Volume: 7 **Issue:** III **Month of publication:** March 2019

DOI: http://doi.org/10.22214/ijraset.2019.3105

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 7 Issue III, Mar 2019- Available at www.ijraset.com

Intelligent Parking in IoT Based on Fog Computing

Michel Jeny Soundharya.J[1], Nathima Nathima Thore.T[2], Roshni.V.S[3]
[1, 2, 3]*Computer Science and Engineering University College Of Engineering Nagercoil, India*

Abstract: Finding a vacant parking slot can be very stressful in densely populated areas, especially in peak hours. Parking process takes long time leading to significant waste of gasoline and emission of extra vehicle exhaust that harms the environment. Nowadays IoT and fog computing are some of the efficient technologies used. IoT refers to the use of intelligently connected devices and system to leverage data gathered by embedded sensors and actuators in machines and other physical objects. Fog computing essentially extends cloud computing and services to the edge of the network, bringing the advantages and power of the cloud closer to where data is created and acted upon. Recently Fog computing is integrated into IoT in order to promote real-time communication. In fog assisted IoT applications, the time-sensitive data is stored, processed and analyzed on fog nodes close to the devices for providing real-time control within millisecond response time. Hence the idea of smart parking assists drivers in finding desirable parking slots more efficiently using fog computing. Fog nodes deployed at each parking lot cooperate with each other, enabling real-time parking slot information provisioning as well as parking request processing. The uniqueness in our proposed system is that the parking request includes the use of hurry-up signal to help people in emergency.
Keywords: IoT, Fog computing, IR sensor, Arduino controller.

I. INTRODUCTION

In recent years, the number of motor vehicles is explosively increasing which has resulted in parking problems. A vehicle on an average spends only 5% of its lifetime on road while the remaining 95% is spent on parking. Often, drivers need to keep circling around the underground parking lot, or wait at the entrance to the ground parking lot, until a slot is available. Hence the experience of finding a vacant parking slot can be very stressful in densely populated area. This time consuming parking process leads to serious environmental issues due to vehicle exhaust emissions and also results in wastage of gasoline. Also people will not be able to reach their work on time. Many approaches have been proposed to overcome the parking problems. Smart parking systems have been developed and applied in many parking lots assist drivers in finding vacant parking slot. But this approach does not provide an effective solution for people with emergency parking demand. To solve this problem, in this paper, we propose a smart parking architecture which prioritize emergency request over normal request.

This paper focuses on finding a vacant parking slot with the help of a mobile application. Initially the user in demand of parking slot login the system. After successful login, the user receives notification about the available slots in the nearby parking area. The interested users send a parking request to fog nodes. The system responds with reply message to confirm the reservation of request. If there is no vacant slot, then the request will be forwarded to the cloud. The cloud in response finds a vacant slot in the nearby parking areas. Once the reservation is confirmed, the vehicles entering the parking area are counted using the IR sensor placed at the entrance thereby decrementing the available slots accordingly.

This paper is considered beneficial for people with emergency parking demands. The parking request may be normal or emergency. In case of emergency parking request, the people can make use of the “hurry-up” signal using the emergency button provided in the application. Normal requests will be processed in chronological order whereas emergency requests are given higher priority and processed first with the help of MADM (Multiple Attribute Decision Making) algorithm. As a whole, this paper helps people in emergency parking need to achieve priority over other vehicles.

II. RELATED WORKS

As the number of motor vehicles is explosively increasing daybyday it has resulted in parking problem. The parking process takes a long time. To overcome this issue, various schemes have been proposed. Currently, there are several efforts which survey and investigate the smart parking solutions [2].Xiao and Zhu proposed the concept of vehicular fog computing. They turn the connected vehicles into mobile fog nodes and offer cost-effective and on-demand fog computing for vehicular applications [3]. Lin *et al* have presented a smart parking ecosystem and classified the current works by different functionalities and problematic focuses. For instance, they proposed three macro-themes based on the different parking solutions, i.e., information collection, system deployment, and service dissemination [4].Wenyu Zhang et al in, discuss the challenges and opportunities of applying fog computing in IOV. The four functions considered are: mobility control, multi-source data acquisition, distributed computation and multi path data transmission [5]. Rongxing Lu et al proposed a system for searching for a vacant parking space in a congested area or a large parking lot and preventing auto theft. This helps to reduce the searching time for slots of the drivers and saves the fuel. Vijay Paidi et al proposed the usage of smart parking sensors, technologies, applications and evaluate their applicability to open parking lots. These applications show drivers the location of common open parking lots. They are also used to count the number of vehicles in the parking lots [7].

Yanfeng Geng et al proposed anew smart paring system based on resource allocation and reservation which Assigns and reserves an optimal parking space based on the driver's cost function that combines proximity to destination and parking cost by solving a mixed-integer linear programming problem at each decision point in a time-driven sequence [8].A guaranteed parking reservations with the lowest possible cost and searching time for drivers and high resource utilization for parking managers is proposed by Amir et al in [9]. Jianbing Ni in proposed a scheme for securing fog computing in IoT applications. . In fog assisted IoT applications, the time-sensitive data is stored, processed and analyzed on fog nodes close to the devices for providing real-time control within millisecond response time [10].

III. PROPOSED SYSTEM

In existing system the request is processed in chronological order in which people will not be able to find a parking slot. Hence the proposed method overcomes this drawback with the help of Hurry-up signal.Here we are using Arduino as core controller, the IR sensor to count the number of vehicles. The block diagram of our system is shown in Figure 1. The Arduino controller collects the vehicle count from IR sensor and processes that data to update the slot status via Wi-Fi. Some of the important terms used are:

A. IoT

The Internet of Things (IoT) refers to the use of intelligently connected devices and system to leverage data gathered by embedded sensors and actuators in machines and other physical objects. The IoT involves extending internet connectivity beyond standard devices, such as desktops, laptops, smartphones and tablets, to any range of traditionally dumb or non-internet-enabled physical devices and everyday objects.

B. Fog computing

Fog computing, also known as fog networking or fogging, is a decentralized computing infrastructure in which data, compute, storage and applications are distributed in the most logical, efficient place between the data source and the cloud. The fog computing paradigm, also known as the edge computing and considered as one of key enablers of IoT and big data applications [1], brings computation and storage resources to the edge of network, enabling it to run the highly demanding applications while meeting strict latency requirements.

C. Real-Time Services for Fog: With computational and storage capabilities, fog nodes behave as a surrogate of cloud or a private cloud close to IoT devices, handling local real-time computation services. Specifically, the fog nodes deployed at the network edge offer IoT applications and services, and receive data from the IoT devices to make decisions and control the activities of these devices within millisecond response time [11]. Therefore, many delay-sensitive IoT applications can be built to achieve fast decision-making based on collected local data. We showcase some examples of fog-assisted IoT applications, in which fog nodes offer real-time control and fast decision-making for users.

1) *Data Dissemination for Fog:* As intermediate nodes in network, fog nodes take conventional communication functions, such as data aggregation, package forwarding and routing. They submit the data collected by IoT devices to the cloud, and distribute the data received from the cloud to the IoT devices. This two-way data delivery builds network connectivity between the cloud and IoT devices. With storage and communication resources on fog nodes, local data collection and content distribution become convenient and flexible. Moreover, the fog nodes can perform simple processing on the received data. Specifically, the fog nodes can not only deduplicate the data collected by numerous IoT devices to improve communication and storage efficiency, but also select proper audiences to increase the accuracy of content distribution. Overall, the involvement of fog nodes can significantly optimize large-scale data collection and content distribution services.
2) *Decentralized Computation for Fog:* With storage and computing resources, it is possible for multiple fog nodes to cooperatively perform decentralized data computation. Specifically, the fog nodes can not only collaboratively take computation tasks for the cloud, but also assist users to perform heavy computational operations on behalf of proxies. Thus, either the cloud or IoT devices can be free from heavy computational tasks.

D. Hardware Components

1) *Arduino Controller:* The ESP8266 Arduino compatible module is a low cost Wi-Fi chip having an integrated MCU (Micro Controller Unit) which gives the possibility to control I/O digital pins via simple programming language. The Arduino

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 7 Issue III, Mar 2019- Available at www.ijraset.com

controller is shown in Figure 1. This little module allows the MCU to connect to Wi-Fi network and create simple TCP/IP connections. Arduino microcontroller can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators.

Figure 1: Arduino Controller

2) *IR Sensor:* An infrared sensor is an electronic device that emits in order to sense some aspects of the surroundings. An IR sensor can detect the motion and also measure the heat of an object. IR sensor is shown in Figure 2. This sensor is analogous to human's visionary senses, which can be used to detect obstacles and it is one of the common applications in real time.

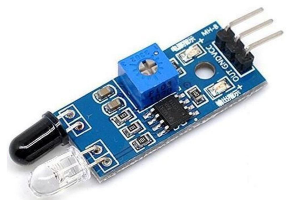


Figure 2: IR Sensor

3) *Relay Driver:* Relays are electromagnetic devices which allow low power circuit to switch a high current ON and OFF switching devices with the help of an armature that is moved by an electromagnet. Relay driver is shown in Figure 3. Driver Circuit is used to boost or amplify signals from micro-controllers to control power switches.

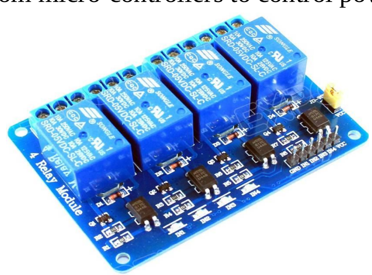


Figure 3: Electromagnetic Relay driver

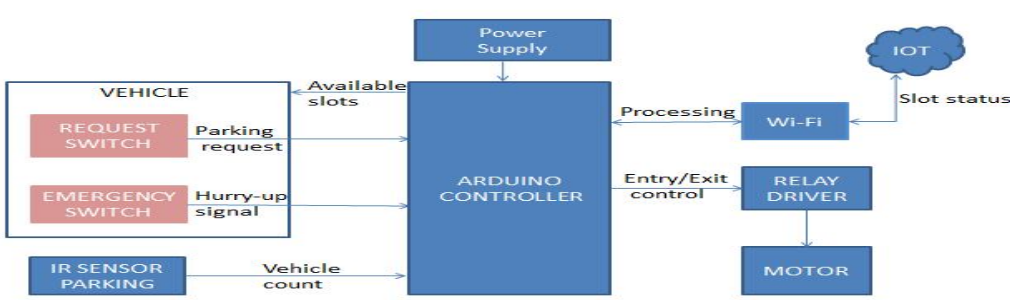


Figure 4: System Architecture

IV. SYSTEM ARCHITECTURE

The advance of Internet of Things (IoT) has enabled much richer means of event monitoring and information collecting by employing various types of sensors in specified monitoring area (e.g., signalized intersections, parking lots, and malls). Accordingly, we assume that the real-time parking information can be obtained in this paper. Actually, most smart parking lots have succeeded in providing accurate and real-time slot occupation information to drivers. Moreover, some smart devices are installed along the road to display the current parking spaces information.

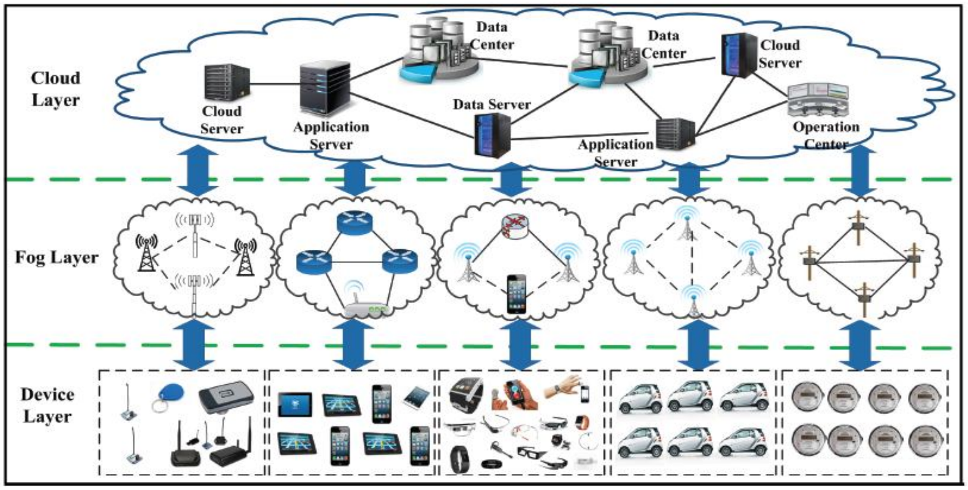


Figure 6: Three-Layer Architecture of Fog Computing for IoT Applications

A. *Information Dissemination*

When a user wants to find a parking slot, he must login to our system. After successful login, a request message is sent to search for a free parking slot. Then, the admin will send back a response message containing the information, including the car park address and the directions to reach it.

B. *Processing Request*

The choice of the car park is based on the current location of the vehicle and the location of the car park. When the user arrives at the car park, he must be authorized to enter. If the information is correct, the user is allowed to park. If the current car park is full, the admin will send a suggestion message that includes information on a new car park, including the address and new directions, with a minimum cost.

C. *Approaching Cloud*

In traffic dense area with multiple parking lots in the vicinity, there may be no vacant parking spaces in all these parking lots in peak hours. Users may select the parking places based on their own preferences, which however can render a longer waiting time if other users with similar preferences behave the same way. To achieve an efficient parking slot allocation with regards to all parking places, the cloud computing is introduced to assist decision making. In our application scenario, each parking lot is equipped with a fog node which is responsible for service provisioning and parking slot prediction, while the cloud nodes located at the remote cloud center take charge of parking slot allocation among multiple parking lots in the coordinated area.

The fog layer consists of network equipment, such as routers, bridges, gateways, switches and base stations, augmented with computational capability, and local servers (e.g., industrial controllers, embedded servers, mobile phones and video surveillance cameras). These devices, called fog nodes in fog computing, can be deployed anywhere with network connections: in a smart phone, on a factory floor, on a roadside unit, in a vehicle or on top of a power pole. The fog nodes are hierarchically distributed between the IoT devices and the cloud servers in the Cloud-Fog-Device framework or above the IoT devices in Fog-Device framework. This layer tends to extend the cloud computing to the network edge. It has certain computing and storage prowess and autonomy to reduce the processing load on resource-constrained IoT devices. Apart from conventional communications (e.g., package forwarding and routing), some real-time and latency-sensitive applications can be relegated from cloud servers to fog nodes. Since the applications are located in the fog nodes only one/two-hop away from devices, they possess regional knowledge about the devices and their owners (i.e., users), e.g., local network condition, users' mobility pattern and precise location information. In Fog-

Device framework, the fog nodes cooperatively offer various services without the involvement of cloud servers, e.g., decentralized vehicular navigation , indoor floor plan reconstruction , smart traffic lights and local content distribution . In Cloud-Fog-Device framework, the fog nodes provide transient storage and real-time analysis on the data collected by IoT devices and periodically sends data summaries to the cloud through the forwarding of other fog nodes located at higher levels in the network hierarchy. The cloud layer in Cloud-Fog-Device framework is a consolidated computing and storage platform that provides various IoT applications from a global perspective. The cloud has significant storage space and computing resources and is accessible for users at any time and from anywhere, as long as their devices are connected to the Internet. It utilizes virtualization technology to achieve the isolation of distinct users' data and IoT applications, such that these applications can independently and concurrently provide different services to distinct users. The cloud receives data summaries from various fog nodes, and performs global analysis on the data submitted by fog nodes and the data from other sources to improve business insight in IoT applications , such as smart power distribution , health status monitoring and network resource optimization . In addition, the cloud also sends policies to the fog layer to improve the quality of latency-sensitive services offered by fog nodes.

V. PROBLEM FORMULATION

In this section, we give an introduction to the involved entities (i.e., fog nodes and cloud nodes), with regards to the corresponding functionality as well as the roles played in the smart parking systems. Then we formulate the evaluation metrics on parking costs mathematically. Finally, the objective function is given which indicates how the parking request is allocated,

A. Fog Node And Their Roles

Parking slots are spatial-temporal resources which can be monitored by various sensors (e.g., surveillance cameras, RFID tags and so on). In the proposed architecture, each parking slot is defined by a six-tuple, *ps*= (*slotID*;*OCC*; *vehID*;*timeStamp*;*DUR*; *SPCL*),of which each element is detailed as follows:

1) Parking slot ID (*slotID*): identification represented by an integer or the corresponding position (*xi*; *yi*) on the Euclidean plane. For instance, for parking slot localization based on surveillance cameras, each parking slot can be denoted by integers; otherwise, a corresponding position (*xi*; *yi*) might be a good alternative.
2) Occupancy (*OCC*): This field denotes the occupancy status of the parking slot. If the parking slot is occupied, set *OCC* to 1 and 0, otherwise.
3) Vehicle ID (*vehID*): If *OCC* equals 1, a unique identification should be used to identify the vehicle occupying the slot. Currently, either the license plate numbers or the OBUs can uniquely label the vehicles. If *OCC* equals zero, this field can be empty or assigned with a default value.
4) Time stamp (*timeStamp*): This field records the timestamp when the vehicle starts to park at the parking space. If *OCC* equals zero, this field can be empty or assigned with a default value.
5) Duration (*DUR*): During the interactions between vehicles and fog nodes, we assume that vehicles with parking demand will provide a rough parking time, so that fog nodes can predict the possible status of parking slots in the near future. Accordingly, we use *DUR* to approximately represent the parking time. If *OCC* equals zero, this field can be empty or assigned with a default value.
6) Special use (*SPCL*): Some parking slots are provisioned exclusively for special purposes (e.g., police cars). We use *SPCL* to denote this type of purposes.

Based on the descriptions above, each parking lot can be regarded as a set of *ps*, which are stored and managed by fog nodes. For instance, when a parking slot is occupied, the information about the parking lot can be updated in real time by fog nodes. Fog nodes disseminate the periodic beacon messages about the current parking lot information to RSUs.

B. Cloud Center And Their Roles

In traffic dense area with multiple parking lots in the vicinity, there may be no vacant parking spaces in all these parking lots in peak hours. Drivers may select the parking places based on their own preferences, which however can render a longer waiting time if other drivers with similar preferences behave the same way [14]. To achieve an efficient parking slot allocation with regards to all parking places, the cloud computing is introduced to assist decision making [15]. In our application scenario, each parking lot is equipped with a fog node which is responsible for service provisioning and parking slot prediction, while the cloud nodes located at the remote cloud center take charge of parking slot allocation among multiple parking lots in the coordinated area. When no parking slot is available in these parking places, the parking requests sent by vehicles will be uploaded to cloud nodes, where the global optimization of parking slot allocation is performed with regards to the specified metrics.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 7 Issue III, Mar 2019- Available at www.ijraset.com

VI. ALGORITHM

Multiple Attributes Decision Making (MADM) algorithm is used in proposed method. MADM is an approach employed to solve problems involving selection from among a finite number of alternatives. An MADM method specifies how attribute information is to be processed in order to arrive at a choice. MADM methods require both inter-attribute and intra-attribute comparisons, and involve appropriate explicit tradeoffs

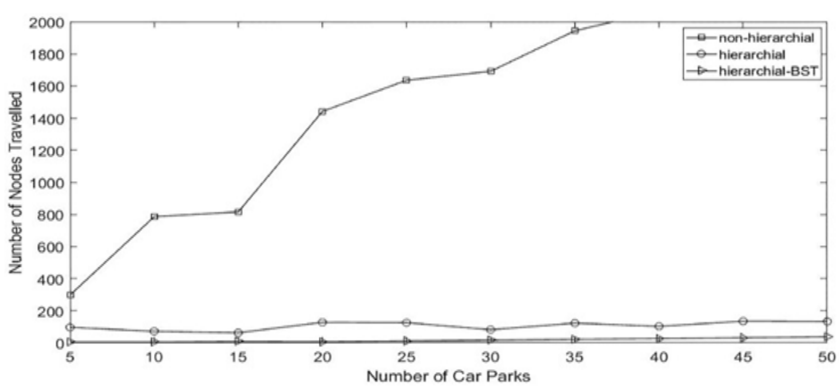
A. Multiple Attribute Decision Making (MAMD) Algorithm

1) Input: Fog Nodes(FN), Parking slot status, parking request.
2) Output: parking allocation decision

void calculate(int TaskValue, int SensorValue, int W_p){
User current location P_s, driving destination P_d,
TempLux=Lux;
If(n>0)
Response to the parking request by providing P_1,P_2,…..P_n individually that are available for user.
Endif
Dis=TaskValue-SensorValue;
if(|Dis|>alfa){
Lux=Lux+Dis;
if(Lux<0){
Lux=0;
Dis=-TempLux;}
foreach(Luminaire Field){
SensorValue=SensorValue+(Dis*D_Matrix[i][W_p]);}
foreach(Luminaire Field active){
Calculate(TaskValue,SensorValue, WorkplaneNo);}
}//End if
Response to the parking request
else
No Action;
}

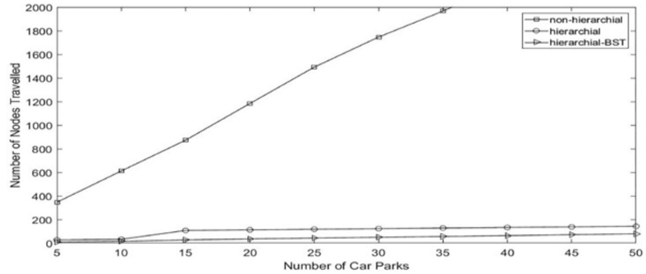
VII. PERFORMANCE EVALUATION

The performance of the proposed system is explained in this section. The uniform and exponential distributions are compared. Both the distributions are continuous probability distributions where the probabilities take any value between two numbers, whereas discrete probability distribution takes only a discrete set of values. The uniform distribution is a distribution that has a constant probability. The exponential distribution is a distribution that describes the time between events in a Poisson process in which events occur continuously and independently.



(a) Uniform distribution

In hierarchical approach with Binary search tree (BST), the number of nodes travelled to find the nearest parking slot is nearly linear. The numbers of car parks almost do not have any effect on search process.



(b) Exponential distribution

On the other hand, when we increase the number of car parks in non-hierarchical approach, number of nodes travelled to find nearest parking slot is increased significantly .Hierarchical approach perform in between but close to hierarchical with BST .When we decrease the number of nodes travelled it directly affects the time consumed for search.

VIII.CONCLUSION

The people in emergency need of parking demand are not able to find a vacant parking slot on time. To solve this problem, the proposed system includes the use of "hurry-up signal" in parking request in case of emergency to achieve priority over other vehicles. The experimental results have proven that our fog computing based smart parking strategies can effectively improve the parking request process.

For the future works, we plan to integrate a mechanism to ensure that people using the hurry-up signal is truly in demand of the emergency parking slot.

REFERENCES

[1] R. Mahmud, R. Kotagiri, and R. Buyya, ``Fog computing: A taxonomy, survey and future directions," in Internet of Things (Technology, Communications and Computing). Singapore: Springer, 2018.

[2] T. Lin, H. Rivano, and F. Le Mouël, ``A survey of smart parking solutions, "IEEE Trans. Intell. Transp. Syst., vol. 18, no. 12, pp. 3229_3253, Dec. 2017, doi: 10.1109/TITS.2017.2685143.

[3] Y. Xiao and C. Zhu, ``Vehicular fog computing: Vision and challenges," in Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (Per-Com Workshops), Kona, HI, USA, Mar. 2017, p. 6-9, doi: 10.1109/PERCOMW.2017.7917508.

[4] 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, doi: 10.1109/ TVT.2016.2532863.

[5] Wenyu Zhang, Zhenjiang Zhang, Han Chieh Chao, "Cooperative fog computing for dealing with big data in the internet of vehicles: Architecture and Hierarchical Resource Management", IEEE Communications Magazine, Vol: 55, Issue: 12, 2017.

[6] Rongxing Lu, Xiaodong Lin, Haojin Zhu, Xuemin , " SPARK: a new vanet-based smart parking scheme for large parking lots", IEEE INFOCOM, 2009.

[7] Vijay Paidi, Hasan Fleyeh, Johan Hakansson, Roger G.Nyberg, "Smart parking sensors, technologies and applications for open parking lots: a review", IET Intelligent Transport Systems, Vol: 12, Issue: 8, 2018 .

[8] Yanfeng Geng, Christos G.Cassandras, "A new smart parking system based on resource allocation and reservations", IEEE Transactions on Intelligent Transport Systems, Vol: 14, Issue: 3, 2013.

[9] Amir O.Kotb, Yao-Chun Shen, Xu Zhu, Yi Huang, "iPARKER-a new smart car parking system based on dynamic resource allocation and pricing", IEEE Transactions on Intelligent Transportation Systems, Vol: 17, Issue: 9, 2016.

[10] Jianbing Ni, Kuan Zhang, Xiaodong Lin, Xuemin Shen, "Securing fog computing for iot applications: challenges and solutions", IEEE Communications Surveys and Tutorials, Vol: 20, Issue: 1, 2018.

[11] Cisco, "Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are," Cisco White Paper, 2015.

[12] X. Masip-Bruin, E. Mar´ın-Tordera, G. Tashakor, A. Jukan, and G. Ren, "Foggy Clouds and Cloudy Fogs: A Real Need for Coordinated Management of Fog-to-Cloud Computing Systems," IEEE Wirel. Commun., vol. 23, no. 5, pp. 120–128, 2016.

[13] K. Liang, L. Zhao, X. Chu, and H.-H. Chen, "An Integrated Architecture for Software Defined and Virtualized Radio Access Networks with Fog Computing," IEEE Netw., vol. 31, no. 1, pp. 80–87, 2017.

[14] P. Yang, N. Zhang, Y. Bi, L. Yu, and X. Shen, "Catalyzing Cloud-Fog Interoperation in 5G Wireless Networks: An SDN Approach," IEEE Netw., to appear.

[15] S. Yi, C. Li, and Q. Li, "A Survey of Fog Computing: Concepts, Applications and Issues," in Proc. of Mobidata, 2015, pp. 1–6.