IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

SPECIAL SECTION ON SUSTAINABLE INFRASTRUCTURES, PROTOCOLS, AND RESEARCH CHALLENGES FOR FOG COMPUTING

Received February 2, 2019, accepted February 28, 2019, date of publication March 12, 2019, date of current version April 1, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2903302*

# vFog: A Vehicle-Assisted Computing Framework for Delay-Sensitive Applications in Smart Cities

**SYED SARMAD SHAH[1], MUHAMMAD ALI[1], ASAD WAQAR MALIK[1,2], MUAZZAM A. KHAN[1], AND SRI DEVI RAVANA[2]**
[1]School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[2]Department of Information System, Facutly of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia

Corresponding author: Asad Waqar Malik (asad.malik@seecs.edu.pk)

**ABSTRACT** The inception of the smart cities concept provides a compelling platform to support innovative applications. It provides distinctive view of cities, where mobile devices, pedestrians, and electronic gadgets can communicate with each other to build an effective urban environment to further improve the living standards. Similarly, the role of the Internet of Things (IoT) and vehicular computing has emerged due to smart cities. This is further complemented by edge and fog computing architectures. The emerging concept of vehicular fog computing has enabled the platform to support delay-sensitive applications and to reduce the workload on the backend networks. Vehicular fog computing is a paradigm that touches the boundaries of thinking vehicles as an infrastructures-as-a-service. The use of vehicles to provide computation on-the-move poses various challenges. The vehicles with onboard computing equipment can facilitate delay-sensitive applications. These vehicles can act as an edge device to reduce the load from a backbone network. However, due to continuous mobility, it is difficult to use traditional frameworks to distribute the computation task among vehicles. In this paper, we propose a framework termed vFog. The vFog is designed to provide computing facilities from nearby fog vehicles. The framework utilizes the onboard computing facility of vehicles without the support of roadside units (RSUs). Moreover, the proposed framework handles churn behavior and supports multi-hop communication to improve the task delivery ratio. The proposed framework allows researchers to benchmark their own task distribution algorithms over the dynamic vehicular networks.

**INDEX TERMS** Vehicular fog computing, tasks scheduling policy, edge devices, multi-vehicle relay.

## I. INTRODUCTION

With technology advancement and the inception of smart cities, the concept of smart and connected vehicles are also emerging. The smart vehicles are equipped with onboard units that can compute, store data and communicate with other vehicles either through a short-range communication channel or WiFi. The onboard units are connected with various sensors installed inside the vehicle to monitor the state of the vehicle. The smart vehicles also share information with other vehicles and Road Side Units (RSUs). This information is used to facilitate vehicles in the selection of least congested route to the destination. Moreover, this information is also useful for first responders in emergency situations. Fig. 1 shows the overall vehicular network design, where vehicles communicate with other vehicles and RSUs for information sharing. The vehicles can be used for infotainment applications where other vehicles search for the multimedia contents in its surrounding vehicles. It is beneficial to get the contents from nearby vehicles instead of requesting through the internet. Thus, the vehicles can form a real-time interest group for sharing multimedia contents and serve as a cache. This dynamic network forming feature of smart vehicles has evolved the concept of vehicular computing (VCC). In VCC, it provides an infrastructure where vehicles can produce the contents, store them and share with others on request [1].

Vehicular Fog Computing (VFC) is the recent advancement made to bring computing near to end users. The existing solutions are based on Road Side Units (RSUs) or deployment of computing resources at the edge networks. These solutions are costly and require proper maintenance; therefore, the providers charge heavily based on resource usage. As there are a significantly large number of vehicles in

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

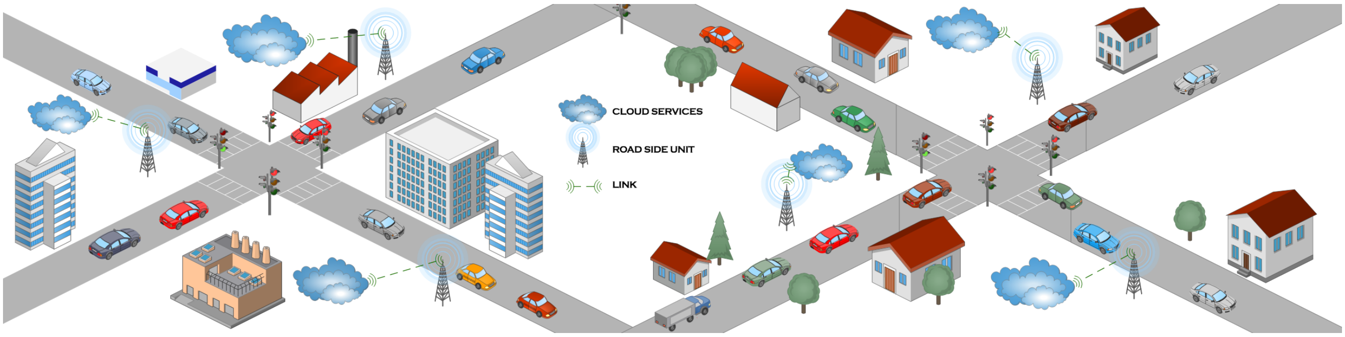

**FIGURE 1.** High level vehicular network with connected RSUs and base stations.

urban areas; using these vehicles to share resources can act as a first tier to support delay-sensitive computing applications. VFC utilizes many onboard sensors that gather real-time data from surroundings. The processed data is shared with other vehicles on-demand. This data is used to make real-time traffic routing decisions.

The VFC is being adopted to address the issues raised in the Vehicular Cloud Computing (VCC). The VCC provides computational and storage capabilities using backend data centers. All the computation requests are sent to data centers for execution and results are shared with the users.

In VFC, the vehicles can provide a computation layer to support delay-sensitive applications. The vehicles send their requests to mobile fog vehicles and utilize the services on demand. The fog layer acts as a super peer, it can provide computation, storage and fetch contents from other fog nodes and backend data center when required. Thus, the user vehicle does not need to worry about the contents, the fog nodes make it available for them. The VFC [7] can serve as the backbone of communication given the scenario of VPN connectivity in an organization. In traditional vehicular networks, the vehicles communicate with edge servers, which are placed at fixed locations and it provides services to a large number of vehicles in the region. Whereas, the VFC, the same request is treated by the neighboring vehicle. Thus, network bandwidth consumption has been reduced significantly using fog vehicles. In VFC, the edge node is also the part of the infrastructure, therefore, it is easier in VFC to communicate with other edge devices to achieve extensive computations [8]. The concept of vehicles as Infrastructure has emerged that served as a foundation for computation and communication.

The concept of fog computing in a vehicular environment addresses the need for computation and storage services. It is still a buzz word that has many challenges to its name. The associated academia and industry started exploring the opportunities that are being provided through VFC [26], [22].

In this paper, we have proposed a vehicular fog computing framework termed as vFog. The vFog is designed to simulate the vehicle-assisted computing environment. As per our knowledge, no such open source framework exist that facilitate researchers to explore the benefits of fog computing inside vehicular networks. The proposed framework is flexibly designed that allow researchers to incorporate their own scheduling policies and modify network characteristic to simulate a realistic environment. The proposed framework supports two types of vehicles, i.e. fog vehicles and end-user vehicles. The fog vehicles provide computation to the user vehicles and also to other fog vehicles; whereas, the user vehicles can only request the resources. In order to handle the churn behavior, the vFog framework supports relay based communication model between fog and user vehicles. The fog vehicles maintain a queue of requests and assigned resources based on the implemented policy. The vFog modular design allows researchers to easily incorporate their own scheduling policies.

The next section covers the recent contribution in the field of vehicular cloud and fog computing. The system model is presented in Section 3. Section 4 covers the proposed system design and its implementation. The performance evaluation is presented in Section 5. Finally, Section 6 and 7 covers the conclusion and future directions.

## II. LITERATURE REVIEW

Vehicular Fog Computing plays a vital role in the field of infrastructure-as-a-service to support latency-sensitive applications. In this section, we cover the recent advances made in the field of vehicular computing.

Li *et al.* [14] proposed a carpooling scheme through block chain-assisted fog vehicles. The main objective is to facilitate passengers to connect with drivers efficiently without undermining their private information.

Chen and Wang [4] proposed a data scheduling mechanism using fog computing in a vehicular environment. With growing number of vehicles, the need for efficient data dissemination increased. Authors present two dynamics schemes for data scheduling using fog computing to improve the vehicular network environment.

Truong *et al.* [23] proposed a scheme that combines Software Defined Networks (SDNs) and VFC in order to address the issues that exist in traditional VANETs. The authors provide a framework for non-critical applications such as data streaming, infotainment, and social media contents. Moreover, the vehicles also share blind-spot information with nearby vehicles.

Luan *et al.* [16] proposed a Vtube framework for content distribution by deploying fog servers at different locations in the city to facilitate vehicular users. The objective is to reduce the download time for vehicular users. Basudan *et al.* [3] proposed a scheme for improving security in crowd-sensing based monitoring system for road surface condition while maintaining privacy. Kim *et al.* [11] solved the parking problem using fog nodes and roadside units. Using these infrastructures, parking spaces are shared so that the drivers can find the space to park their vehicles.

Kang *et al.* [10] proposed a framework known as Fog computing based Internet of Vehicles (F-IoV). Using the edges devices, the privacy-preserving pseudonym ($p^3$) scheme is presented to address the privacy issues of location in IoV.

Chun *et al.* [5] proposed a fog computing architecture that is built using a publish/subscribe model for IoV. A traffic congestion control is also suggested by the authors in the proposed scheme. In VANETs, the identity of a vehicle or a driver can easily be acquired by listening into beacon messages that are broadcast through the vehicles. Such messages can give the basic information of the driver. To overcome this issue, Arif *et al.* [2] presented a solution based on Fog anonymizer using Fog servers. With the increase in road vehicles, traffic efficiency is becoming a major concern. For the optimal traffic flow, traffic control lights need to be efficiently controlled. Liu *et al.* [15] proposed a secure traffic light control system while using fog computing. The authors propose two schemes in which the traffic lights are assumed as a fog device. It prevents the denial-of-service (DOS) attacks and works in low vehicle density as well as in high vehicle density.

In order to secure the communication network between vehicles, Soleymani *et al.* [21] proposed a fuzzy trust model for securing the network in vehicular networks. The suggested trust model executes a progression of security checks to guarantee the rightness of the information obtained from approved vehicles. Also, fog nodes are utilized to assess the extent of precision of the event's location. With the inception of 5G technology, new opportunities arise for information sharing. Ge *et al.* [6] proposed a scheme based on 5G and SDN for the vehicular environment. Moreover, fog cells are introduced to cover the vehicles and their mobility in order to prevent recurring handovers among the vehicle and RSUs. Usually, the cloud-connected vehicular networks are useful for computational tasks; however, to reduce the computational cost and latency of the computations tasks, Zhang *et al.* [28] proposed a framework known as cloud-based mobile edge computing (MEC). The authors demonstrate the effects of the proposed framework on the communication between Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I). Moreover, the framework required well-organized traffic policies in order to monitor traffic violations. As manual labor work for the detection of traffic violations leads to large overhead for the controlling body, which can be efficiently optimized by using automated systems. Roy *et al.* [19] propose fog based Intelligent decision support system for a traffic violation and driver's safety. The authors claim that their proposed framework can be easily implemented and further optimize the decision support systems.

With the increase in the density of vehicles, the computational task at RSUs also increases which causes service degradation. In order to tackle this issue, Ye *et al.* [27] used public buses as fog servers. The buses provide computation resources but also helps the RSUs in offloading the heavy tasks. As vehicular applications are increasing day by day, the demand for an optimized architecture for dealing with big data in the respective environment is much needed. Zhang *et al.* [29] propose cooperative fog computing based intelligent vehicular network (CFC-IoV) for smart cities which caters the mobility, latency, localization, and scalability using the local fog servers (LFSs). A multi-level resource management model is proposed by the authors to streamline the execution of CFC-IoV, which incorporates intra-fog energy-aware and inter-fog QoS-aware resource management. Moreover, in such situations, the software-defined network (SDN) has been proving very effective in handling data in a huge extent of networks. It has been very productive by segregating control plan from the data plane which is the principal theme of SDN. In the developing cloud situation, smart devices assume an essential job. However, mobile devices face latency and connectivity issues. In order to resolve these issues, fog nodes are deployed to improve connectivity and reduce communication delay. Sahoo and Sahoo [20] explore the effectiveness of vehicular networks with SDN and fog nodes. The objective is to improve the execution of traffic management and QoS for the dissemination of real-time data. Table 1 summarize the recently used computing for delay-tolerant applications. The next section covers the preliminaries and system model for the proposed vFog framework.

## III. PRELIMINARIES

The proposed framework is based on vehicle-to-vehicle (V2V) communication. The vFog framework allows vehicles to request computing resources from dedicated fog vehicles ($V_f$). The $V_f$ are dedicated vehicles that can facilitate other vehicles through sharing their computing infrastructure. We also assume that all the vehicles posses limited computed capacity ($C_{cap}$) whereas, the $V_f$ have much better computation infrastructure installed. The tasks can be shared through ad-hoc communication. Every task includes the computation request in the form of millions of instructions per seconds (MIPS). The $V_f$ can receive a request from multiple vehicles, the tasks are first stored in an input queue ($I_q$) and picked up from $I_q$ one-by-one based on priority or any other mechanism. Here, we are using first come first serve model; however, more sophisticated models can be easily incorporated in the vFog framework. The $V_f$ node keep track of requested vehicles to send their results back. If the requested vehicle is not in the direct communication range, the ad-hoc mechanism is adapted to relay the results.

**TABLE 1.** Comparison of existing work on vehicular/edge/fog computing.

| | Fog Nodes | Task Scheduling | Cloud Support | Multi-hop Support | Open Source | Framework Support | Simulator |
|---|---|---|---|---|---|---|---|
| Truong et al. [23] | RSU | ✓ | ✓ | ✗ | ✓ | generic | - |
| Basudan et al. [3] | RSU | ✗ | ✓ | ✗ | ✓ | generic | - |
| Kim et al. [11] | local infrastructures | ✗ | ✗ | ✓ | ✓ | urban | Python based |
| Chen et al. [4] | RSU | ✓ | ✓ | ✗ | ✓ | generic | - |
| Sookhak et al. [22] | vehicles | ✓ | ✓ | ✗ | ✓ | generic | - |
| Kang et al. [10] | RSU | ✓ | ✓ | ✗ | ✓ | generic | - |
| Huang et al. [9] | Fog Infrastructure | ✗ | ✓ | ✗ | ✓ | generic | - |
| Tariq et al. [17] | Edge nodes | ✓ | ✗ | ✓ | ✓ | IoT devices | OMNeT++ |
| Proposed vFog | vehicles | ✓ | ✗ | ✓ | ✓ | urban/highway | SUMO, VEINS, OMNeT++ |

**TABLE 2.** Nomenclature.

| Acronym | Abbreviations |
|---|---|
| V2V | Vehicle-to-Vehicle |
| V2I | Vehicle-to-Insfrastructure |
| RSU | Road-Side-Unit |
| OBU | On-Board-Unit |
| VANET | Vehicular Ad-hoc Network |
| MIPS | Million Instructions per Second |
| $V$ | Vehicles |
| $V_f$ | Fog Vehicle |
| $C_{cap}$ | Computing Capacity |
| $V_u$ | User Vehicle |
| VFC | Vehicular Fog Computing |
| VCC | Vehicular Cloud Computing |
| TTL | Time-to-live |
| $I_q$ | Input Queue |

Moreover, the vehicles are represented with $V$ and task as $t$. Other abbreviations used in this paper are listed in Table 2.

### A. SYSTEM MODEL

In vehicular networks, the effects of channel fading can reduce the overall performance of the entire system. To model the fading effect for wireless signals between vehicle to vehicle (V2V), we have used Nakagami distribution [18].

$$p(x; m, \Omega) = 1 - \frac{\Gamma_{mx^2/\Omega}(m)}{\Gamma(m)} \tag{1}$$

$$= e^{-\frac{m}{\Omega}x^2} \frac{2m^m}{\Gamma(m)\Omega^m} x^{2m-1} \tag{2}$$

where $\frac{\Gamma_{mx^2/\Omega}(m)}{\Gamma(m)}$ represents the cumulative distributive function for signal reception, $x^2$ describes the reception threshold of the propagation signal and $\Omega$ represents the average power level of the received signal. The fading parameter $m$ is defined based on the distance $d$ between the vehicles $i$ and $j$ as:

$$m = \begin{cases} 3.0 & : d < 50m \\ 1.5 & : 50m \leq d < 150m \\ 1.0 & : \text{otherwise} \end{cases} \tag{3}$$

The resident number of vehicles is modeled as the population, whereas the staying time is modeled as the lifetime of the process. We assume incoming process is a Poisson process with time-variant parameter $\lambda_n(t)$ with a period $T$. We also assumed the outgoing rate time-variant $\mu_n(t)$. The distributions of the resident vehicles and the outgoing vehicles followed a Poisson distribution with mean $E_n(t)$ [7].

$$E_n(t) = E_n(t+T) = \int_{-\infty}^{t} \lambda_n(s) \exp\left( - \int_{s}^{t} \mu_n(r)dr \right) ds \tag{4}$$

The vehicles are represented as directly or indirectly connected. The directly connected vehicles can establish the direct communication channel with the fog vehicles. However, the indirectly connected vehicles require relay mechanism for data transfer. We have defined the set of connected vehicles V(t) as given below [12].

$$\begin{aligned} V(t) = \{v_0\} \bigcup \{v_i \mid \exists v_j\} \epsilon V \\ Where\ i \neq j, t_1 < t_0 < t, s, t\ e_{ij} \epsilon E(t_0) \\ and\ v_j \epsilon V(t_1), V(0) = \{v_0\} \end{aligned} \tag{5}$$

The number of connected vehicles for directly or indirectly connecting mode are:

$$\begin{aligned} N_n = |V(t)| \quad and\ N_b(t) = |\{v_i | \exists t_0 \leq t \\ where\ e_{i0} \epsilon E(t_0), v_i \epsilon V(t_0)\} \end{aligned}$$

Given the number of the busy fog vehicles for the tasks with priority $i$, we can derive expected completion time $W_i$ for the task as below:

$$\begin{aligned} W_i &= \sum_{n=l_i+1}^{\infty} (n - l_i) \frac{1^n}{(l_i)!(l_i)^{n-l_i}} (\frac{\lambda_i}{\mu_t})^n p_0 \\ &= \frac{p_0}{l_i!} (\frac{\lambda_i}{\mu_t})_i^l \sum_{n=l_i+1}^{\infty} (n - l_i) \frac{1}{l_i^{n-l_i}} (\frac{\lambda_i}{\mu_t})^{n-l_i} \\ &= \frac{p_0}{l_i!} (l_i \rho)^{l_i} \cdot \frac{\rho}{(1-\rho)^2} \end{aligned} \tag{6}$$

where $p$ is the utilization factor for the model. However, if no priority task $i$ is in the system, the probability is $p_0$ [25]. We represent the vehicle-based fog nodes model as an M/M/1 queuing system. That is, the message flow entering the moving vehicle-based fog system follows a Poisson process with an arrival rate of $\lambda_i^m$. The moving vehicle flow can be viewed as a fixed server with process rate $\lambda_i^{vehicle}$ and the service rate is computed as $p_i^{mf} = \lambda_i^m / \lambda_i^{vehicle}$. We have observed that the average response time from fog vehicle is directly related to message arrival rate $\lambda_i^m$ and moving-vehicle $\lambda_i^{vehicle}$ which can be computed as in [24]:

$$\begin{aligned} E(t^m_{stol}) &= E(t^m_{que}) + E(t^m_{ser}) + 2\tilde{A} - t^m_{up} \\ &= \frac{p_i^{mf}}{\lambda_i^{vehicle}(1 - p_i^{mf})} + \frac{1}{\lambda_i^{vehicle}} \\ &\quad + 2 \times d_{r_i} \rightarrow mfog \\ &= \frac{1}{\lambda_i^{vehicle} - \lambda_i^m} + 2 \times d_{r_i} \rightarrow mfog \qquad (7) \end{aligned}$$

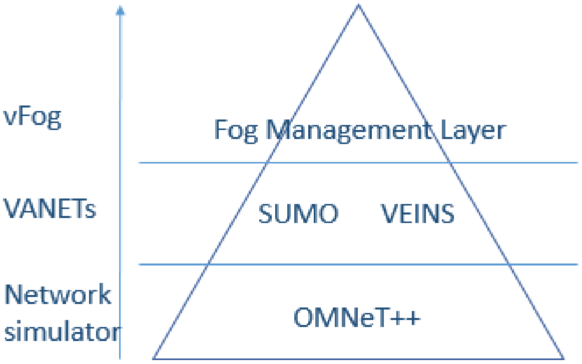

**FIGURE 2.** vFog architecture.

## IV. PROPOSED FRAMEWORK

The proposed vFog framework is designed on the top of OMNeT++, using SUMO and Veins as shown in Fig. 2. The fog management layer is added, which handles task scheduling policy, relay mechanism, user and fog vehicles as shown in Fig. 3. The proposed framework allows vehicles to share computation tasks with other vehicles. Here we have assumed that the requested vehicles do not have enough resources to execute the task locally. All the communication among vehicles is based on the IEEE 802.11p/WAVE protocol. The vehicles in the proposed framework are categorized as user vehicles $V_u$, which has computation jobs. The second category of vehicles are fog $V_f$ vehicles that have onboard computation available to serve $V_u$ vehicles.

**Fog and User Vehicles –** The fog vehicles ($V_f$) periodically advertise its computing resources through control messages ($M_{adv}$). The $M_{adv}$ include the specification of available computing capacity ($C_{cap}$) and possible route of $V_f$. The interested vehicles receive such advertisements from multiple $V_f$ that are in the direct transmission range. This allows the $V_u$ vehicle to select the suitable fog vehicle for task computation. Due to churn behavior, the user vehicles can deviate from nearby $V_f$ vehicle. Therefore, the faster execution of the task and good communication channel is a critical parameter used to take the decision. Moreover, the framework also supports the multi-hop communication model, which can be utilized to transfer the result to the $V_u$. Before, transfer of task, the first step is the handshake process which is important as it allows $V_u$ to know the route of fog node; thus, improves the job success rate. In response to the $M_{adv}$ the $V_u$ vehicle sends the response message $M_{res}$ to show its interest in using the available computing power at the $V_f$ vehicle. In response to $M_{res}$ the $V_f$ vehicle sends its queue length, expected waiting time and its route. The waiting time is computed based on the pending tasks computation requirements (Million Instruction Per Second - MIPS) and available computing power. Therefore, non-interested vehicles do not respond if the waiting time is not suitable for their application. The further interested $V_u$ vehicles send a computing task ($T_i$) to selected $V_f i$ vehicle. The received $T_i$ is inserted in the $Q$ from where it is served based on the scheduling policy. On $T_i$ completion, the $V_f$ vehicle sends the results $M_{res}$ to the $V_u$ vehicle.

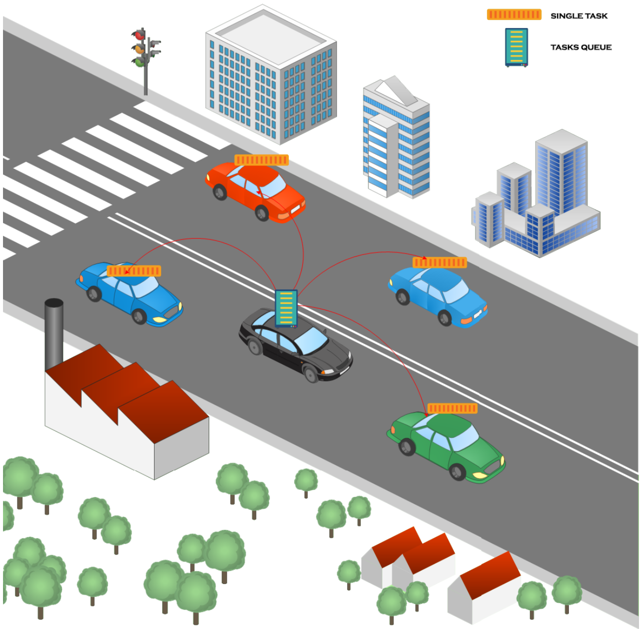

**FIGURE 3.** vFog computing framework – showing the task sharing with fog vehicle.

Further, the $V_u$ vehicle has the leverage to assign the same $T_i$ to multiple $V_f$ vehicles. In the existing vFog framework this mechanism is allowed as it can provide the fault tolerance mechanism in vehicular networks. Moreover, in the peer-to-peer model, it is helpful to validate the task result using a mechanism such as Byzantine failure [13]. Algo. 1 and Algo. 2 shows the execution of $V_f$ and $V_u$ vehicles.

**Relay Mechanism –** Due to the high mobility of vehicles in Vehicular Adhoc Networks (VANETs), the connection time between the $V_f$ and $V_u$ vehicles can be quite low. The lesser connection time between $V_f$ and $V_u$ vehicles can cause packet drops which can affect the performance of the overall

**Algorithm 1** Fog Vehicle

```
 1: $recReq_{flag} \leftarrow 0$
 2: while $recReq_{flag} == 0$ do
 3:     if $recReq == true$ then
 4:         $recReq_{flag} \leftarrow 1$
 5:         if $MIPS\_available == true$ then
 6:             send $C_m$ and queue size
 7:         else
 8:             go to line 2
 9:         end if
10:         if $recACK == true$ then
11:             compute Job
12:         else
13:             go to line 2
14:         end if
15:     end if
16: end while
17: if $in\_range == true$ then
18:     send $R_m$
19: else
20:     add TTL
21:     go to Algo 3
22: end if
```

**Algorithm 2** User Vehicle

```
 1: received advertisement
 2: check MIPS
 3: if $desiredMIPS == true$ then
 4:     send $Req_m$
 5: else
 6:     go to end
 7: end if
 8: if $C_m == true$ then
 9:     add entry in table
10:     send $A_m$ to optimal $V_f$
11: else
12:     go to line 4
13: end if
14: change job status upon job completion
```

**Algorithm 3** Relay Mechanism

```
 1: received completed job
 2: if $is\_V_f == true$ then
 3:     send $Res_m$
 4: else
 5:     check neighbor list
 6: end if
 7: if $is\_neighbor == true$ then
 8:     unicast $Res_m$
 9: else
10:     decrement TTL
11:     broadcast $Res_m$
12: end if
```

**TABLE 3.** Simulation parameters.

| Parameters | Values |
|---|---|
| Simulation area | 3km × 3km |
| Simulation time | 3000s |
| Simulation runs | 40 |
| Transmission range | 250m |
| Data transmission rate | 6 Mbps |
| MAC protocol | IEEE 802.11p WAVE |
| Distribution Model | Nakagami |
| Vehicle Density | 20–120km |
| Fog Vehicle | Every fifth vehicle |
| Vehicle Velocity | 14–54 $m/s$ |
| Maximum acceleration | 2.6 $m/s^2$ |
| Road Side Units | No |
| Road Type | Two-way supported |
| Computation request rate | random |
| Computation Task size | random( 500 - 900) MIPS |

network by reducing the packet delivery ratio. Moreover, the $T_i$ is counted as unsuccessful if the requested $V_u$ vehicle cannot receive the response of its computation task. Thus, this can decrease the packet delivery ratio. In order to tackle the packets drops due to the churn behavior of vehicles, we have used the relay mechanism. The relay mechanism ensures the result must reach the $V_u$ vehicle if it is within the range of $\delta$h hops. On losing the direct connection with $V_u$ vehicle, the $V_f$ vehicle broadcast the result $M_{res}$. This result $M_{res}$ message also includes the Time-To-Live (TTL) field which allows this packet to traverse a few vehicles to find the $V_u$ vehicle. If $V_u i$ receives the $M_{res}$ destined for $V_j$, the $V_u i$ first check the vehicles which are in its direct transmission range to send $M_{res}$ directly to that vehicle otherwise it broadcast after reducing the TTL value. The execution of the relay mechanism is listed in Algo. 3.

## V. PERFORMANCE EVALUATIONS

In this section, we have benchmarked the proposed framework. The simulation parameters are shown in Table 3. An area of 3km × 3km is used with variable $V_u$ and $V_f$ vehicles. The entire simulation is performed multiple time and results are reported in the next section. Moreover, the computation tasks are randomly selected in the range of 500 to 900 MIPS. In the benchmark study, the $V_f$ vehicles are moving to random directions and provide computation to the nearby vehicles. The $V_u$ vehicles find the $V_f$ through the beacons. After identifying the $V_f$ vehicles, the user vehicles generate a computation request at random intervals. The $V_f$ receives the computation request from other vehicles, inform them about the expected queuing delay and return results after computation. In vehicular networks, the most challenging part is the mobility of the vehicle, therefore, to handle this problem, vehicles can act as a relay node to forward the computing request of $V_u$ to $V_f$ and similarly, the results are also reported back through multi-hop relay mechanism.

### A. RESULTS AND DISCUSSIONS

The vFog is evaluated to benchmark the resource usage and other network related parameters such as transmission delay, packet delivery ratio, multi-hop dissemination, and system efficiency.

### 1) TRANSMISSION DELAY

Fig. 4 shows the relationship between vehicle density and average transmission delay. The average transmission delay is measured during the communication among $V_u$ and $V_f$ vehicles. It has been observed that the transmission delay increases with the increase in vehicle density. This is due to the fact the with more vehicles, more communication messages are generated which eventually congest the network and results in a high packet drop rate.

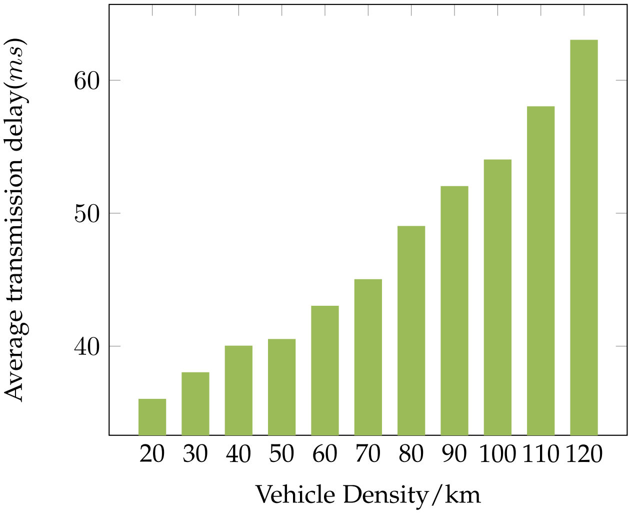

**FIGURE 4.** **Average transmission delay between $V_u$ and $V_f$ vehicle communication.**

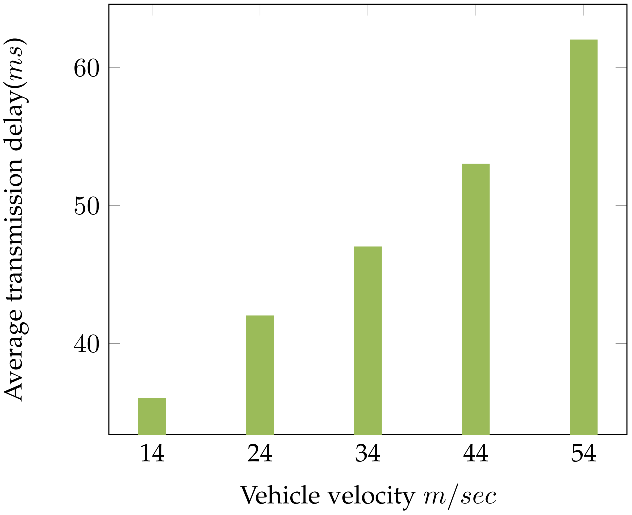

**FIGURE 5.** **Vehicle velocity VS average transmission delay (*ms*).**

Fig. 5 shows a comparison between the average transmission delay with varying vehicle speed. Low latency has been observed at low speed, this is due to the fact that at low velocity, vehicles have stable and longer network connection time, therefore the packets are delivered swiftly. Whereas, at high speed, vehicle mobility increases which affect the network connection time. Moreover, the $V_u$ vehicle which requests a computation from $V_f$ quickly moves out of the communication range of $V_f$ vehicle. In that case, the completed tasks are delivered through the multi-hops relaying mechanism. However, such tasks can be dropped after a fixed number of hops. Another reason for an increase in transmission delay is the overhead at the $V_f$ vehicle due to the computation requests of other nearby vehicles.

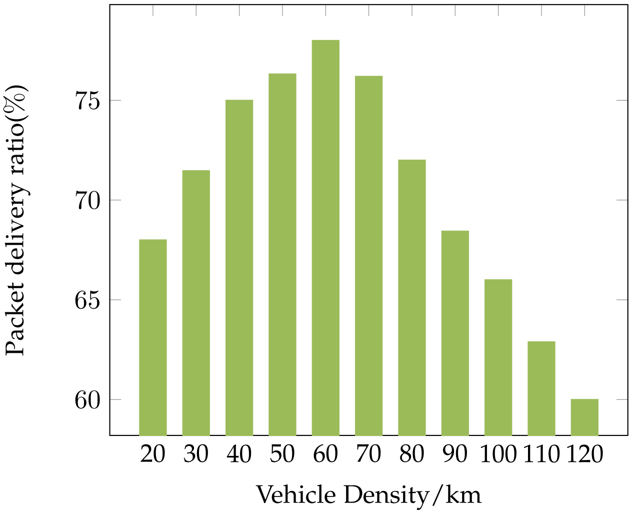

**FIGURE 6.** **Average packet delivery ratio between $V_u$ and $V_f$ with varying vehicle density.**

### 2) PACKET DELIVERY RATIO

The packet delivery ratio is shown in Fig. 6. The packet delivery ratio increases with an increase in vehicle density. However, after reaching mid value i.e. 60 vehicles/km, the packet delivery ratio becomes saturated. The reason for the packet drop is due to the increasing vehicle density where every vehicle is generating a computation request for its nearby $V_f$ vehicles. Moreover, on the other hand, it is difficult for the $V_f$ vehicle to successfully receive the multiple requests, thus, cause the bottleneck at nearby $V_f$ vehicle.

### 3) MULTI-HOP RESULT DISSEMINATION

The multi-hop benchmark parameter is used to analyze the traffic that is generated due to the result of the relay mechanism. Moreover, due to mobility, the $V_u$ and $V_f$ vehicles can change its route which can lead to task failure, thus, multi-hop delivery can improve the delivery ratio. Fig. 7 shows the number of hops taken to deliver the results back to the $V_u$ vehicles. In the proposed framework, the $V_u$ vehicle directly send their tasks to nearby $V_f$ vehicles; whereas, after execution, results can be received through a relay mechanism. In the simulated scenario, it has been observed that after initiating the computation task at nearby $V_f$ vehicle, around 40.5% $V_u$ remains within the direct communication range with $V_f$; whereas, other 54.4% deviate their path but still able to get the results if they are connected with other vehicles.

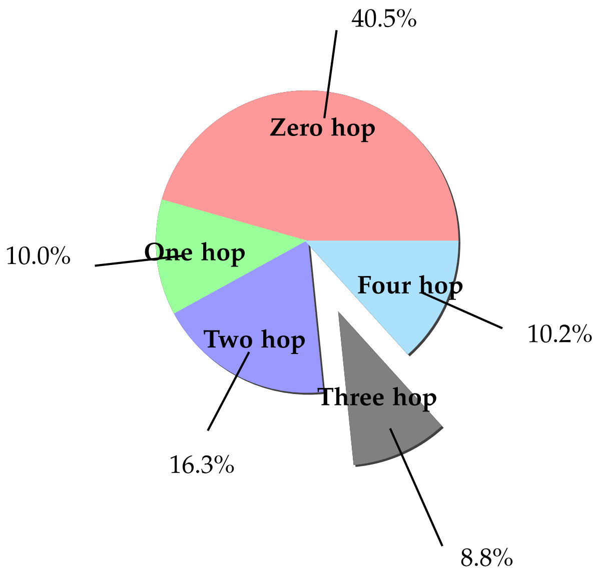

**FIGURE 7.** Multi-hop task delivery.

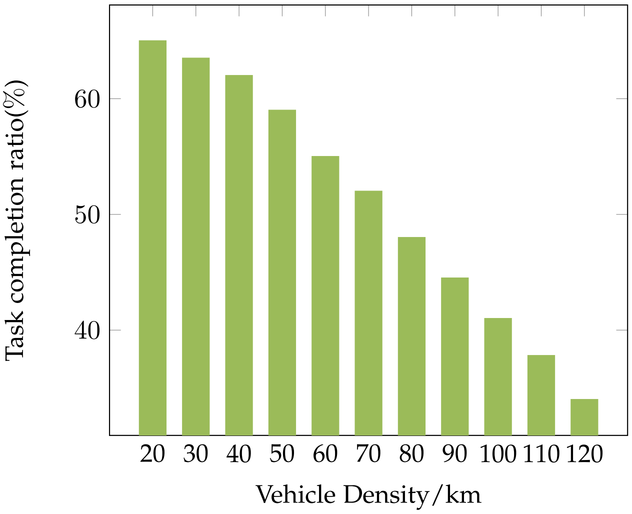

**FIGURE 8.** Efficiency - ratio of task generated to the completed tasks.

However, 14.2% $V_u$ vehicles are disconnected from the $V_f$ connected network.

#### 4) EFFICIENCY

*Efficiency is the ratio of computation task generated to the task successfully completed.* The system efficiency highly depends upon the number of available $V_f$ vehicles. The more the $V_f$ vehicles, more task can be successfully received and executed. With limited $V_f$ vehicles, the packet delivery ratio is also decreased with an increase in vehicle density and it also affects the efficiency. In the present case study, every fifth node is $V_f$ vehicle. Therefore, with an increased number of $V_u$ vehicles, and every $V_u$ vehicle is generating random size computation requests, the $V_f$ vehicles are unable to handle the request; this can lead to an increase in queue size and eventually, $V_f$ start dropping the received packets.

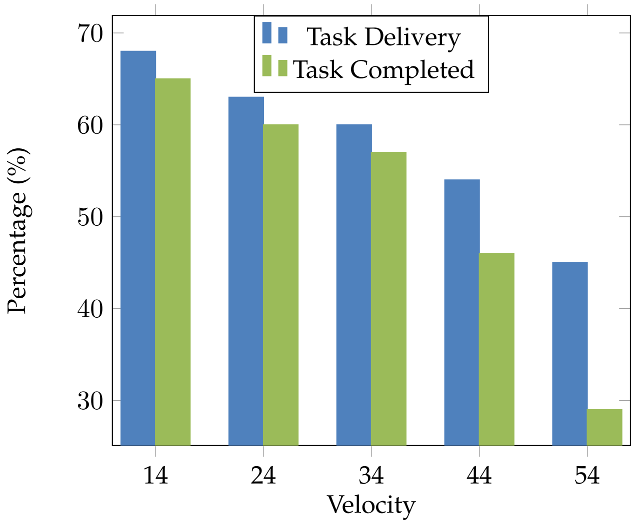

**FIGURE 9.** Task delivery and task completed with varying vehicles speed.

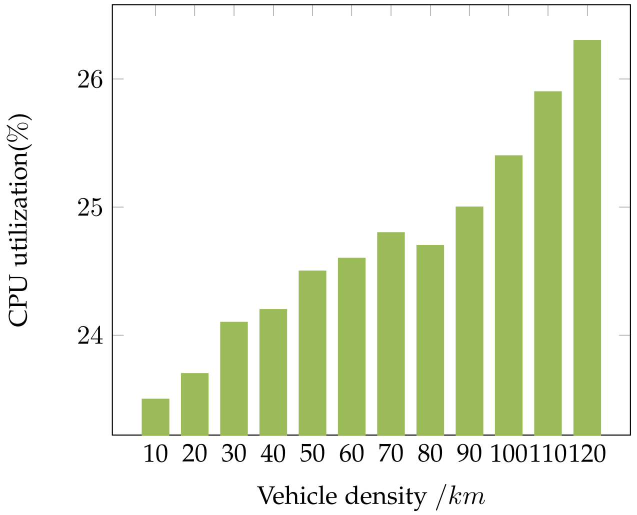

**FIGURE 10.** vFog CPU utilization on commodity hardware with varying vehicles density per km.

Fig. 9 shows the impact of task delivery and task completed with varying vehicle speed. From the figure, it can be observed that in all the simulations, the total task successfully received at $V_f$ vehicles is greater than task completion. It is also observed that with increasing speed, the total task successfully received decreases. This is due to the limited communication time between $V_u$ and $V_f$ vehicles. However, the decrease in completion is also observed, this is due to the high speed where $V_f$ covers a much larger distance in less time, thus, a large number of $V_u$ vehicles try to establish a connection with $V_f$ vehicle which disrupts the communication and execution.

#### 5) SYSTEM RESOURCES USAGE

This section covers the vFog resource utilization in terms of CPU and memory usage by increasing the number of vehicles. Fig. 10 shows the CPU utilization alongside vehicle

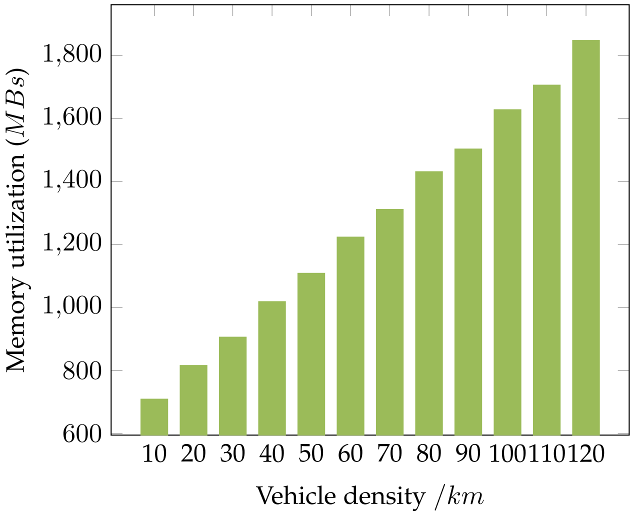

**FIGURE 11.** vFog memory utilization on commodity hardware with varying vehicles density per km.

density. It has been observed that vFog CPU consumption is lower, this is due to the fact that all the modules are based on the event-driven model. The communication between different modules is achieved through event-based message passing. Thus, no infinite loops are used to make this framework more scalable. The maximum CPU utilized is around 26 % when the node density is kept at 120 vehicles/km. Here, in Fig. 10, vehicle density represents both the $V_f$ and $V_u$ vehicles. However, the limitations observed is due to the system constraints, if we kept on increasing the vehicle density, at a certain point the user can feel performance lag in Graphical User Interface (GUI) and module initializations.

- Fig. 11 shows memory usage with varying vehicle density. It is observed the memory utilization is low with low vehicle density. However, only 1800 MB of memory is used when the vehicle density is kept at 120 vehicles/km. This shows that vFog can easily run on commodity hardware with limited resources. The vFog is designed to avoid memory leaks, which is thoroughly tested using Valgrind.[1]

## VI. CONCLUSION

In this paper, we proposed a vehicular fog computing framework which is an emerging area in the context of smart cities. The proposed framework addresses the issues that exist in the traditional vehicular cloud-based techniques, such as latency, and centralized services are being used. The vFog framework can be used to analyze the behavior of fog computing for smart cities. Our proposed framework is lightweight can easily run on commodity hardware. Most of the requests are computed and delivered directly to the original initiator of the request while others are carried out through the multi-hops mechanism. As per our literature review, vFog is the first approach towards the design of vehicular fog computing framework.

## VII. FUTURE WORK

The open source vFog framework can facilitate researchers to extend this work further. The vFog framework allows users to incorporate their own task scheduling algorithms. Moreover, traffic data sets can also be used to benchmark the performance of particular scenarios. Further, the use of RSUs can also be integrated to support fog vehicles. The RSUs can be used to gather real-time fog vehicle data to predict future trends, therefore, the number of fog vehicles can be managed. Thus, the framework provides an opportunity for researchers to focus on their work, without worrying about the underlying system.

Moreover, vFog is built on the top of OMNeT++, which also support the simulation of mobile devices. Therefore, the proposed framework can be integrated with IoT devices to simulate more complex real-world scenarios.

## REFERENCES

[1] B. Ahmed, A. W. Malik, T. Hafeez, and N. Ahmed, "Services and simulation frameworks for vehicular cloud computing: A contemporary survey," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, p. 4, Jan. 2019.
[2] M. Arif, G. Wang, and V. E. Balas, "Secure VANETs: Trusted communication scheme between vehicles and infrastructure based on fog computing," *Stud. Inform. Control*, vol. 27, no. 2, pp. 235–246, 2018.
[3] S. Basudan, X. Lin, and K. Sankaranarayanan, "A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 772–782, Jun. 2017.
[4] X. Chen and L. Wang, "Exploring fog computing-based adaptive vehicular data scheduling policies through a compositional formal method–PEPA," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 745–748, Apr. 2017.
[5] S. Chun, S. Shin, S. Seo, S. Eom, J. Jung, and K.-H. Lee, "A pub/sub-based fog computing architecture for Internet-of-vehicles," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Dec. 2016, pp. 90–93.
[6] X. Ge, Z. Li, and S. Li, "5G software defined vehicular networks," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 87–93, Jul. 2017.
[7] 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.
[8] C. Huang, R. Lu, and K.-K. R. Choo, "Vehicular fog computing: Architecture, use case, and security and forensic challenges," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 105–111, Nov. 2017.
[9] C.-Y. Huang and K. Xu, "Reliable realtime streaming in vehicular cloud-fog computing networks," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Jul. 2016, pp. 1–6.
[10] J. Kang, R. Yu, X. Huang, and Y. Zhang, "Privacy-preserved pseudonym scheme for fog computing supported Internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2627–2637, Aug. 2018.
[11] O. T. T. Kim, N. D. Tri, V. D. Nguyen, N. H. Tran, and C. S. Hong, "A shared parking model in vehicular network using fog and cloud environment," in *Proc. IEEE 17th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Aug. 2015, pp. 321–326.
[12] X. Kui, Y. Sun, S. Zhang, and Y. Li, "Characterizing the capability of vehicular fog computing in large-scale urban environment," *Mobile Netw. Appl.*, vol. 23, no. 4, pp. 1050–1067, 2018.
[13] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982.
[14] M. Li, L. Zhu, and X. Lin, "Efficient and privacy-preserving carpooling 564 using blockchain-assisted vehicular fog computing," *IEEE Internet Things J.*, to be published.
[15] J. Liu *et al.*, "Secure intelligent traffic light control using fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 817–824, Jan. 2018.
[16] T. H. Luan, L. X. Cai, J. Chen, X. Shen, and F. Bai, "VTube: Towards the media rich city life with autonomous vehicular content distribution," in *Proc. 8th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw. (SECON)*, Jun. 2011, pp. 359–367.

---

[1] http://www.valgrind.org/

[17] T. Qayyum, A. W. Malik, M. A. K. Khattak, O. Khalid, and S. U. Khan, "FogNetSim++: A toolkit for modeling and simulation of distributed fog environment," *IEEE Access*, vol. 6, pp. 63570–63583, 2018.

[18] M. H. Rahman and M. Nasiruddin, "Impact of two realistic mobility models for vehicular safety applications," in *Proc. IEEE Int. Conf. Inform., Electron. Vis. (ICIEV)*, May 2014, pp. 1–6.

[19] S. Roy, R. Bose, and D. Sarddar, "A fog-based DSS model for driving rule violation monitoring framework on the Internet of Things," *Int. J. Adv. Sci. Technol.*, vol. 82, pp. 23–32, Jan. 2015.

[20] K. S. Sahoo and B. Sahoo, "SDN architecture on fog devices for realtime traffic management: A case study," in *Proc. Int. Conf. Signal, Netw., Comput., Syst.*, Springer, 2017, pp. 323–329.

[21] S. A. Soleymani *et al.*, "A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing," *IEEE Access*, vol. 5, pp. 15619–15629, 2017.

[22] M. Sookhak *et al.*, "Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing," *IEEE Veh. Technol. Mag.*, vol. 12, no. 3, pp. 55–64, Sep. 2017.

[23] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular adhoc network with fog computing," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 1202–1207.

[24] X. Wang, Z. Ning, and L. Wang, "Offloading in Internet of vehicles: A fog-enabled real-time traffic management system," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4568–4578, Oct. 2018.

[25] Z. Wang, Z. Zhong, and M. Ni, "Application-aware offloading policy using smdp in vehicular fog computing systems," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2018, pp. 1–6.

[26] Y. Xiao and C. Zhu, "Vehicular fog computing: Vision and challenges," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2017, pp. 6–9.

[27] D. Ye, M. Wu, S. Tang, and R. Yu, "Scalable fog computing with service offloading in bus networks," in *Proc. IEEE 3rd Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)*, Jun. 2016, pp. 247–251.

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

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

**SYED SARMAD SHAH** received the B.Sc. degree in information technology from IBMS, The University of Agriculture, Peshawar, Pakistan. He is currently pursuing the M.Sc. degree in information technology from the National University of Sciences and Technology (NUST), Islamabad, Pakistan. His research interests include VANETs, vehicular fog computing, and distributed systems.

**MUHAMMAD ALI** received the B.Sc. degree in computer science from COMSATS University, Lahore, Pakistan. He is currently pursuing the M.Sc. degree in information technology from the National University of Sciences and Technology (NUST), Islamabad, Pakistan. His research interests include VANETs, the Internet of Vehicle, fog computing, and distributed systems.

**ASAD WAQAR MALIK** received the Ph.D. degree in computer software engineering from the National University of Sciences and Technology (NUST), Pakistan, where he started his academic career as an Assistant Professor, in 2012. In 2019, he joined the University of Malaya as a Senior Lecturer. His research interests include parallel and distributed simulation, cloud/fog computing, and large-scale networks.

**MUAZZAM A. KHAN** received the master's degree in mobile ad hoc networks from IIUI and the Ph.D. degree in computer sciences as a sandwich program from IIUI and the University of Missouri–Kansas City (UMKC), USA, in 2011. He completed his first Postdoctoral Research with the University of Ulm, Germany, in 2013, and the second Postdoctoral Research with UMKC, in 2016, where he was with the Networking and Multimedia Lab as a Research Fellow. In 2011, he joined the CS Department, Abdul Wali Khan University Mardan, as an Assistant Professor/Chair. Later, he joined the National University of Sciences and Technology, Islamabad, Pakistan, as an Assistant Professor, in 2013, where he is currently a Tenured Associate Professor/Associate Dean with the Department of Computing, SEECS. His research interests include wireless sensors network, body area networks, image processing, image compression, image encryption, and data network security.

**SRI DEVI RAVANA** received the bachelor's degree in information technology from the National University of Malaysia, in 2000, the master's degree in software engineering from the University of Malaya, Malaysia, in 2001, and the Ph.D. degree from the Department of Computer Science and Software Engineering, The University of Melbourne, Australia, in 2012. She is currently an Associate Professor and the Head of the Department of Information Systems, University of Malaya. Her research interests include information retrieval heuristics, text indexing, data analytics, and data mining. She received a couple of best paper awards in international conferences within the area of information retrieval.