ sec., we do not estimate the location of the taxi between t_1 and t_2. If $t_2 - t_1 > 10$ sec., then in order to get the location of the taxi at any time $t \in (t_1, t_2)$, we estimate the location (x_t, y_t) by the following interpolation

$$l_t = l_1 + \frac{t - t_1}{t_2 - t_1}(l_2 - l_1), \quad \text{with } l = x \text{ or } y. \tag{1}$$

After obtaining (x_t, y_t), we again need to adjust it to be on a city's road using the city map. Note that the accuracy of this interpolation with real map based adjustment is much higher than the accuracy of using just the interpolation (1).

After the data preprocessing, we obtain an instantaneous two-dimensional distribution map of the taxis' positions for every minute. One such map of the downtown region of Beijing is shown in Fig. 3, which covers an area of 900 square kilometres and includes more than 70% of the taxis in the trace.

Preprocessing of Nanjing trace and Shanghai trace is carried out in the same way.

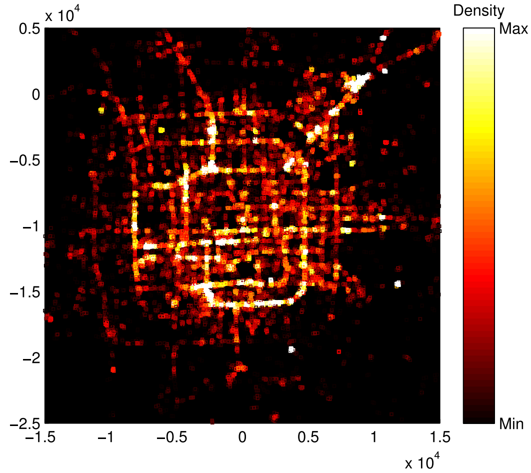


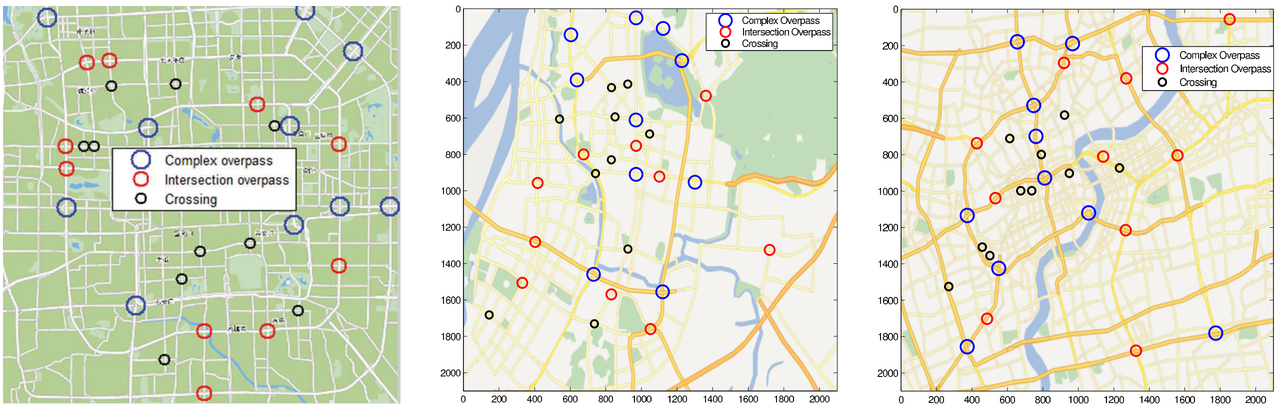
FIGURE 3. Vehicular mobility data processing and visualization: Instantaneous distribution of vehicles in Beijing at 10:00 on May 1, 2009.

C. STATISTICAL METHOD

We now introduce our statistical method for analyzing the data traces. Clearly, the resident number of vehicles, the patterns of staying time, the incoming and outgoing processes all influence the computation capacity of a vehicular mobile cloudlet. Again take Beijing trace as an example. By observing the traffic loads in Beijing, we find that vehicles mainly concentrate in the area within the Fifth Ring Road, and they are always congested at intersections despite the vast scale of this central area. Therefore, we divide the intersections in Beijing into three classes, which are complex overpasses, intersection overpasses, and crossings with traffic lights. Then we choose ten intersections with high traffic jams inside the Fifth-Ring-Road area for each class. For each intersection, we set an effective region to study. Specifically, a taxi within a predefined maximum distance from the center of an intersection is assumed to be in the mobile cloudlet. For the three classes, the maximum distances are set to 250 m, 200 m, and 150 m, respectively, which cover most of the usually congested regions of different intersections recorded in the traces. The map of the downtown area of Beijing and the locations of the selected regions are shown in Fig. 4 (a).

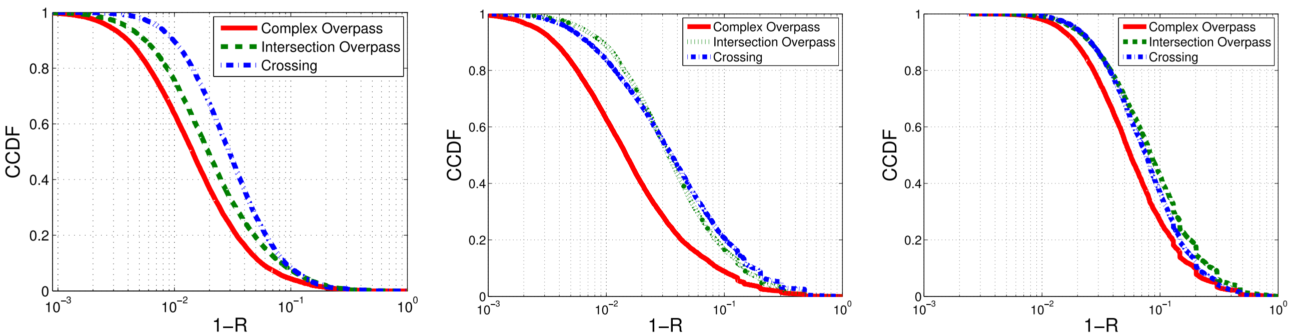
In each minute, the incoming and outgoing information of each vehicle of each effective region can be acquired from the trace. Since the regions are the circles with a radius of 250 m, 200 m or 150 m, we can reasonably assume that all the vehicles inside the circle are connected through radio frequency based on V2V communications. Therefore, the resident number of vehicles, the patterns of staying time, the incoming and outgoing processes can all be acquired.

The same statistical method is applied to Nanjing trace and Shanghai trace. The downtown area of Nanjing and the locations of the selected regions for the three classes are illustrated in Fig. 4 (b), while Fig. 4 (c) depicts the case for Shanghai.



(a) (b) (c)

FIGURE 4. Vehicular mobility data processing and visualization: Locations of the selected regions in: (a) Beijing, (b) Nanjing, and (c) Shanghai.



(a) (b) (c)

FIGURE 5. CCDFs of $1 - R$ for justification of using taxi traces: (a) Beijing trace, (b) Nanjing trace, and (c) Shanghai trace.

D. JUSTIFICATION OF USING TAXI TRACE

Since we use taxis mobility traces to collect statistics, which do not include all types of vehicles in the cities, such as buses and private cars, a natural question to ask is whether these traces are sufficient for such a statistical study.

First, the scales of the traces are sufficiently large. To justify this, we randomly select 10% to 100% of the taxis from the traces and count the average number of taxis in each effective region in every ten minutes. Fig. 5 shows the complementary cumulative distribution functions (CCDFs) of $1 - R$ of Beijing, Nanjing and Shanghai traces, respectively, where R is the correlation coefficient of the percentage of taxis selected and the number of taxis in each region. We observe from Fig. 5 that more than 90% of the coefficients are greater than 0.9 for all the three types of regions. Thus the selection of taxis in just one small area can have sufficient randomness because the traces are sufficiently large.

Second, because of the diversity of customers' demand and the nature of taxis' service to satisfy it, each taxi normally travels through different regions. As shown in Fig. 6, although it only contains traces of 300 taxis of one day from 8:00 to 15:00 in Beijing, these traces cover almost all main streets and areas of the city. Obviously, for a road with greater capacity, which means that the road is wider and has more lanes, the number of passing vehicles is larger. In addition, the passing vehicles are usually from or to different directions. The closer an area is to the city center, the more crowded the vehicles are. Observe that the distributions of taxis illustrated in Figs. 3 and 6 clearly reflect well these city's vehicular traffics.

Ideally, we would prefer to use the vehicular data traces that include buses, private cars and other types of vehicles. Collecting private cars' mobility traces is difficult and have serious legal issues. Like taxis, it is practical to collect bus mobility data. But buses' coverage is much restricted both in space and time and, therefore, bus mobility traces are less useful to extract meaningful statistics. By contrast, taxis have much broader coverage in space and time over a city. Since our taxis traces are sufficiently large, mobility statistics extracted from them are statistically meaningful and represent well a city's vehicular traffic statistics.

Third, again using Beijing as an example, we count the incoming and outgoing numbers of taxis in all the selected regions shown in Fig. 4 (a) at different time periods in a day. We find that the variation trends of these numbers are highly consisted with the behaviors of all vehicles on road.

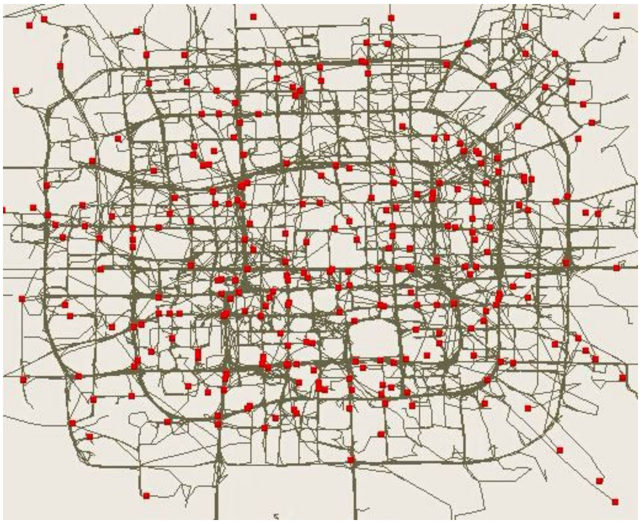


FIGURE 6. Justification of using taxi traces: Traces of 300 taxis of one day from 8:00 to 15:00 in Beijing.

More detailed discussion about the incoming and outgoing numbers of taxis appears in the following section. Since the recorded taxis in the trace are well mixed with other vehicles in the city, it is reasonable to assume that the proportion of the recorded taxis to all the vehicles is approximately the same in every central region of the city. Then we may estimate the total computing capacity of cloudlets by multiplying the capacity of taxi cloudlets by a scaling coefficient.

Based on the above justifications, in the sequel, we will simply use the term vehicle, rather than taxi, in our discussions.

IV. MOBILE CLOUDLETS: MOBILITY MODEL AND COMPUTATION CAPACITY PREDICTION

We first consider the mobility patterns of vehicles based on the trace, study the incoming as well as the outgoing processes of the mobile cloudlets, and obtain the statistics on the staying time and resident number of vehicles. Then, we propose a general mathematical mobility model to describe the whole process for each cloudlet. Finally, we predict the computation capacity of vehicular mobile cloudlet by comparing the results obtained from both the real trace and our model.

A. MOBILITY PATTERNS ANALYSIS

We continue considering all the selected intersections with congested vehicles shown in Fig. 4. First, by selecting three representative intersections of complex overpasses, intersection overpasses, and crossings, we depict the probability density functions (PDFs) of the incoming as well as the outgoing numbers of vehicles per minute, and the PDFs of the resident numbers of vehicles as well as the CCDFs of staying time of vehicles during the whole traces. The results for Beijing and Nanjing traces are shown in Fig. 7 (A) and (B), respectively, where the three rows of the subplots (a), (b) and (c) are related to the three types of intersections, while different distributions plotted with different points in each subplot indicate the data acquired from the time periods of 0:00-24:00, 8:00-24:00, and 0:00-8:00, respectively.

As can be observed from the first three columns of plots in Fig. 7 (A) and (B), the distributions of the incoming and the outgoing numbers of vehicles as well as the resident numbers of vehicles are time varying. By observing the arrival and departure time interval in a short time period, we find that these three variables follow exponential distributions approximately. Thus we use Poisson distributions to fit the numbers of incoming, outgoing and resident vehicles. First, we divide the period of one day into 12 periods of 2 hours and the distribution of each variable in each period is fitted by a Poisson distribution using the maximum likelihood estimate. For example, the estimated parameters of the Poisson distribution for the complex overpass in Beijing, whose statistics are plotted in Fig. 7 (A-a), are shown in Table 2. Then we composite the distributions according to the periods indicated by different colors and plot them with solid lines. We observe that the model fitted distribution matches well with the empirical distribution directly obtained from the trace data, which demonstrates the accuracy of the Poisson process model.

To quantitatively measure the closeness of the model fitted distributions to the empirical ones, we use the proposed Poisson model to fit the empirical distribution for all the 30 intersections in each trace. The goodness of fit for the distribution of each intersection is measured quantitatively by the fitting accuracy defined in the range of 1 to 0. The fitting accuracy is 1 when the fitted distribution is the same as the empirical one, and 0 indicates the minimum fitting accuracy. After obtaining all the fitting accuracy statistics for all the intersections, we plot the aggregated accuracy distributions, i.e., the CCDFs of fitting accuracy of all the intersections for the incoming number, outgoing number and resident number. The results for Beijing and Shanghai traces are shown in Fig. 8 (A) and (B), respectively. In constructing each CCDF from Beijing and Shanghai traces, we have 160 and 130 samples from 16 and 13 days, respectively, and 10 intersections. We observe from Fig. 8 that the accuracies of most fittings are higher than 90% across all the classes of intersections and all the types of variables. This further confirms the accuracy of the proposed Poisson-based model to describe the vehicular mobility patterns.

The right-most column of Fig. 7 shows the CCDFs of staying times. From these three sub-plots, we can observe that the staying time approximately follows the power-law distribution, which indicates that the incoming and outgoing processes are not independent, specifically, vehicles that enter a region earlier tend to exit later.

B. VEHICULAR MOBILITY MODEL FOR MOBILE CLOUDLET

Since the computation capacity can be deduced from vehicular mobility, we propose a mathematical model to characterize it. Consider a JamCloud including N cloudlets, which is denoted by $\mathcal{N} = \{A_1, A_2, \ldots, A_N\}$. We begin with building a

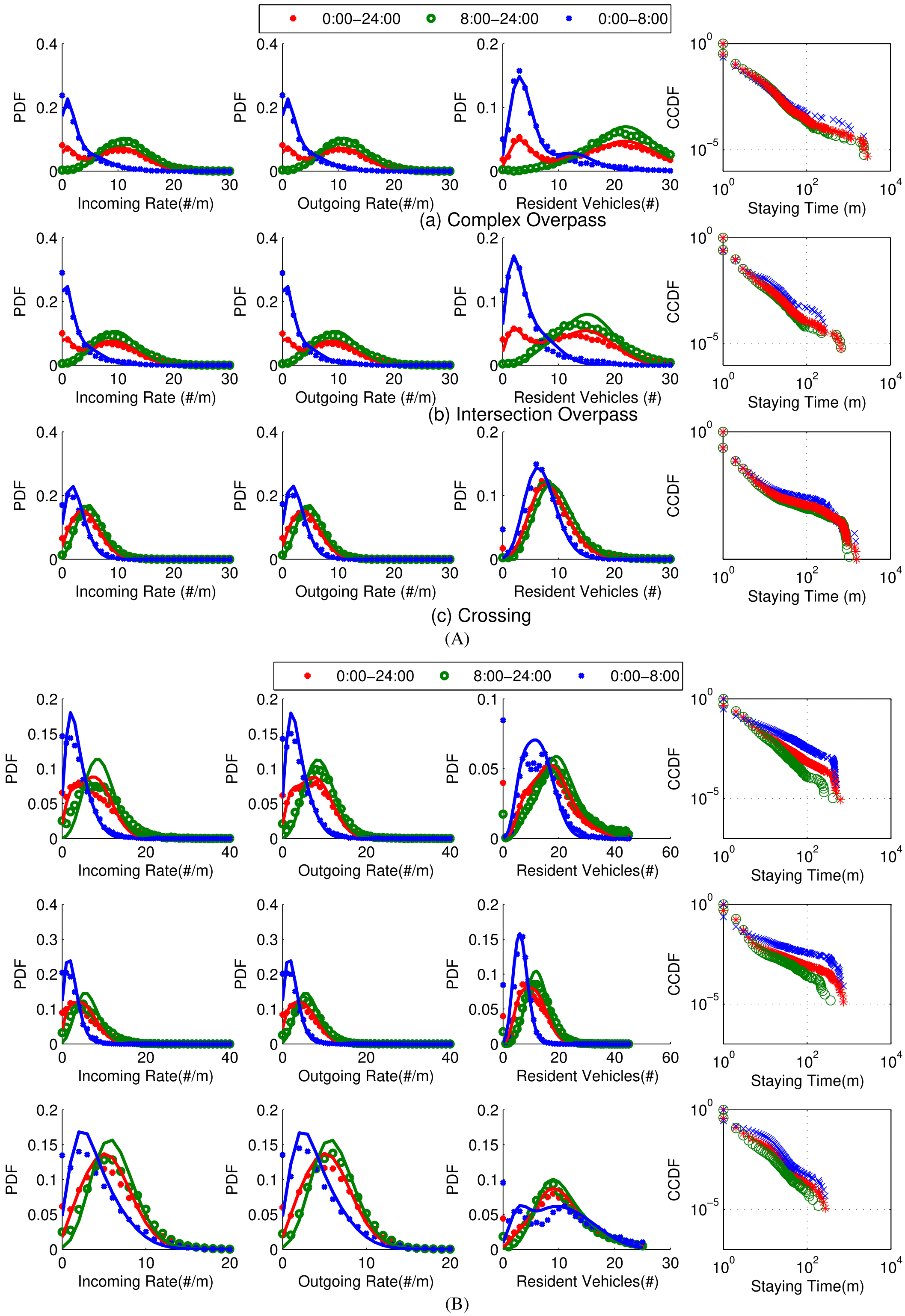


FIGURE 7. Distributions of incoming and outgoing numbers of vehicles per minute, resident numbers of vehicles and staying time (minute) in three representative regions of complex overpasses, intersection overpasses, and crossings, where the points are the empirical results obtained from the trace, while the solid curves are the theoretical results calculated by our proposed model. (A) Beijing trace, and (B) Nanjing trace.

mathematical mobility model to describe the process related to a cloudlet A_n. Based on the fitted Poisson distribution for the incoming number of vehicles per minute, we assume that the incoming process is a periodic non-homogeneous Poisson process with a time-varying rate parameter $\lambda_n(t)$ and a period T of one day. We also assume that there

TABLE 2. Fitting parameters for a complex overpass of beijing.

Period	1	2	3	4	5	6	7	8	9	10	11	12
Mean of Incoming number	2.63	0.97	1.44	6.71	11.62	12.35	11.08	13.28	13.02	11.60	10.63	6.95
Mean of outgoing number	2.66	0.97	1.41	6.59	11.62	12.33	11.10	13.23	13.03	11.62	10.67	7.05
Mean of Resident number	4.74	2.83	3.52	12.95	20.99	22.60	21.23	23.97	26.13	23.51	21.50	12.68

exists a time-varying outgoing rate $\mu_n(t)$ with the same period T. Therefore, the whole process is modeled as a periodic non-homogeneous immigration-death process, denoted by $\{X_n(t)\}$. The resident number of vehicles is modeled as the population while the staying time is modeled as the lifetime of this process. Then we have the following theorem, which matches the fitting models described previously.

Theorem 1: The distributions of the resident number of vehicles and outgoing number of vehicles per minute are both Poisson with mean $E_n(t)$ and $E_n(t)\mu_n(t)$, respectively, where

$$E_n(t)=E_n(t+T)=\int_{-\infty}^{t} \lambda_n(s) \exp\left(-\int_{s}^{t} \mu_n(r)\mathrm{d}r\right) \mathrm{d}s. \quad (2)$$

Proof: The forward Kolmogorov equations of the process are

$$\frac{\mathrm{d}p_i(t)}{\mathrm{d}t} = \lambda_n(t)p_{i-1}(t) + \mu_n(t)(i+1)p_{i+1}(t) - \left(\lambda_n(t) + \mu_n(t)\, i\, p_i(t)\right), \quad (3)$$

with $p_{-1} = 0$. Let $\Phi_n(z,t) = \sum_{i=0}^{+\infty} p_i(t)z^i$, $0 < z \leq 1$, be the probability generating function of the process. We have

$$\frac{\partial \Phi_n(z,t)}{\partial t} = \lambda_n(t)z\Phi_n(z,t) + \mu_n(t)\frac{\partial \Phi_n(z,t)}{\partial z} - \lambda_n(t)\Phi_n(z,t) - \mu_n(t)z\frac{\partial \Phi_n(z,t)}{\partial z}. \quad (4)$$

By solving the above equation we obtain $\Phi_n(z,t) = \exp(z-1)E_n(t)$ and it is the probability generating function of the Poisson distribution with mean $E_n(t)$. We also have

$$\begin{aligned} E_n(t+T) &= \int_{-\infty}^{t+T} \lambda_n(s) \exp\left(-\int_{s}^{t+T} \mu_n(r)\mathrm{d}r\right) \mathrm{d}s \\ &= \int_{-\infty}^{t+T} \lambda_n(s-T) \exp\left(-\int_{s}^{t+T} \mu_n(r-T)\mathrm{d}\right) r)\mathrm{d}s \\ &= \int_{-\infty}^{t} \lambda_n(s) \exp\left(-\int_{s}^{t} \mu_n(r)\mathrm{d}r\right) \mathrm{d}s = E_n(t). \end{aligned} \quad (5)$$

Because the death rate $\mu_n(t)$ is a constant when t is fixed, the distribution of the associated process is also a Poisson distribution with mean $E_n(t)\mu_n(t)$. ■

C. MODEL VALIDATION AND COMPUTATION CAPACITY PREDICTION

We simulate and predict the computing capacity of a mobile cloudlet based on the real traces and the proposed model under the assumption that individual computing capacity of the vehicles follows normal distribution [5]. The mean and standard deviation of computing capacity of vehicles are set to 1 Gflops and 0.3 Gflops, respectively. For each city, we first validate the proposed mobility model via a persuasive approach, in which we randomly choose 50% of the vehicles to train the parameters in the model, and compare the results obtained by the model and real trace. Then, we train the model's parameters again by all the vehicles in the trace, and use it to predict the computation capacity of a cloudlet with real vehicular traffics. Specifically, we choose one of typical most congested complex overpasses to study, and regard the sectionalized statistic parameters $\{\lambda_i, \mu_i | i = 1, 2, \cdots, 12\}$ acquired from the 50% vehicles as the instantaneous parameters of the middle time of the sectionalized time period and estimate the time-variant parameters $\lambda(t)$ and $\mu(t)$ at all times using linear interpolation. Then we make scale transformation for different numbers of vehicles, which means we consider the parameters have a linear relationship with the number of vehicles in the system.

1) BEIJING

Fig. 9 (A-a) shows the results of four days based on Beijing trace and the proposed model with the two parameters that are estimated from 50% of the vehicles. Observe that both the empirical and model based computing capacities have strong periodicity with period of one day, and they are time-variant. More important, the empirical results obtained directly from the real traces are close to those based on the proposed model. Note that our model is periodical with a period of one day and time-variant parameters.

To further quantitatively evaluate the accuracy of our model-based simulation, we calculate the average difference between the two curves in Fig. 9 (A-a) in every 10 minutes, and the CCDFs of the average differences for 0:00-24:00, 8:00-24:00, and 0:00-8:00 time periods are shown in Fig. 10 (a). From the results, we observe that the difference between the computation capacities obtained from the real trace and the model based on the estimated parameters is very small. In more than 80% of time, the difference is smaller than 6 Gflops. Thus, the computation capacity of 100% vehicles of the trace is well predicted by the parameters estimated from 50% of the vehicles.

The above results validate our proposed model for Beijing trace. Then we use this model to predict the computation capacity when there are much more vehicles on road. This is necessary because the number of vehicles on road in reality is much larger than the number of vehicles in the trace. Fig. 9 (A-b) shows the predictions of the computing capacity of the cloudlet under the condition that the total number of vehicles on road in Beijing is 3 million according to the government census [3], which indicates that the potential computing capacity of one cloudlet is really huge.

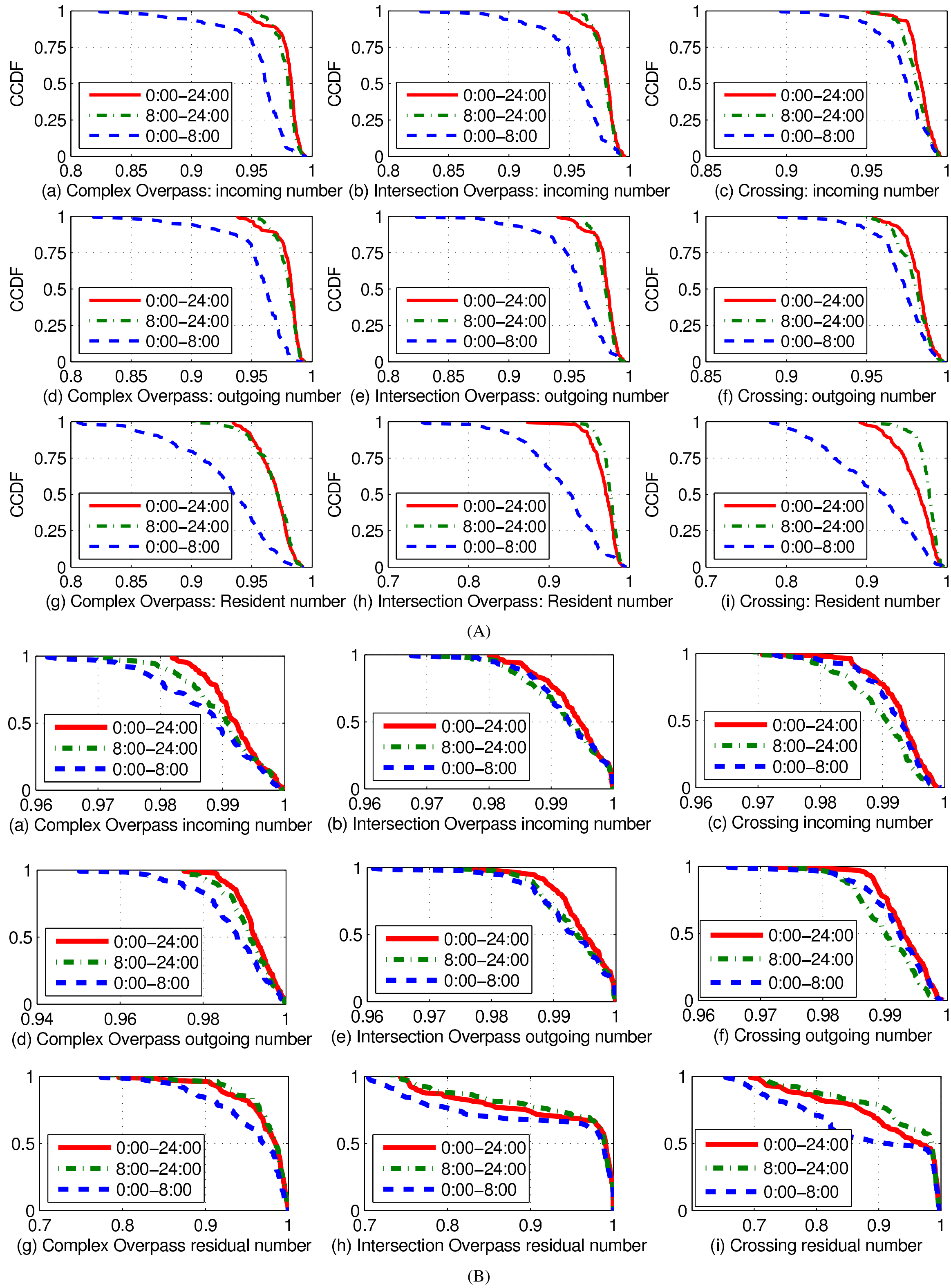


FIGURE 8. **CCDFs of fitting accuracy for the incoming and outgoing rates and the resident numbers of vehicles. (A) Beijing trace, and (B) Shanghai trace.**

We further analyze the predicted computation capacity of Fig. 9 (A-b), Fig. 10 (b) and (c) show the PDFs and CDFs of the distributions of the predicted computation capacities for the time periods of 0:00-24:00, 8:00-24:00, and 0:00-8:00, respectively. From Fig. 10 (b) and (c), we observe that the computation capacity follows

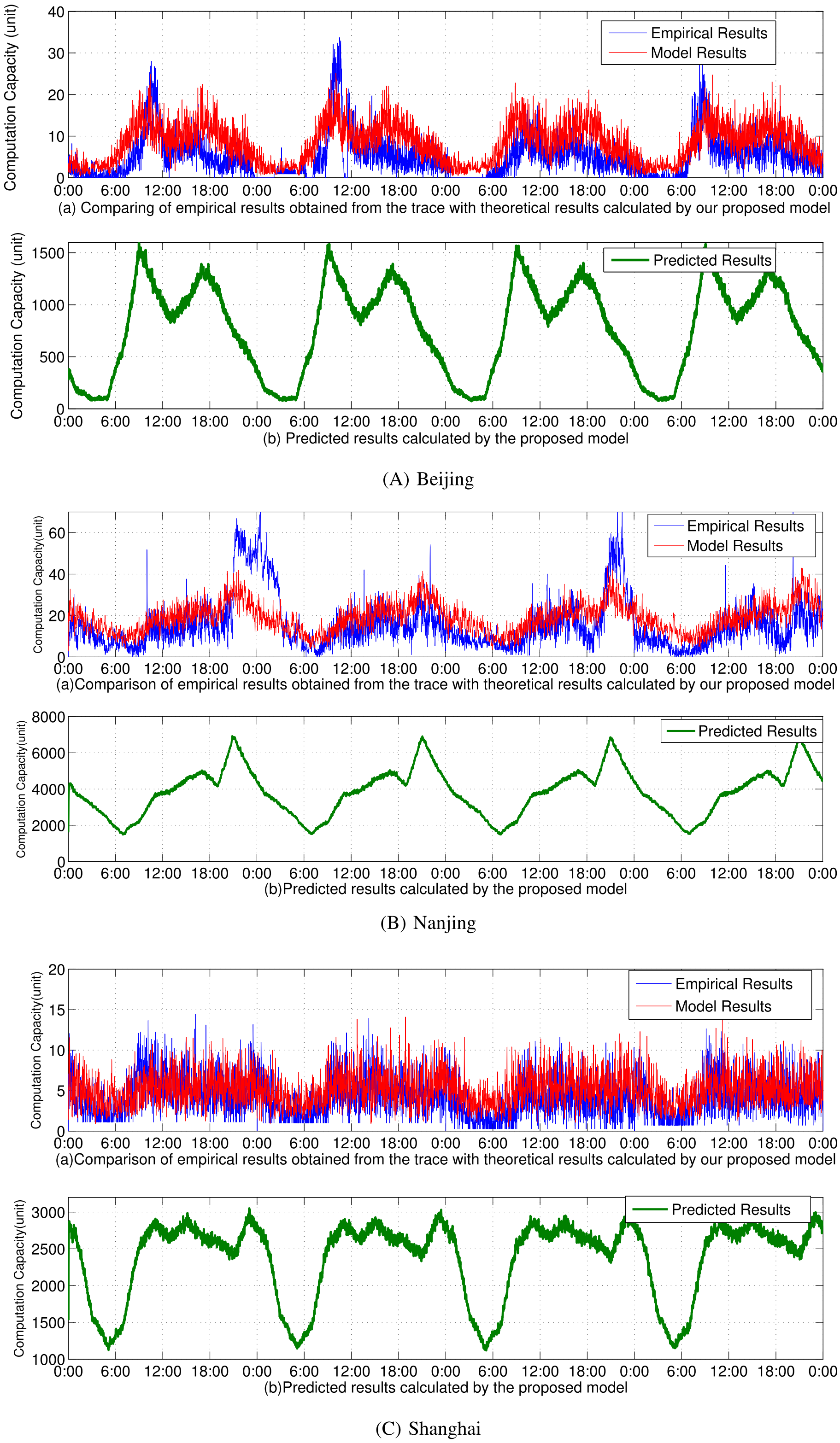


FIGURE 9. Computing capacity validation and prediction: (A) Beijing, (B) Nanjing and (C) Shanghai.

a similar distribution as the resident number of vehicles in a cloudlet, which indicates that the variation of the resident number of vehicles is a key factor to the computation capacity. We also observe that the capacity is usually small during the night and it is large during the day, which reflects the real-life situation. Overall, the computation capacity is larger than 650 Gflops with probability above 60%.

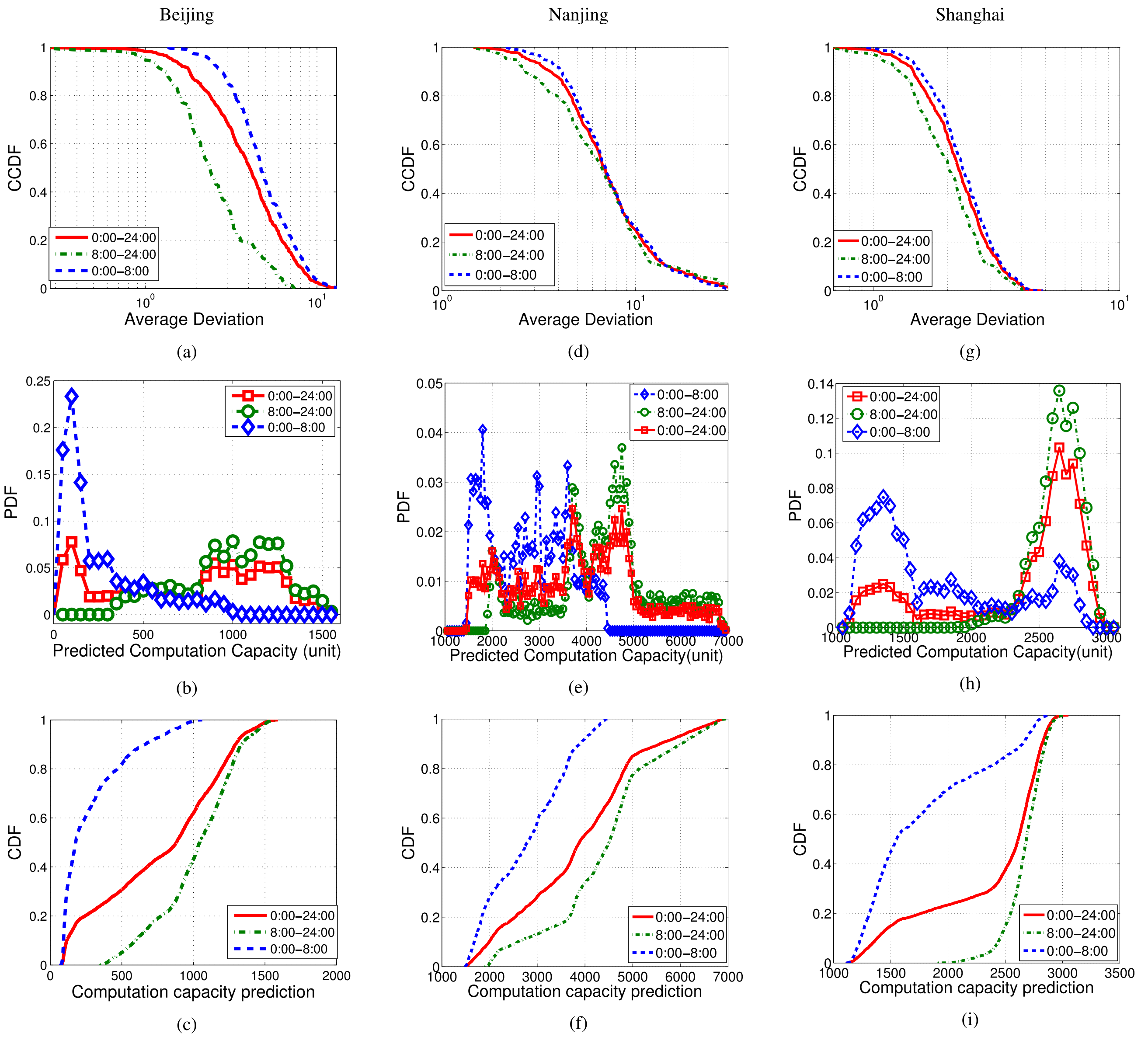


FIGURE 10. **Model validation and prediction of Beijing, Nanjing and Shanghai: (a,d,g) CCDFs of the differences between the two computation capacity simulations (empirical and model) for the time periods of 0:00-24:00, 8:00-24:00 and 0:00-8:00, (b,e,h) PDFs and (c,f,i) CDFs of the predicted computation capacities in time periods of 0:00-24:00, 8:00-24:00 and 0:00-8:00.**

2) NANJING

Fig. 9 (B-a) shows the results of four days based on Nanjing trace. Similar to the results for Beijing trace, we observe that both the empirical and model based computing capacities have strong periodicity of one day period and are time-variant. The empirical results obtained directly from the real traces are still close to those based on the proposed model. The CCDFs of the average differences between the empirical and model curves for 0:00-24:00, 8:00-24:00, and 0:00-8:00 time periods are shown in Fig. 10 (d). From these results, we observe that in more than 60% of time, the difference is smaller than 8 Gflops. Thus, the computation capacity of 100% vehicles of the trace is well predicted by the parameters estimated from 50% of the vehicles.

Then we use this model to predict the computation capacity of the mobile cloudlet in Nanjing where in reality the number of vehicles on road is much larger than the number of vehicles in the trace. Fig. 9 (B-b) shows the predictions of the computing capacity of the cloudlet under the condition that the total number of vehicles on road in Nanjing is 1,574,200, which indicates that the potential computing capacity of the cloudlet in Nanjing is even greater than that in Beijing.

We further analyze the predicted computation capacity of Fig. 9 (B-b), and show the results in Fig. 10 (e) and (f) with the PDFs and CDFs of the distributions of the predicted computation capacities for the time periods of 0:00-24:00, 8:00-24:00, and 0:00-8:00, respectively. From Fig. 10 (e) and (f), we observe again that the computation capacity follows a similar distribution as the resident number of vehicles in a cloudlet, and the capacity in daytime is larger than in nighttime, which reflects the real-life situation. Overall, the computation capacity is larger than 2000 Gflops with probability above 90%.

3) SHANGHAI

Fig. 9 (C-a) shows the results of four days based on Shanghai trace. Again the empirical results obtained directly from the real traces are close to those based on the proposed model. The CCDFs of the average differences between the empirical and model curves in Fig. 9 (C-a) for 0:00-24:00, 8:00-24:00, and 0:00-8:00 time periods are shown in Fig. 10 (g). From these results, we observe that in more than 80% of time, the difference is smaller than 3 Gflops. Fig. 9 (C-b) shows the predictions of the computing capacity of the cloudlet under the condition that the total number of vehicles on road in Shanghai is 2,470,000.

We further analyze the predicted computation capacity of Fig. 9 (C-b), and Fig. 10 (h) and (i) show the PDFs and CDFs of the distributions of the predicted computation capacities for the time periods of 0:00-24:00, 8:00-24:00, and 0:00-8:00, respectively. The results of Fig. 10 (h) and (i) again indicate that the computation capacity follows a similar distribution as the resident number of vehicles in a cloudlet, and the time varying nature of this capacity reflects the real-life situation. Overall, the computation capacity is larger than 2500 Gflops with probability above 60%.

According to our analysis and prediction results of the three traces, we conclude that we can estimate the realistic computation capacity at any time and any intersection using our proposed model as long as a few model parameters have been estimated accurately.

V. JAMCLOUD: SYSTEM MODEL AND PERFORMANCE ANALYSIS

In the previous section, we have modeled the computation capacity of one vehicular mobile cloudlet. The computation capability of the overall JamCloud system, which includes multiple cloudlets in the city connected with the remote cloud, is still unknown. Although the computation capacity of one vehicular mobile cloudlet can be predicted to be very large, so can the computation demand. When more than one vehicular mobile cloudlets are connected with each other, what is the overall performance of the system and how often can the system satisfy the computation demand are key issues, which need to be addressed. Based on our observations in the analysis of vehicular mobile cloudlet, we propose a queueing network model for the whole system and define some appropriate metrics to evaluate the overall performance of satisfying computation demand. Then we carry out the analysis and simulations using this model to study the system asymptotic performance.

A. SYSTEM MODEL FOR JAMCLOUD

Consider the JamCloud system with N vehicular mobile cloudlets, denoted by A_n, $n \in \{1, 2, \cdots, N\}$. Assume that there are M vehicles, denoted by v_m, $m \in \{1, 2, \ldots, M\}$, traveling around different cloudlets. When a vehicle moves in a city, it travels along a road and comes across an intersection region of cloudlet. It may be congested for some time at the intersection, and then travels to another road to enter the region of another cloudlet. Consequently, we can use a queueing network to model the system. This queueing network model includes N server nodes with infinite queue size, which models the N mobile cloudlets in the system. The servers are also denoted by set $\mathcal{N} = \{A_1, A_2, \cdots, A_N\}$. As mentioned in Section IV, the vehicles move into a cloudlet with certain rate, stay in the cloudlet for a while, transfer to another cloudlet, and finally leave the system. These actions are consistent with the packet transmission in the queueing network. Thus we may view the whole process as: the vehicles enter the system with certain rate, stay in the queue of the server, then transfer to other servers, and finally the vehicles leave the system.

Therefore, we model the vehicular mobility system as an open queueing system. If this open queueing system is an open Jackson network, then we can further rely on the well-known results for user distribution and waiting time in open Jackson networks [32] to analyze our JamCloud system. To relate our vehicular mobility system to the Jackson network, we need to demonstrate that the exogenous arrival to each server follows Poisson process. If this property holds, the queueing network can be modeled by a network with infinite server queue, that is, in this queueing network of infinite queue size, the arrival process is a Markov process and the service process is a general process. Thus, we need to study the property of the exogenous arrival rate in the system. By analyzing Beijing trace, we find that the actual exogenous arrival process of the vehicular mobility matches well with the exponential distribution. Thus, the vehicular mobility system can be modeled as an open Jackson network. Based on the model proposed in Section IV, the marginal distribution of resident number of vehicles for individual server A_n, can be expressed as $P(W_n = w_n) = {E_n}^{w_n} e^{-E_n}/w_n!$, where the random variable W_n represents the number of resident vehicles in A_n. Therefore, according to the Jackson networks [32], for the vehicular mobility system, the joint distribution of the resident number of vehicles in the JamCloud system of $\{A_1, A_2, \cdots, A_N\}$ at time t is

$$\pi(\boldsymbol{w}, t) = \prod_{j=1}^{N} \frac{E_j(t)^{w_j} e^{-E_j(t)}}{w_j!}, \tag{6}$$

where $\boldsymbol{w} = [w_1\ w_2 \cdots w_N]^T$, and we have use the random vector $\boldsymbol{W} = [W_1\ W_2 \cdots W_N]^T$ to represent the numbers of resident vehicles in the N mobile cloudlets.

B. SYSTEM PERFORMANCE METRICS

Based on the above queueing network model, we derive the closed-form expressions for performance metrics, including the probability of cloud-wide satisfaction and the average number of mobile cloudlets that satisfy their computation demands, with the two kinds of cloud computing resources of local cloudlet and remote cloud.[1] We say a cloudlet is in the computationally satisfactory state if it has sufficient computation resources to meet all its computation demands.[2] When the JamCloud is in the computationally satisfactory state, all the cloudlets in the JamCloud are in the computationally satisfactory state. For a JamCloud, it may be difficult to enjoy the computationally satisfactory state all the time. Hence, the metric of the steady-state probability that a JamCloud is in the computationally satisfactory state is an important parameter to evaluate the performance of the system. Another important metric is the expected number of cloudlets that are enjoying the computationally satisfactory state. We now provides more specific definitions of these two metrics.

The remote computation capacity provided by the remote cloud via the BS or RSU in cloudlet A_n is denoted by r_n. Because the allocation of r_n cannot be adjusted by the remote control center instantaneously, in reality, r_n remains constant in a short time period at least. Clearly, a constant resource cannot match the strong mobility dynamics of vehicles. However, owing to the statistical features that the number of vehicles in a cloudlet has an approximately stationary distribution in such a short time period, this distribution can be utilized to evaluate the system performance. First we define the computation capacity index for cloudlet A_n, denoted by I_n, as follows

$$I_n = \frac{r_n + q_n(W_n)}{d_n(W_n)}, \tag{7}$$

where W_n is the number of vehicles in A_n, $q_n(W_n)$ is the computation capacity of the local mobile cloudlet A_n, and $d_n(W_n)$ is the computation demand of all the vehicles in A_n which includes the computation tasks assigned to A_n by the BS or RSU. Based on I_n, the probability that A_n is in the computationally satisfactory state can be defined as

$$P_n^S = \Pr(I_n \geq 1), \tag{8}$$

while the cloud-wide probability that the overall JamCloud system is in the computationally satisfactory state can be defined as

$$P^S = \Pr(I_n \geq 1, n = 1, 2, \cdots, N). \tag{9}$$

We also define the average number of cloudlets that are enjoying the computationally satisfactory state as

$$N^S = \sum_{n=1}^{N} \Pr(I_n \geq 1). \tag{10}$$

We now calculate the satisfying probability P_n^S of A_n. Assume that there are K classes of vehicles in each cloudlet. The vehicles in the kth class have the same computation capacity of b^k and the number of vehicles W_n^k in the kth class of A_n follows Poisson distribution with mean E_n^k. We further assume that each vehicle in A_n needs the computation capacity of c_n. Then the satisfying probability of A_n is given by

$$\begin{aligned} P_n^S &= \Pr\left(r_n + \sum_{k=1}^{K} b^k W_n^k \geq \sum_{k=1}^{K} W_n^k c_n\right) \\ &= \sum_{w_n^1=0}^{\infty} \cdots \sum_{w_n^K=0}^{\infty} P(f(\boldsymbol{w}_n) \leq r_n) \prod_{k=1}^{K} P(W_n^k = w_n^k) \\ &= \sum_{f(\boldsymbol{w}_n)\leq r_n, \forall k, 0 \leq w_n^k} \prod_{k=1}^{K} \frac{\left(E_n^k\right)^{w_n^k}}{w_n^k!} e^{-\sum_{k=1}^{K} E_n^k}, \end{aligned} \tag{11}$$

where $f(\boldsymbol{w}_n) = \sum_{k=1}^{K} \left(c_n - b^k\right) w_n^k$. Consequently, we can obtain the cloud-wide satisfying probability P^S and the number of satisfied cloudlets N^S respectively as follows

$$P^S = \prod_{n=1}^{N} \Pr(I_n \geq 1) = \prod_{n=1}^{N} P_n^S, \tag{12}$$

$$N^S = \sum_{n=1}^{N} \Pr(I_n \geq 1) = \sum_{n=1}^{N} P_n^S. \tag{13}$$

C. ASYMPTOTIC ANALYSIS

Next we analyze the asymptotic performance of JamCloud, i.e., when the number of vehicles M tends to infinity. Specifically, we ask the question that under what conditions, the probability of any cloudlet, A_n, satisfying the computation capacity is high. We consider this question in two scenarios.

1) HOMOGENEOUS VEHICLES

In this case, there is only one class of vehicles in the system. We have the normal distribution as an approximation to the Poisson distribution if $M \to \infty$, which means $E_n^1 \to \infty$. Thus the satisfying probability of A_n is given by

$$\begin{aligned} P_n^S &= \lim_{E_n^1 \to \infty} \sum_{0 \leq w_n^1, (c_n - b^1) w_n^1 \leq r_n} \frac{\left(E_n^1\right)^{w_n^1}}{w_n^1!} e^{-E_n^1} \\ &= \begin{cases} \lim\limits_{E_n^1 \to \infty} 1 - \Phi\left(\dfrac{r_n}{c_n - b^1}\right) = 1, & \text{if } c_n \leq b^1, \\ \lim\limits_{E_n^1 \to \infty} \Phi\left(\dfrac{r_n}{c_n - b^1}\right) = 0, & \text{if } c_n > b^1. \end{cases} \end{aligned} \tag{14}$$

where $\Phi(x)$ is the cumulative distribution function (CDF) of the normal distribution with both mean and variance as E_n^1.

2) HETEROGENEOUS VEHICLES

When there are K classes of vehicles, by denoting $\frac{E_n^k}{\sum_{k=1}^{K} E_n^k} = r^k$ and similar to the homogeneous case, we have the

[1] We do not include the computational resources of BSs and/or RSUs.

[2] Computation demands of a cloudlet include the computation requirements of all the individual vehicles in the cloudlets and the computation jobs assigned by the BS or RSU to the cloudlet.

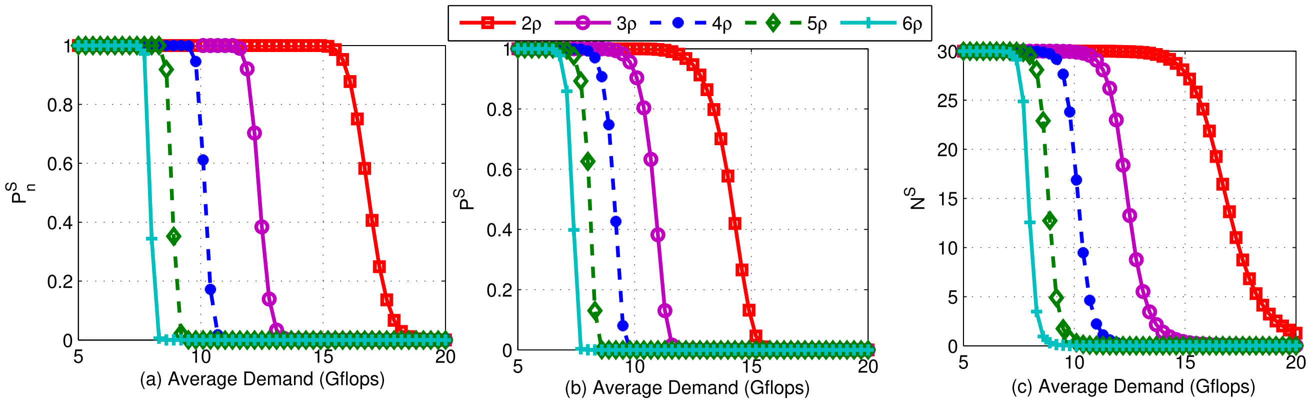


FIGURE 11. System performance of JamCloud as functions of computation demand: (a) cloudlet satisfying probability, (b) cloudlet-wide satisfying probability, and (c) number of satisfied cloudlets.

satisfying probability of cloudlet A_n given by

$$P_n^S = \lim_{\boldsymbol{E}_n \to \infty} \int_{\sum_{k=1}^{K} \left(c_n - b^k\right) w_n^k \le r_n} \prod_{k=1}^{K} \frac{1}{\sqrt{2\pi E_n^k}} e^{\frac{\left(w_n^k - E_n^k\right)^2}{2E_n^k}} \mathrm{d}\boldsymbol{w}_n$$
$$= \begin{cases} 1, & \text{if } c_n \le \sum_{k=1}^{K} r^k b^k, \\ 0, & \text{if } c_n > \sum_{k=1}^{K} r^k b^k, \end{cases} \tag{15}$$

where $\mathrm{d}\boldsymbol{w}_n = \prod_{k=1}^{K} \mathrm{d}w_k$ and $\boldsymbol{E}_n \to \infty$ means that $E_n^k \to \infty$ for $k = 1, 2, \cdots, K$.

The above analysis results tell us that when the number of vehicles or cloudlets in the system is very large and the remote computation capacity is not infinite, the performance of a cloudlet mainly depends on whether the local computation capacity can satisfy the demands. In this case, the allocation of resources mainly happens inside the local cloudlet.

D. PERFORMANCE SIMULATION

Based on the above derivation, we setup a JamCloud environment to observe the performance. There are 30 mobile cloudlets in this simulated system. The allocated remote computation capacity to each cloudlet is proportional to the expectation of the number of vehicles in each cloudlet and the total remote capacity is 50,000 Gflops. The vehicles are divided into two classes, with the computation capacity of individual vehicles in the two classes being 2 Gflops and 5 Gflops, respectively. We choose these parameters because the computation capacity of a single core CPU is usually several Gflops and a remote computing center may have tens of thousands times computation capacity of a single core CPU.[3] Related to the number of vehicles in each cloudlet, we use the selected 30 intersections in the mobility trace of Beijing, and choose the number of vehicles in the two classes as the average number of taxis from 10:00 to 12:00 in each cloudlet. We set this value as ρ and extend it by 2, 3, 4, 5 and 6 times for simulation, because in the real life, the number of taxis is only a fraction of all the vehicles in each cloudlet. We also set the mean of the computation demand of vehicles to be the same in each cloudlets. Under the above settings, we obtain the JamCloud system performance of cloudlet satisfying probability, cloud-wide satisfying probability and the number of satisfied cloudlets.

By varying the mean of the computation demands of vehicles, the results of the cloudlet satisfying probability P_n^S are shown in Fig. 11 (a), while Fig. 11 (b) and Fig. 11 (c) depict the results of the cloud-wide satisfying probability P_S and the number of satisfied cloudlets N^S, respectively. From Fig. 11, we can observe that when the average number of vehicles in each cloudlet is 6ρ, the cloudlet satisfying probability is near 100% for the average demand less than 6 Gflops, and the cloud-wide satisfying probability is almost 100% for the average demand less than 4 Gflops, while almost every cloudlet in the system is satisfied for the average demand less than 5 Gflops. By contrast, given that the average number of vehicles in each cloudlet is 2ρ, P_n^S is 100% for the average demand less than 15 Gflops, and P_S is almost 100% for the average demand less than 11 Gflops, while $N^S \approx 30$ for the average demand less than 14 Gflops. It can also be seen that with the increase of demand, the satisfying probability decreases, and the larger the average number of vehicles, the sharper the decreasing rate. From these results, we can decide how much remote computation capacity a cloudlet need according to the specific requirements. We can also use these results to design the system and to allocate remote computation capacity to mobile cloudlets.

We also simulate the system performance by varying the computation capacity of remote cloud computing centers. In this simulation, we set the mean of the computation demand of vehicles to 10 Gflops and the results of the three metrics are shown in Fig. 12. As expected, with the increase of the computation capacity of remote cloud, all

[3] In real world, some vehicles, such as buses, may have higher computation capacity. Also in near future, many cars on road will be electric and they have higher computation capacity. Consequently, computation capacity of cloudlets will be significantly higher and we will require less remote cloud capacity to meet the demands.

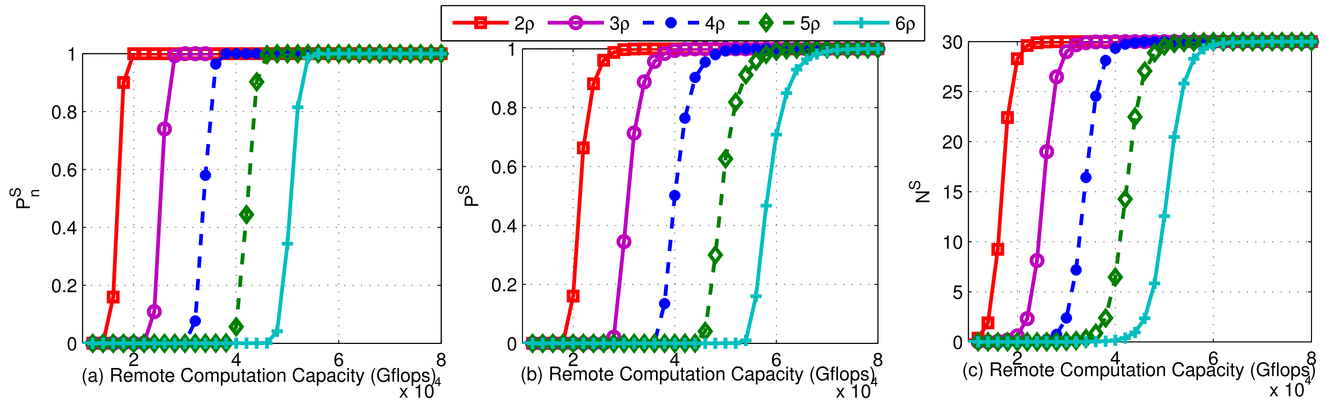


FIGURE 12. System performance of JamCloud as functions of remote computation capacity: (a) cloudlet satisfying probability, (b) cloudlet-wide satisfying probability, and (c) number of satisfied cloudlets.

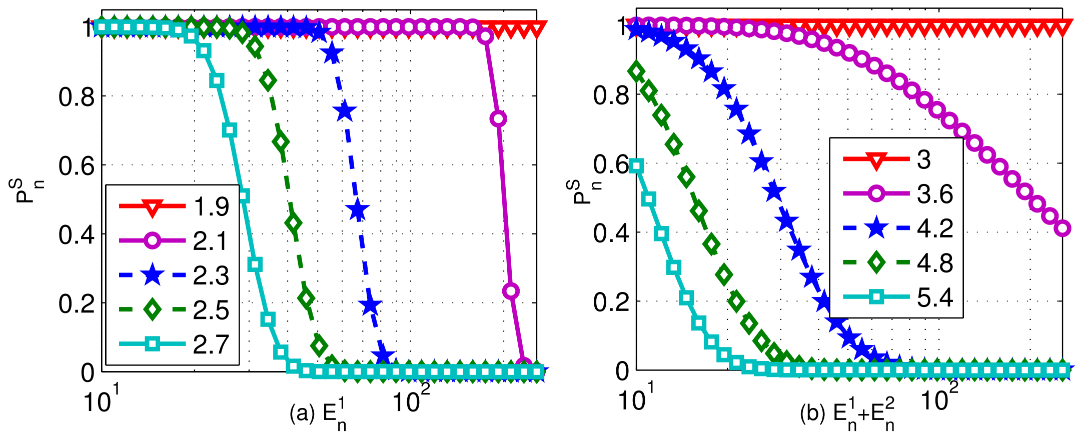


FIGURE 13. Asymptotic performance of JamCloud: (a) $K = 1$, computation capacity per vehicle is 2 Gflops, and remote computation capacity is 20 Gflops, and (b) $K = 2$, computation capacity for two-class vehicles are 2 Gflops and 5 Gflops, respectively, and remote computation capacity is 20 Gflops. In both (a) and (b), different curves are with different values of average number of vehicles in each cloudlet.

the three metrics increase. Observe that the satisfying probability increases sharply from 0 to 1, indicating that there exists a critical value of remote computation capacity for the JamCloud to achieve the computationally satisfactory state. In other words, given the specific settings of mobile cloudlets, the performance of the system will be seriously limited if the capacity of remote cloud is under some critical value. Therefore, an optimized allocation protocol needs to be carefully designed for attaining the achievable system performance.[4]

Next we carry out simulations on asymptotic performance of mobile cloudlets with different classes of vehicles K, and the results of P_n^S are shown in Fig. 13. From Fig. 13, we observe that depending on the relationship between local computation demand and capacity, P_n^S reduces from 1 to with the increase of the number of vehicles, which complies with our asymptotic analysis. This further demonstrates the correctness of our theoretical analysis.

[4]As aforementioned, in the near future, computational capacity of individual vehicles are expected to be much higher than the values used in this simulation. Consequently, the computational capacity of individual mobile cloudlets will be significantly higher. The remote cloud capacity will become less important to the system performance. In fact, in most situations, we may not need remote cloud at all.

VI. CONCLUSION

In this paper, we have proposed and evaluated JamCloud, a system that collect and aggregate computation resources from local congested vehicles and remote cloud computing centers to better satisfy the computation demand of vehicles. We have modeled the system based on the model of stochastic process and a queueing network. We have also simulated and analyzed the computation capacity of mobile cloudlets, the overall and asymptotic performance of the system. We have observed that as a time-variant system, the computation capacity of a vehicular mobile cloudlet depends on the resident number of vehicles. Given limited computation resources of vechicle, we have found that the probability of satisfying computation demand decreases sharply if local computation capacity is insufficient and the remote computation capacity is lower than a certain critical value. These conclusions help us to design practical JamCloud system.

With the ongoing revolution of making vehicles on road electric and the coming 5G, JamCloud's time has come. In near future, the computation capacity of individual local vehicle cloudlet will be substantially higher. Consequently, even without the aid of remote cloud, a local mobile cloudlet not only is capable of meeting the demands of all the local vehicles but also has sufficiently large unused computation resources. These extra computation resources can naturally be utilized to support the local 5G BS or RSU's baseband computational tasks. It is worth highlighting that the dynamics of local mobile cloudlet's capacity shown in Fig. 9 match well with the demands of local 5G baseband signal processing tasks. Therefore, JamCloud offers an ideal means for aiding green 5G communications in city by harvesting the resources of congested vehicles, which otherwise would be wasted.

REFERENCES

[1] *The Survey of Number of Vehicles in Use Worldwide*. Accessed: Mar. 26, 2019. [Online]. Available: https://www.statista.com/statistics/281134/number-of-vehicles-in-use-worldwide/

[2] *International Car Sales Since 1990*. Accessed: Mar. 26, 2019. [Online]. Available: https://www.statista.com/statistics/200002/international-car-sales-since-1990/

[3] *Beijing is Still Struggling Deal Traffic Congestion*. Accessed: Aug. 22, 2013. [Online]. Available: http://www.scmp.com/news/china/article/1298559/beijing-still-struggling-deal-traffic-congestion/

[4] *INRIX 2018 Global Traffic Scorecard*. Accessed: Mar. 26, 2019. [Online]. Available: http://inrix.com/scorecard/

[5] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *J. Netw. Comput. Appl.*, vol. 40, pp. 325–344, Apr. 2014.

[6] A. Jolley. *The Rise of Electric Cars: An in-Depth Look at a Motoring Revolution*. Accessed: Mar. 26, 2019. [Online]. Available: https://www.confused.com/car-insurance/electric-cars-report

[7] T. H. Silva, P. O. S. Vaz de Melo, A. C. Viana, J. M. Almeida, J. Salles, and A. A. F. Loureiro, "Traffic condition is more than colored lines on a map: Characterization of Waze alerts," in *Proc. SocInfo*, Kyoto, Japan, Nov. 2013, pp. 309–318.

[8] Statista. *5G—Statistics & Facts*. Accessed: Mar. 26, 2019. [Online]. Available: https://www.statista.com/topics/3447/5g/

[9] P. Rost, C. J. Bernardos, A. De Domenico, M. Di Girolamo, M. Lalam, A. Maeder, D. Sabella, and D. Wübben, "Cloud technologies for flexible 5G radio access networks," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 68–76, May 2014.

[10] J. Bartelt, P. Rost, D. Wubben, J. Lessmann, B. Melis, and G. Fettweis, "Fronthaul and backhaul requirements of flexibly centralized radio access networks," *IEEE Wireless Commun.*, vol. 22, no. 5, pp. 105–111, Oct. 2015.

[11] M. Peng, Y. Wang, T. Dang, and Z. Yan, "Cost-efficient resource allocation in cloud radio access networks with heterogeneous fronthaul expenditures," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4626–4638, Jul. 2017.

[12] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Services Appl.*, vol. 1, no. 1, pp. 7–18, May 2010.

[13] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comput.*, vol. 13, no. 18, pp. 1587–1611, Dec. 2013.

[14] M. Shiraz, A. Gani, R. H. Khokhar, and R. Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1294–1313, 3rd Quart., 2013.

[15] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generat. Comput. Syst.*, vol. 29, no. 1, pp. 84–106, 2013.

[16] E. E. Marinelli, "Hyrax: Cloud computing mobile devices using mapreduce," M.S. thesis, Dept. Electron. Eng., Carnegie Mellon Univ., Pittsburgh, PA, USA, 2009.

[17] G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in *Proc. 1st ACM Workshop MCS*, San Francisco, CA, USA, Jun. 2010, pp. 1–5.

[18] N. Fernando, S. W. Loke, and W. Rahayu, "Dynamic mobile cloud computing: Ad hoc and opportunistic job sharing," in *Proc. 4th IEEE UCC*, Melbourne, NSW, Australia, Dec. 2011, pp. 281–286.

[19] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, "Serendipity: Enabling remote computing among intermittently connected mobile devices," in *Proc. ACM MobiHoc*, Hilton Head Island, SC, USA, Jun. 2012, pp. 145–154.

[20] E. Koukoumidis, D. Lymberopoulos, K. Strauss, J. Liu, and D. Burger, "Pocket cloudlets," in *Proc. ASPLOS*, Newport Beach, CA, USA, Mar. 2011, pp. 1–13.

[21] Y. Li and W. Wang, "Can mobile cloudlets support mobile applications?" in *Proc. INFOCOM*, Toronto, QC, Canada, Apr./May 2014, pp. 1060–1068.

[22] S. Olariu and M. C. Weigle, Eds., *Vehicular Networks: From Theory to Practice*. Boca Raton, FL, USA: CRC Press, 2010.

[23] J. A. Torkestani, "Mobility prediction in mobile wireless networks," *J. Netw. Comput. Appl.*, vol. 35, no. 5, pp. 1633–1645, Sep. 2012.

[24] Y. Qin, D. Huang, and X. Zhang, "VehiCloud: Cloud computing facilitating routing in vehicular networks," in *Proc. TrustCom*, Liverpool, U.K., Jun. 2012, pp. 1438–1445.

[25] N. Liu, M. Liu, W. Lou, G. Chen, and J. Cao, "PVA in VANETs: Stopped cars are not silent," in *Proc. INFOCOM*, Shanghai, China, Apr. 2011, pp. 431–435.

[26] F. Malandrino, C. Casetti, C.-F. Chiasserini, C. Sommer, and F. Dressler, "The role of parked cars in content downloading for vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 9, pp. 4606–4617, Nov. 2014.

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

[28] P. Hui, T. Wu, and C. Peylo, "System for constructing stopped vehicle-infrastructure communication network," U.S. Paten 9 996 993 B2, Jun. 12, 2018.

[29] C. Wang, Y. Li, D. Jin, and S. Chen, "On the serviceability of mobile vehicular cloudlets in a large-scale urban environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2960–2970, Oct. 2016.

[30] J. Shen, X. Liu, and M. Chen, "Discovering spatial and temporal patterns from taxi-based floating car data: A case study from Nanjing," *GISci. Remote Sens.*, vol. 54, no. 5, pp. 617–638, Mar. 2017.

[31] H. Zhu, M. Li, L. Fu, G. Xue, Y. Zhu, and L. M. Ni, "Impact of traffic influxes: Revealing exponential intercontact time in urban VANETs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 8, pp. 1258–1266, Aug. 2011.

[32] J. Walrand and P. Varaiya, "Sojourn times and the overtaking condition in Jacksonian networks," *Adv. Appl. Probab.*, vol. 12, no. 4, pp. 1000–1018, Dec. 1980.

XUEFENG XIAO was born in 1979. He received the Ph.D. degree in computer science from Central South University, in 2014. He was a Postdoctoral Researcher with Beijing Jiaotong University, in 2017. He is currently a Teacher with the School of Economics and Management, Beijing Information Science and Technology University. His research interests include Internet QoS, wireless networks, and big data fields.

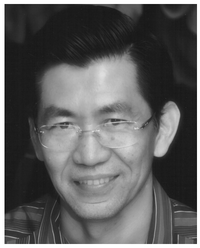
XUESHI HOU received the B.Eng. degree in electronic engineering from Tsinghua University, Beijing, China, in 2015. She is currently pursuing the Ph.D. degree with the University of California San Diego, La Jolla, CA, USA. Her research interests include multimedia, virtual reality, and wireless communications.

CHUANMEIZI WANG is currently pursuing the bachelor's degree in electronic engineering with Tsinghua University. His research interests include vehicular communications and networking.

YONG LI received the B.S. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2007, and the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2012.

He was a Visiting Research Associate with Telekom Innovation Laboratories and The Hong Kong University of Science and Technology, in 2012 and 2013, respectively. From 2013 to 2014, he was a Visiting Scientist with the University of Miami, Coral Gables, FL, USA. He is currently a Faculty Member of the Department of Electronic Engineering, Tsinghua University. His research is granted by the Young Scientist Fund of the Natural Science Foundation of China, the Postdoctoral Special Fund of China, and industrial companies of Hitachi and ZET. His research interests are in the areas of networking and communications, including mobile opportunistic networks, device-to-device communication, software-defined networks, network virtualization, and the future Internet. He has served as a Technical Program Committee (TPC) Chair for the WWW workshop of Simplex 2013 and the TPC member for several international workshops and conferences. He was a recipient of the Outstanding Postdoctoral Researcher, Outstanding Ph.D. Graduates, and Outstanding Doctoral Thesis awards from Tsinghua University. He is currently an Associate Editor of the *EURASIP Journal on Wireless Communications and Networking*. He is a Guest Editor of the *ACM Springer Mobile Networks and Applications (MONET) Journal*, and the Special Issue on the *Software-Defined and Virtualized Future Wireless Networks*.

PAN HUI received the B.Eng. and M.Phil. degrees from the Department of Electrical and Electronic Engineering, The University of Hong Kong, and the Ph.D. degree from the Computer Laboratory, University of Cambridge. He is currently a Faculty Member with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, where he directs the System and Media Laboratory. He also serves as a Distinguished Scientist of Telekom Innovation Laboratories (T-labs), Germany, and an Adjunct Professor of social computing and networking with Aalto University, Finland. Before returning to Hong Kong, he has spent several years in T-labs and Intel Research Cambridge. He has published more than 100 research papers and has several granted and pending European patents. He has founded and chaired several IEEE/ACM conferences/workshops, and has served on the technical program committee of numerous international conferences and workshops, including the IEEE Infocom, SECON, MASS, Globecom, WCNC, and ITC.



SHENG CHEN (M'90–SM'97–F'08) received the B.Eng. degree in control engineering from the East China Petroleum Institute, Dongying, China, in 1982, the Ph.D. degree in control engineering from the City, University of London, in 1986, and the D.Sc. degree from the University of Southampton, Southampton, U.K., in 2005. From 1986 to 1999, he held research and academic appointments with The University of Sheffield, U.K., The University of Edinburgh, U.K., and the University of Portsmouth, U.K. Since 1999, he has been with the School of Electronics and Computer Science, University of Southampton, U.K., where he holds the position of Professor in intelligent systems and signal processing. He has published over 650 research papers. He has over 13 000 Web of Science citations and over 27 000 Google scholar citations. His research interests include adaptive signal processing, wireless communications, modeling and identification of nonlinear systems, neural networks and machine learning, intelligent control system design, and evolutionary computation methods and optimization. He is a Fellow of the United Kingdom Royal Academy of Engineering, a Fellow of IET, a Distinguished Adjunct Professor with King Abdulaziz University, Jeddah, Saudi Arabia, and an original ISI Highly Cited Researcher in engineering (2004).