Load-balancing of computing resources in vehicular fog computing

Ahmad Raza Hameed and Kashif Munir
National University of Computer and Emerging Sciences,
Islamabad 44000, Pakistan

Saif ul Islam
Institute of Space Technology,
Islamabad 44000, Pakistan

Ishfaq Ahmad
University of Texas at Arlington,
USA

2020 3rd International Conference on Data Intelligence and Security (ICDIS) | 978-1-7281-9379-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICDIS50059.2020.00020

***Abstract*—The fog paradigm has the potential to efficiently address the time constraints of real-time critical applications by providing cloud-like services closer to the Internet of Things (IoT) devices. In this context, an emerging application is a cluster of auto-driven vehicles that are networked and can process the requests of IoT devices attached to the vehicles. With the ability to collectively provide abundant computing power as well as communication and storage resources, the cluster of vehicles can be viewed as a pool of fog devices. In order to utilize such resources, we devise a mechanism that provides load-balancing among the vehicular devices despite their mobility. The proposed mechanism functions in two ways: clustering among mobile fog devices and capacity-based load-balancing among clusters. The clustering algorithm efficiently builds clusters, allowing the selection of a cluster head based on the fog mobility factor. In addition to reducing congestion in the network, the proposed capacity-based load-balancing algorithm efficiently balances the load within a cluster as well as among the clusters of vehicles. Simulations conducted on NS2 demonstrate the efficiency of the proposed scheme in terms of end-to-end delay and packet delivery ratio (PDR).**

***Index Terms*—Vehicular fog computing, Cloud computing, Vehicular ad hoc networks, Internet of Things, Capacity-based load-balancing, Clustering.**

I. Introduction

The internet of things (IoT) is emerging as a revolutionary paradigm to enact the connectivity in various application domains, especially in smart healthcare, smart power-grid, smart agriculture [1] surveillance, intelligent transportation systems (ITS) and smart buildings. As a result, a massive amount of data is generated by heterogeneous IoT devices, which requires sophisticated processing for further utilization in the smart applications to allow user intelligent decision making. However, the resource-constrained nature of IoT devices restricts them to process data by using their own resources [2, 3]. Hence, cloud services are often used by IoT devices to overcome their limitations by offloading processing and storage. The data generated from IoT devices is transported to a cloud platform for performing compute-intensive operations and storage. When the number of requests sent to the cloud increases, the congestion on the network also exacerbates, which is due to the higher core network bandwidth usage and the centralization of cloud datacenters. These effects can become more severe when the billions of IoT devices will be deployed worldwide. Therefore, the utilization of external resources is not a judicious solution for real-time critical applications. To address these underlying issues faced in the IoT-cloud paradigm, fog computing is utilized that creates a middle layer between IoT devices and cloud datacenters [4]. Due to the ever-increasing demand for resources, it is crucial to identify and optimize the potential capabilities of processing and storage used in other application domains in the vicinity of the end devices [5].

A vehicular cloud (VC) paradigm is introduced to connect the vehicles to form a cloudlet [6]. These cloudlets commonly exist at parking-lots that provide cloud-like services and are further integrated with traditional cloud network to collaboratively

978-1-7281-9379-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDIS50059.2020.00020

complete the tasks [7]. However, cloudlets are limited in geographical distribution and a certain amount of vehicles are required to create a cloudlet. Therefore, due to a limited number of cloudlets, congestion is created in the network, and resources of vehicles are not fully utilized. So, vehicles as fog devices are considered [8, 9] to provide cloud-like services.

As discussed earlier, a vehicular fog network being mobile in nature poses more challenges to be utilized as a distributed system. To avoid the bottleneck issue and to utilize the available bandwidth efficiently, load-balancing techniques are considered an efficient solution in mitigating the aforementioned problems [10]. However, implementing scheduling and load-balancing techniques in a dynamic and mobile network like VANET is extremely challenging. While a majority of the existing load-balancing schemes focus on static fog environments [11, 12, 13]. The proposed load balancing mechanism works in two phases: First, it performs clustering while considering the mobility of fog devices. Second, a capacity-based load-distribution mechanism efficiently balances the load among the vehicular nodes. The contributions of this paper are summarized as follows:

- We propose a vehicular fog distributed computing architecture to process the IoT requests. This is done by dividing the network into a bottom IoT layer, a middle vehicular fog layer, and a cloud layer.
- We design a dynamic clustering approach that considers the position, speed, and direction of vehicles. Moreover, we derive a mechanism to identify the time of leaving (of a vehicle) from the cluster. Additionally, the technique predicts the future position of the vehicles considering the dynamic nature of vehicular networks.
- Another contribution is a capacity-based load-distribution mechanism to perform the local load-balancing within the cluster and inter-cluster global load-balancing. This technique allows us to avoid the congestion and improve the performance of the network.
- We carry out simulations using the state-of-the-art NS2 network simulation environment. The results demonstrate that the proposed scheme achieves reduced network delay and improved PDR.

The rest of the paper is organized as follows: Section II presents the study of several relevant research work. Section III illustrates the proposed methodology. Section IV provides evaluation and experimentation details. Section V includes conclusions and future research extensions.

II. Related work

A variety of techniques for efficient resource management in fog computing have been proposed. In this section, we discuss achievements, parameters, and features of existing time-sensitive load-balancing approaches. In doing so, we classify the related work into static fog networks and mobile fog networks.

A. Static fog networks

In order to reduce the response time of the network, the authors propose a two-level resource scheduling model in [12]. The proposed model presents a novel fog computing architecture that divides the network into the edge, middle, and core layers. The resource scheduling model schedules the jobs among various fog clusters and then performs task scheduling within the same fog cluster. Moreover, a multi-objective optimization task scheduling scheme is proposed to efficiently reduce task latency. In order to improve the stability of task execution, improved non-dominated sorting genetic algorithm $(NSGA-II)$ is proposed. This scheme achieves more stability and reduced end-to-end delay.

In [13], authors propose an online geographical based load-balancing mechanism for energy harvesting mobile edge computing (MEC). The proposed algorithm namely geographical load-balancing $(GLOBE)$ minimizes the cost of the network by balancing the computation delay constraint and data traffic load in edge devices. As a result, the proposed algorithm operates online. This algorithm finds a close optimal cost as compared to the ordinal cost carried by the offline algorithm that requires future information.

In [14], the authors propose a latency-aware application module management policy to reduce the response time in fog network. This policy works in two phases. Firstly, in order to minimize the latency, this approach efficiently decomposes the application modules based on their execution time. The execution time of the application module on a fog device depends on the run time and requirement of associated device resources such as CPU, RAM, and bandwidth for a particular task. Secondly, this policy efficiently explores and differentiates the application modules that are time-sensitive and other application modules that do not have time constraints. This approach schedules the time-critical applications to the nearby located fog nodes. Furthermore, the latency-aware policy is evaluated in ifogSim [15] and archives less context switching, less time to forward module to the target node, and use the minimum time to find the target node.

Theoretical modeling of response time in fog computing is formulated in [16]. In this work, the authors develop a theoretical model considering the individual components of fog nodes such as – CPU, RAM, bandwidth, transmission power, received signal power, and energy dissipation. The proposed mathematical formulation is evaluated in terms of latency. In [17], the authors propose a reinforcement learning technique to manage the fog resources for improving average time cost, computing power demand, task distribution, and offloading.

B. Mobile fog networks

Vehicles as a viewpoint of infrastructure (called vehicular fog computing (VFC)) is proposed in [8]. In this work, authors employ vehicles as infrastructures for communication and computation to enhance the availability and capability of traditional resources. Moreover, four different scenarios of parked and moving vehicles are explained in terms of communication and computation capacity. Authors analyze the capability and feasibility of vehicles as infrastructure by using real traces of vehicles' data. Ye *et al.* propose a scalable fog network to utilize transport buses as fog devices for reducing the overloaded burden of roadside cloudlets [9]. When RSUs are fully loaded, it offloads the tasks to bus fog devices. Furthermore, a genetic algorithm-based strategy is used to offload tasks from RSUs to bus fog resources. The proposed scheme reduces the response time of the network and improves network utilization.

In [18], authors collaboratively used vehicles as cloudlets along with fixed cloudlets to accommodate the requirements of mobile users. Moreover, a flexible offloading strategy (FOS) is proposed to discover and utilize the underutilized resources of vehicles. This scheme selects a suitable resource (vehicle) based on its current utilization in the vehicular-cloud network. This approach achieves reduced response time of the network by efficiently utilizing the underutilized vehicular resources.

The majority of the existing techniques consider the load-balancing and data offloading in static fog networks [12, 19] where fog nodes are fixed. Whereas, the proposed technique takes into account the load-balancing and data offloading in mobile fog network where fog nodes are moving vehicles. The existing work conducted in vehicular fog computing takes the vehicles as cloudlets or RSUs [18, 20]; however, the proposed technique considers individual vehicle as a separate fog entity. Moreover, the dynamic clustering approach, intra- and inter-cluster resource management also exhibits the novelty of the proposed approach. The proposed technique takes the multi-objective offloading decision by considering the mobility, direction, and capacity of vehicular fog devices.

III. Proposed approach

In the proposed technique, the cloud is considered as a large-scale processing platform that is located away from IoT devices that are geographically distributed and generate data signals through sensing the real environment. Due to resource constraints of IoT devices, the generated data signal is further forwarded to the cloud for processing or storage. In the proposed approach, fog acts as a middle layer between cloud and IoT devices. The fog layer contains fog devices to serve the requests of real-time-critical applications at the edge of the network. However, fog devices have limited communication, computation, and storage capabilities in comparison to cloud datacenters.

A. Network architecture

The network architecture is divided into three layers – bottom layer IoT devices, intermediate vehicular fog layer, and cloud layer. In network, IoT devices periodically sense the data signals and forward them to the fog gateway node. Fog gateway contains a classification mechanism, which efficiently classifies the task as time-critical and delay-tolerable tasks. Before assigning the time-critical tasks to the fog layer, the availability and capacity of fog devices are checked. After verification, the fog layer accomplishes the required task and returns the response to IoT devices. If the data is required to be stored, it is forwarded to the cloud. The mobility of vehicles, capacity constraints are considered while offloading the tasks to the fog layer. The prediction of mobility and balanced offloading are described in upcoming sections.

B. Routing Mechanism

It is assumed that IoT devices are connected to the upper layer and send their tasks to vehicular fog devices.

1) Route setup: Modern vehicles are equipped with a wide range of varied types of sensors. The sensors systems in modern vehicles are used to sense the movement and track the distance traveled by vehicles. The radio-range of each vehicle covers the entire width of the road. To establish a connection among vehicles, beaconing messages are transmitted by each vehicle in the network. This message contains the identification of vehicles, location coordinates, the condition of the road, and time stamp. Based on the beaconing message, vehicles acquire the estimated distance among each other and connect with neighbor nodes in their communication range. These chunks of connected vehicles are said to be clusters. Furthermore, we use a random way-point model (RWP) in simulation to evaluate the vehicular mobile network [21]. The RWP model generates the mobility of devices to depict real mobile scenarios such as location, speed, and acceleration of moving devices.

2) Cooperative offloading: After the route setup phase, IoT sends the requests to the fog layer for the required services. Given this, the fog gateway receives the requests from IoT devices that are in its communication range. Further, fog gateway sends the tasks to the cluster head (CH) fog device that is responsible to distribute the tasks among the clusters and within the cluster of vehicles.

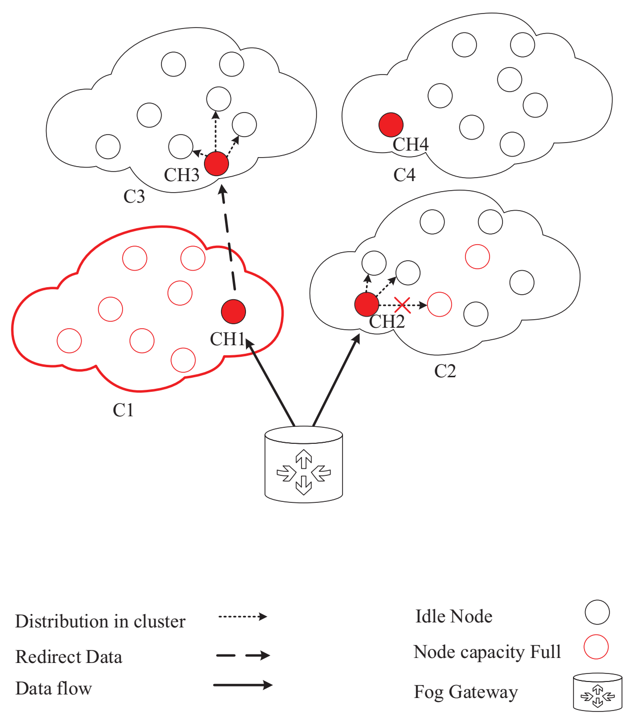

Fig. 1: The cooperation mechanism among vehicular fog nodes.

a) Load-distribution: To select the appropriate node as a CH, fog gateway uses beaconing and prediction mechanisms. By doing this, fog gateway acquires information of surrounding mobile nodes. The gateway node selects CH considering proximity (near to the fog gateway) and velocity of mobile node less than or equal to a calculated threshold. The threshold value we consider is the deadline for the task. The elected CH acquires a list of neighboring nodes that are members of the cluster. Multiple clusters are created depending on the number of nodes and network conditions. On the other hand, if CH moves outside the range of the fog gateway before acknowledging results then the data packets are considered to be dropped. This situation is recovered by task replication that ensures job completion with some increase in job completion time.

After cluster formation, fog gateway offloads the task to CH. The elected CH is further connected with other CHs that are not in the communication range of fog gateway. The CH distributes the tasks among clusters and within cluster based on the capacity of nodes. The capacity of a node includes the processing power of fog node and queue length of the node. In the same context, Fig 1 shows both cases of load-balancing within-cluster and among clusters. The load on the $C1$ is higher, therefore, $CH1$ redirects the load to nearby cluster $C3$. Furthermore, $CH3$ distributes the load within the cluster. Same as $CH2$, which distributes the load between idle fog nodes within the cluster. Hence, we achieve a balanced load and reduced response time of the network.

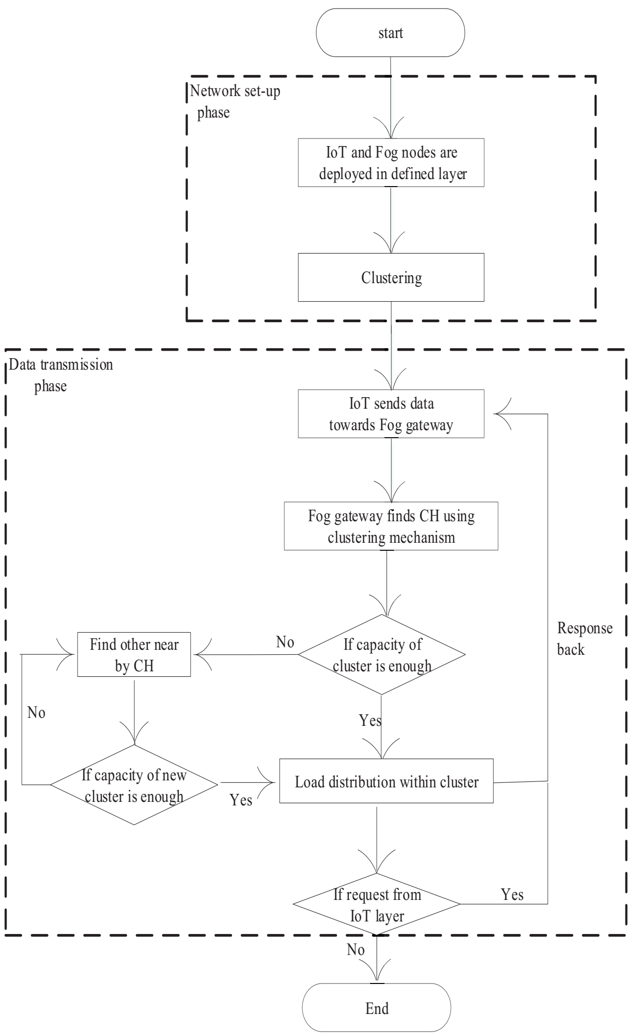

Fig. 2: The workflow of the proposed technique.

Workflow of the proposed methodology is shown in Fig 2. The methodology of the proposed scheme is divided into two phases – the network setup phase and the data transmission phase. In the network setup phase, IoT and mobile fog nodes are deployed in a defined area. The connection among fog nodes is established based on proximity and mobility factor. The proposed scheme classifies the network based on the utilization of resources. Similarly, in the data transmission phase, IoT devices send sensed data towards the fog gateway. The fog gateway identifies critical data and elects CH using the proposed clustering mechanism. The CH has the ability to inform the remaining computation capacity of a cluster to fog gateway node. After that, the gateway node assigns IoT tasks to CH based on computation capacity.

IV. Performance Evaluation

We evaluated the proposed approach using the state-of-the-art NS2 simulator [22]. We considered two network simulation scenarios in terms of the load of the network: balanced load and without balanced load. We evaluated the proposed scheme in terms of end-to-end delay, packet delivery ratio, and control packet overhead in the network.

- **End-to-End delay**: It refers to the time an IoT device sends the data to the fog network and then gets back the result. It includes multiple delays such as communication, queuing, and processing.
- **Control packet overhead**: It is the ratio of the number of beacon packets to the total number of generated packets in the network.
- **Packet delivery ratio (PDR)**: It defines the ratio of successful delivery of the IoT tasks to fog devices to the total number of tasks generated from IoT devices in the network.

We consider a network area of $(1000 \times 500)m^2$ having 31 fog nodes and 10 IoT devices. Table.

TABLE I: Simulation and system parameters

Simulation parameters (units)	Value
Channel	Wireless Channel
Propagation model	TwoRayGround
Mac layer	Mac/802-11
Mobility model	Random Waypoint Model
Mobile fog nodes	31
IoT node	10
Range of mobile fog in clusters	4-6
Range of mobile fog speed (Km/h)	50-90
Simulation time (sec)	500
Transmission range (m)	50-300
Data rate (Kbits/sec)	250

I includes the system and simulation parameters used in our experiments. Moreover, we utilized the spyder anaconda tool [23] (which utilizes matplotlib library) to obtain the graphical results. We calculated the simulation results by averaging the results of 5 runs.

A. *Results and discussion*

In the case of unbalanced load-distribution, selected fog nodes are overloaded and other fog nodes stay in the idle state. Consequently, the processing time of fog nodes closer to the gateway is high. When the processing time of fog nodes increases, it creates congestion in the network. Due to higher network congestion, the average response time of IoT devices is also increased and sometimes may exceed the job deadline. The proposed load-balancing approach efficiently distributes the load among fog nodes that are in idle state to reduce the congestion in the network. Due to the reduction in congestion, the end-to-end delay of the network is also reduced. Fig. 3 illustrates that the end-to-end delay of both approaches is the same at the start. Because initially in both approaches, the data load is received by the fog nodes closer to the IoT devices. In the case of unbalanced load-distribution, the load on fog nodes closer to the gateway gets higher with the passage of time. Hence, it results in an increased end-to-end delay. The proposed approach efficiently

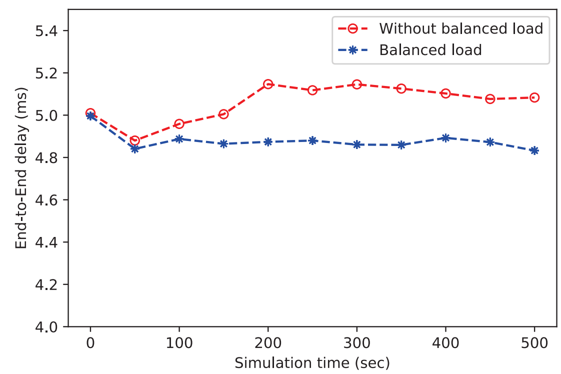

Fig. 3: End-to-end delay of the network.

distributes the load among fog nodes based on the processing capacity of fog nodes. Furthermore, job completion is also ensured before offloading the task to another eligible fog node for processing. Because of this mechanism, the proposed approach reduces the 15% response time as compared to without balanced load approach.

To better illustrate the performance of the proposed methodology in terms of PDR, we simulate load balanced and without balanced load scenarios in vehicular fog environment. Fig. 4 shows the PDR of both policies. Initially, the PDR in both scenarios is the same because the load is distributed to fog devices closer to the gateway. Later on, in the case of without balanced load scenario, the fog nodes are fully utilized according to their capacity that has a considerable impact on the PDR. As a result, the PDR value of the network is reduced. It indicates that a certain number of IoT requests are dropped. However, the proposed load-balancing approach efficiently manages the load among clusters either in the range of fog gateway or not and exhibits an improved PDR. The proposed approach exhibits higher control packets overhead that can be noted from Fig. 5 where the control packet overhead of the proposed scheme is higher than without balanced load approach. The overhead beaconing message

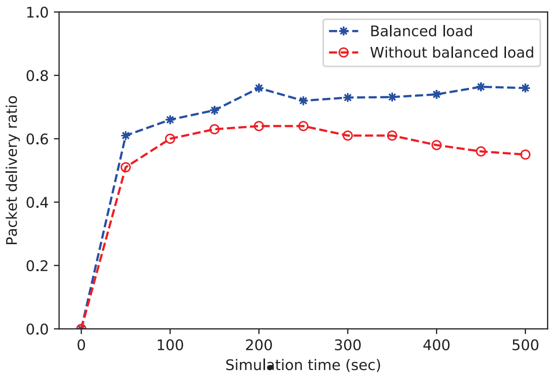

Fig. 4: The variation in packet delivery ratio.

contains the information of vehicles' current position, velocity, and direction. In the case of without balanced load scheme, PDR of the network reduces after some time due to the higher utilization of the fog nodes closer to a gateway. Consequently, the newly arriving requests from the IoT layer are discarded due to the overloading of these fog nodes that results in a reduction in the generation of control packets. While, in balanced load approach, resources are efficiently utilized and more requests are entertained from the IoT layer that results in increased control packet overhead.

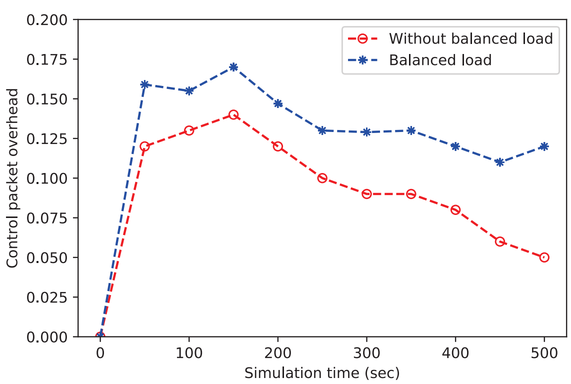

Fig. 5: Control packets overhead.

V. Conclusion

This paper considered the problem of executing IoT jobs on vehicular fog computing environment. It proposed a load-balancing technique that distributes these IoT jobs into clusters of vehicles. The future position of a vehicle is predicted that helps in the automatic formation of clusters. The proposed capacity-based load-balancing algorithm improved the end-to-end delay and throughput by efficiently managing the load at the cluster and network level. Moreover, the proposed scheme avoided congestion by managing vehicular fog nodes and IoT traffic that resulted in lower end-to-end delay and higher throughput. However, the proposed technique exhibited higher control packets overhead incurred by the collaboration among vehicles. In future, we aim to focus on learning-based adaptive and intelligent techniques to optimize the energy and quality of service in the vehicular fog network.

References

[1] T.-C. Hsu, H. Yang, Y.-C. Chung, and C.-H. Hsu, "A creative iot agriculture platform for cloud fog computing," *Sustainable Computing: Informatics and Systems*, vol. 1, p. 100285, 2018.

[2] A. R. Hameed, S. ul Islam, M. Raza, and H. A. Khattak, "Towards energy and performance aware geographic routing for iot enabled sensor networks," *Computers & Electrical Engineering*, vol. 85, p. 106643, 2020.

[3] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, 2017.

[4] L. Zhang and J. Li, "Enabling robust and privacy-preserving resource allocation in fog computing," *IEEE Access*, vol. 6, pp. 50 384–50 393, 2018.

[5] A. A. Mutlag, M. K. A. Ghani, N. a. Arunkumar, M. A. Mohamed, and O. Mohd, "Enabling technologies for fog computing in healthcare iot systems," *Future Generation Computer Systems*, vol. 90, pp. 62–78, 2019.

[6] E. Lee, E.-K. Lee, M. Gerla, and S. Y. Oh, "Vehicular cloud networking: architecture

and design principles," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 148–155, 2014.

[7] S. Bitam, A. Mellouk, and S. Zeadally, "Vanet-cloud: a generic cloud computing model for vehicular ad hoc networks," *IEEE Wireless Communications*, vol. 22, no. 1, pp. 96–102, 2015.

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

[9] D. Ye, M. Wu, S. Tang, and R. Yu, "Scalable fog computing with service offloading in bus networks," in *2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)*. IEEE, 2016, pp. 247–251.

[10] S. Sharma and H. Saini, "A novel four-tier architecture for delay aware scheduling and load balancing in fog environment," *Sustainable Computing: Informatics and Systems*, vol. 24, p. 100355, 2019.

[11] P. Hu, H. Ning, T. Qiu, Y. Zhang, and X. Luo, "Fog computing based face identification and resolution scheme in internet of things," *IEEE transactions on industrial informatics*, vol. 13, no. 4, pp. 1910–1920, 2017.

[12] Y. Sun, F. Lin, and H. Xu, "Multi-objective optimization of resource scheduling in fog computing using an improved nsga-ii," *Wireless Personal Communications*, vol. 102, no. 2, pp. 1369–1385, 2018.

[13] H. Wu, L. Chen, C. Shen, W. Wen, and J. Xu, "Online geographical load balancing for energy-harvesting mobile edge computing," in *2018 IEEE International Conference on Communications, ICC 2018*. Institute of Electrical and Electronics Engineers Inc., 2018, p. 8422299.

[14] R. Mahmud, K. Ramamohanarao, and R. Buyya, "Latency-aware application module management for fog computing environments," *ACM Transactions on Internet Technology (TOIT)*, vol. 19, no. 1, p. 9, 2018.

[15] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.

[16] S. Sarkar and S. Misra, "Theoretical modelling of fog computing: A green computing paradigm to support iot applications," *Iet Networks*, vol. 5, no. 2, pp. 23–29, 2016.

[17] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 3, pp. 361–373, 2017.

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

[19] M. Aazam and E.-N. Huh, "Dynamic resource provisioning through fog micro datacenter," in *2015 IEEE international conference on pervasive computing and communication workshops (PerCom workshops)*. IEEE, 2015, pp. 105–110.

[20] B. Brik, N. Lagraa, A. Lakas, and Y. Ghamri-Doudane, "Finding a public bus to rent out services in vehicular clouds," in *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*. IEEE, 2015, pp. 1–7.

[21] A. K. Saha and D. B. Johnson, "Modeling mobility for vehicular ad-hoc networks," in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*. ACM, 2004, pp. 91–92.

[22] T. Lu, S. Chang, and W. Li, "Fog computing enabling geographic routing for urban area vehicular network," *Peer-to-Peer Networking and Applications*, vol. 11, no. 4, pp. 749–755, 2018.

[23] A. Kadiyala and A. Kumar, "Applications of python to evaluate environmental data science problems," *Environmental Progress & Sustainable Energy*, vol. 36, no. 6, pp. 1580–1586, 2017.