EESM6980 Project Report
MEC Application in Vehicular Energy Network

Yitong Li

Catalog

1. Introduction and Background

Cloud computing has gained significant popularity over the past ten years as a means to store and process vast amounts of data. Despite the advantages of cloud computing, there is still a need to bring cloud features closer to user devices. This is because the present cloud model relies heavily on a Wide Area Network (WAN) to communicate between user devices and cloud servers. This can result in delays, network congestion, and other performance issues. The Edge Computing paradigm, which seeks to deliver distributed computing and storage with context awareness at the periphery of networks, was born as a result of this. This section begins with a review on cloud computing, fog computing, and mobile edge computing, followed by a discussion of their advantages and a side-by-side contrast.

1.1 Cloud Computing

For more than ten years, the capacity of cloud computing to handle intensive calculation has made it the solution for large-scale data storage and computation in the IT sectors for its strength in many aspects. First, customers only pay for the services they use and incur no upkeep costs, saving money. Additionally, cloud software is safe and scalable. Since around 2005, cloud computing service like AWS was started and they provided service of storage and computation. After that, big companies like Google, Oracle, Microsoft also began to provide cloud computing services. Cloud computing has had a significant impact on how we live, work, and learn[1][2].

Cloud computing includes three basic components as shown in **Figure 1**:

- Client computers: The end user employs client devices to engage with the cloud. This means that individuals utilize devices such as smartphones, tablets, or laptops to connect to the cloud and access its resources or services.
- Distributed servers: Although the computers may be spread out over various locations, they can still communicate with one another.
- Data centers: Data centers serve as both data storing facilities and collections of computers.

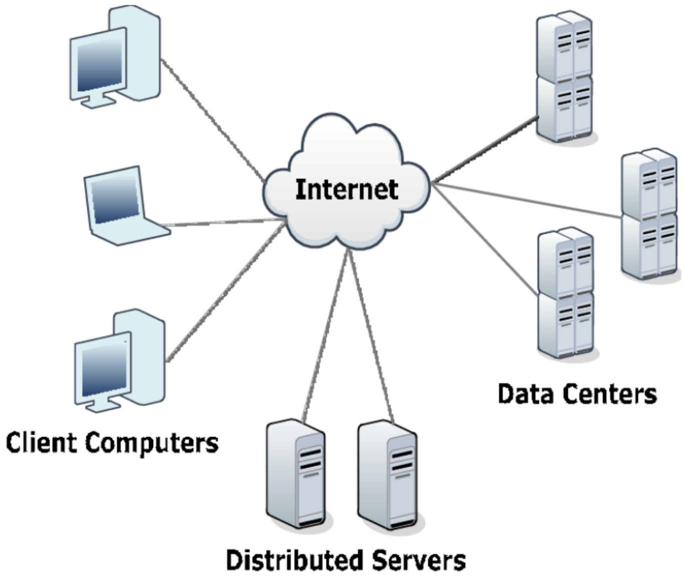

Figure 1 Architecture of Cloud Computing

Cloud service model includes three types as following:

- SaaS (Service as a Service): Software as a service is the method of delivering applications as a service over the internet. Instead of installing software on their device, users can simply utilize this type of service online. Without managing complex software and hardware or buy software/ hardware, user just need to have internet connection and access to application, like Microsoft Office 365.
- PaaS (Platform as a Service): Platform as a service provides a platform, where they can deploy their own software and coding. Users are able to construct their own applications such as Linux, MySql, that can operate on the provider's infrastructure.
- IaaS (Infrastructure as a service): Infrastructure as a service offers a wide range of processing tools, including storage, network, operating system, hardware. Internet connection is required to use the program.

1.2Fog Computing

Fog computing is an extension of cloud computing that was proposed by Cisco in 2012. The main idea behind fog computing is to bring cloud resources and services closer to the network edge, where end-users can access them more efficiently. Fog computing offers elastic resources and services to end-users at the network edge, aiming to improve the performance and reliability of cloud-based applications and services. Fog computing employs a highly virtualized platform to provide computation, storage, and connectivity between end devices and conventional cloud servers. This allows fog computing to leverage the resources of both the cloud and the edge to provide optimal performance and scalability. Fog computing uses emerging techniques such as software-defined network (SDN) and network function virtualization (NFV) to create a flexible and maintainable network environment. This allows fog computing to provide programmable interfaces, application-aware control, traffic monitoring, VM migration, and resource allocation[3].Reliability, connectivity, delay, capacity are key metrics for fog computing.

- Reliability. To enhance reliability, it is common to perform regular checkpoints, reschedule failed tasks, or duplicate tasks for simultaneous execution. These techniques are often employed to improve the dependability of a system or application by ensuring that critical tasks are performed correctly. Replication appears to be more hopeful but requires multiple fog nodes to cooperate since rescheduling and checkpointing may not be suitable for highly volatile fog computing environments.
- Connectivity. It is suggested to use a live AP association approach that not only accomplishes a low throughput but also reduces processing overhead. Meanwhile, the end user's choice of fog server can have a significant impact on speed.
- Delay. Services will be delayed when a fog node computes data that is spread across several neighboring nodes because calculation cannot begin before data aggregation. In order to satisfy the required delay, a spatio-temporal event processing system that utilizes fog computing in an opportunistic manner is needed. The idea behind RECEP is to reuse computation and use fewer resources by taking advantage of similar data interests and findings that are okay to be incorrect.
- Capacity. In order to process search queries for content scattered in fog nodes, search engines may need to be rebuilt due to the dynamic data location and high total capacity volume in fog computing.

1.3From Cloud to Edge

By centralizing all of the duties on a cloud server, cloud computing facilitates effective data handling. Although the speed of data processing is accelerating quickly, the network's transfer capacity has reached a standstill. As a result, the cloud computing model is experiencing a bottleneck due to communication speed. Numerous forces are pushing computing from the cloud to the edge. Internet of Things (IoT) is growing in importance on the worldwide market as a result of the creation of new generation mobile communication techniques (i.e. 5G) as shown in **Figure 2**. Cloud computing is no longer adequate for IoT because there will be more than billions of data in a few years, and there isn't enough capacity to move all of these tasks to the cloud side for processing. Second, privacy security is a major concern because it is unsafe to upload all raw data to the cloud. Finally, because most IoT end nodes have limited energy resources, sending significant amounts of data via wireless transmission can be quite energy-intensive[5].

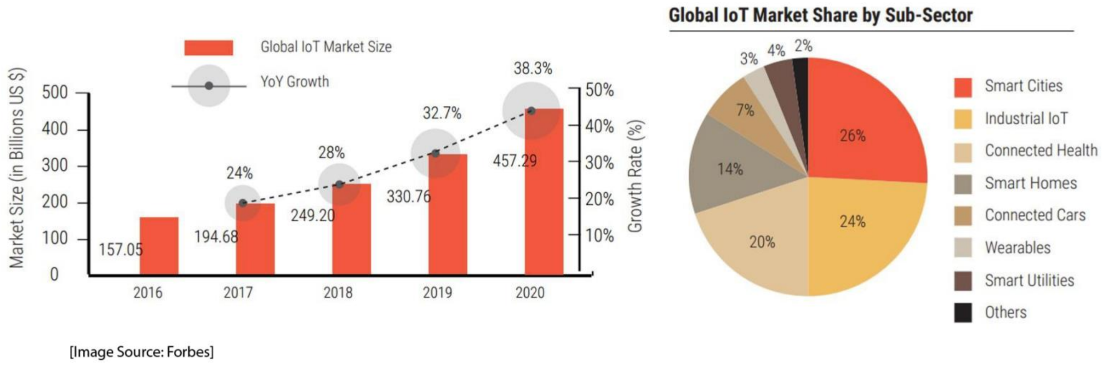

Figure 2 Global IoT market size

The end device's shift from data consumer to data creator is another motivating element. Today's mobile devices generate a significant quantity of data, which transforms edge devices from traditional data consumers to data producers. More feature location at the edge is necessary for this. Video now makes up most of the internet bandwidth as social media being widely spread as shown in **Figure 3**. Before uploading, the picture or video must be compressed and adjusted to the appropriate quality at the border due to their size.

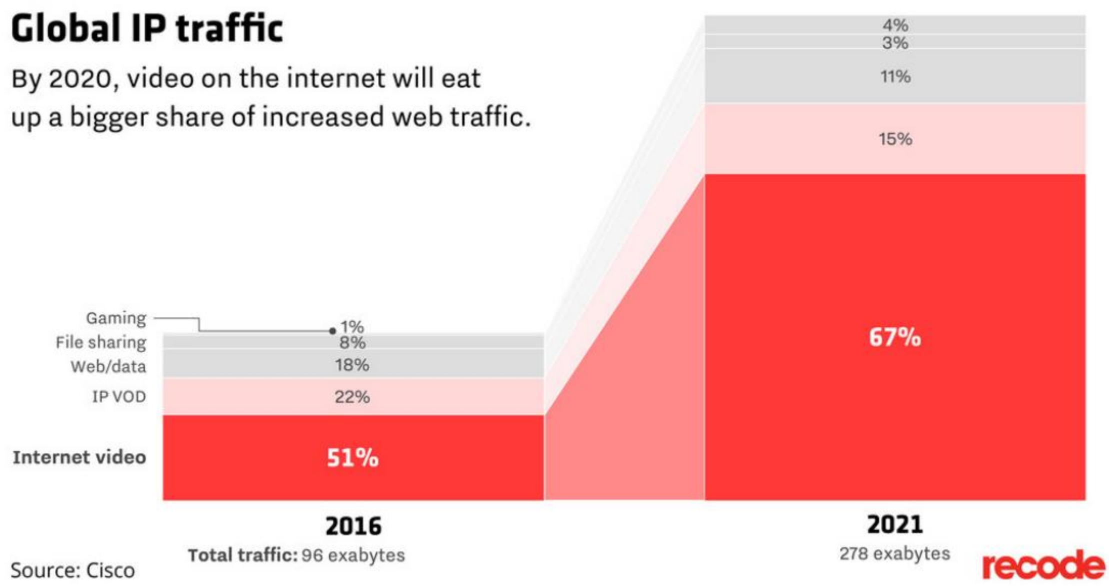

Figure 3 Global IP traffic growth

Edge computing has evolved through the creation of a number of new technologies, including Cloudlet, Fog Computing, Micro Data Centers, and Mobile Edge Computing. Here we focus on two particular cases, Fog Computing and Mobile Edge Computing.

1.4Mobile Edge Computing

MEC was first introduced as the idea of "Mobile Edge Computing," but ETSI expanded it to "Multi-access Edge Computing" in 2016. This implies that edge computing has now been stretched from cellular networks for telecommunications to other wireless access networks. MEC (Multi-access Edge Computing) is an application of edge computing that aims to bring computing and storage capabilities closer to the network's edge within the Radio Access Network (RAN). The primary goal of MEC is to reduce latency and enhance context awareness. Multiple versions of MEC host are running on the server, which can perform processing and storage on a virtualized interface. A Mobile Edge Orchestrator manages the Mobile Edge apps and keeps track of the MEC hosts' services, resources, and network architecture. It also keeps track of the MEC hosts.

MEC servers are capable of providing real-time data to the network, like its capacity and load. It can also provide data on the end devices linked to them, e.g. network data and location information[4]. The queries are received by MEC computers, which are base station co-located, at the Mobile Edge Orchestrator from end users. The ME Platform Manager provides updates on the resources that are accessible to the orchestrator, which keeps an inventory of the apps that are operating on the underlying ME hosts. Requests will be dealt with differently depending on the application's status. If an application is already operating, the request is forwarded to it. If the application is not currently running, but the platform has the capability to run it, and there are available resources, the application will be launched and the request will be authorized. However, if there are no resources available, the request will be sent to the cloud via the network's backbone for processing.

1.5 Comparison between Fog Computing and Mobile Edge Computing

Here, we compare solutions on seven distinct elements of mobile edge computing and fog computing as shown in **Table 1**. The implementation of fog computing proposes the presence of Fog Computing Nodes (FCNs) at various locations between end devices and cloud Data Center Networks (DCNs) to determine the location of nodes, giving users more options for the devices they can use as FCNs. But for mobile edge servers, the storage capacities and computation are usually larger than fog computing. In contrast to other applications that use specialized devices as nodes, the diverse nature of fog devices necessitates the use of abstraction layer. The nearest FCN may actually be several hops away, as the initial router connected to the end device may not have the necessary resources to support the FCN framework. This truth explains why fog computing supports inter-node communication, whereas mobile edge computing uses direct connections made by the devices over WiFi and mobile networks at the base station, respectively.

Table 1 Comparison of Edge Computing implementations

	Fog Computing	Mobile Edge Computing
Node devices	Routers, Switches, Gateways	Server running in base station
Node location	Varying between end devices and cloud	RNC/ macro base station
Software Architecture	Fog abstraction layer based	Mobile orchestrator based
Access Mechanisms	Bluetooth, WiFi	Mobile networks
Internode Communication	Supported	Partial

From several other performance metrics, we present how they will influence fog computing and mobile edge computing paradigms as shown in **Figure 4**.

- Proximity: Proximity can be defined in physical proximity and logical proximity. Physical proximity describes the real separation between the final gadget and its more advanced layer of computation. For instance, physical closeness is important in cloud computing because delays occur when end devices and data centers are located on different countries. Mobile edge computing works better in this regard because processing is delegated to nearby edge devices. The amount of steps between the end layer and end device is known as logical proximity. The likelihood of running into delays along the communication route increases with the number of steps, which adds latency.
- Power consumption: The power consumption on end devices is a major contributing factor as many edge devices are resource constrained. The article[9] proposed that energy consumption within LTE and radio network is much greater than WiFi's. Because mobile edge computing utilizes a wireless network for communication, energy usage when reaching mobile edge nodes is greater than with fog computing.
- Computation time: The amount of time needed for duties to be computed and transmitted is known as computation time. Mobile edge computing has been shown to work better in this area because resources are shared and dynamic resource allocation is used.
- Context awareness: A crucial factor in determining how well a device can gather knowledge about the network and devices around it is context awareness. Mobile edge computing works well in this situation because servers receive information about the position of the devices, the burden on the network, and the network's capability as a result to their placement in RNCs. Fog computing has less context awareness than MEC because its nodes are typically devices with a limited perspective of the network, such as switches or routers.
- Access mediums: The range of connectivity, support for various kinds of devices, and the bandwidth made accessible to end devices are all influenced by the access mediums and methods. End devices can link to peripheral layer nodes using a variety of channels, including Bluetooth, WiFi, mobile radio networks, etc. Fog computing supports a number of protocols, such as Bluetooth and Zigbee, which enables constrained devices that lack memory necessary to operate an HTTP stack

to link to FCNs and offload computation and storage.

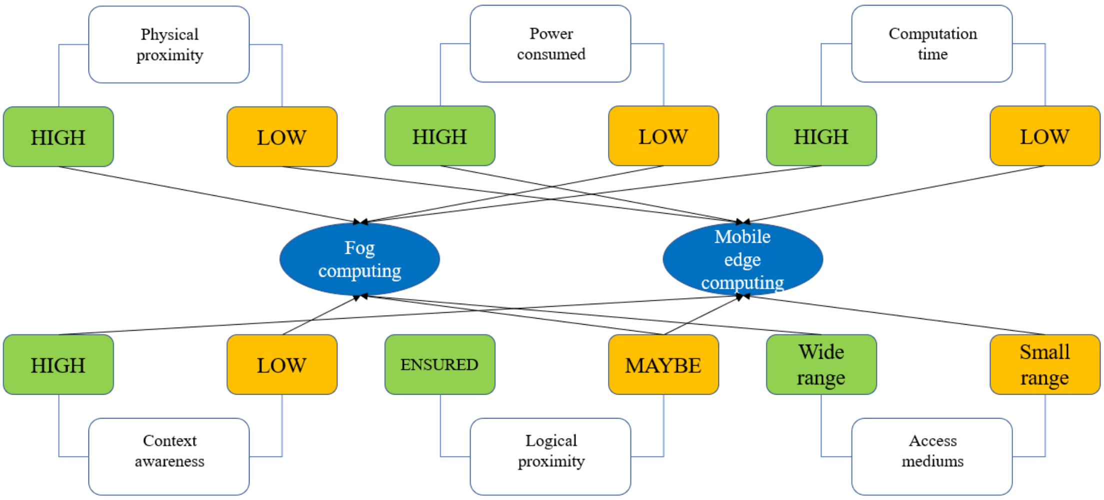

Figure 4 Comparison between fog computing and mobile edge computing

1.6 Application of MEC in Vehicular Energy Network

Currently, research on vehicular energy networks primarily focuses on managing the energy consumption of either battery-powered RSUs or EVs. However, to meet the diverse needs of communication, computing, and storage in the Internet of Vehicles (IoV), an energy-efficient scheduling method can be employed for MEC-enabled RSUs. This approach aims to minimize energy usage while meeting the latency requirements of computational tasks in vehicular energy networks. MEC is a developing ecosystem that brings communication and computing resources closer to end devices, extending the centralized computing capabilities of the cloud to the edge in close proximity to those devices. MEC is adaptable to optimize network resources locally and to host applications that require intensive computing. As a result, computing at the edge can have a significant impact on network performance in terms of execution latency and energy consumption in various ways[6]. Some promising aspects for energy-efficient scheduling in vehicular energy network are discussed:

- MEC-Enabled Offloading: Offloading computation is essential for accelerating computational processes and conserving energy in vehicular energy networks. The offloading method enabled by MEC in vehicular energy networks can assist in selecting appropriate MEC servers for task management. This method takes into account both vehicle mobility and computational tasks to make comprehensive offloading decisions[7].
- Collaborative MEC: Resource-constrained devices are unable to manage all of their computational tasks, making it nearly impossible. Therefore, collaborative MEC has the potential to integrate diverse computing and storage resources among connected entities.
- Energy-Efficient Scheduling: Having knowledge of vehicular routes can be highly beneficial in reducing the energy consumption of RSUs. MEC is a crucial

component in vehicular energy networks, where vehicles connect to RSUs that are equipped with MEC servers to perform real-time computational tasks.

2. Case study: Mobile Edge Computer Geared v2x for E-mobility Ecosystem [8]

2.1 Introduction

The rise of electric vehicles has given rise to new concerns about e-mobility. To predict the dynamic state of Charging Stations (CSs), the possibility of reserving charging by taking into account the expected arrival time and charging time of an electric vehicle (EV) has been investigated[10].

2.2 Provisioning and Analysis of MEGEE

In this part, we first introduce the network entities of MEGEE as shown in **Figure 5**.

1. Electric Vehicle (EV): The Expected Earliest Available Time for Charging (EEATC), which indicates whether a charging spot is available at CS, is one piece of information that electric cars receive from CSs via the MEGEE. EV also has a State of Charge (SoC), which carries out the following two tasks:
   - If the ratio of its leftover energy is less than the SoC cutoff, EV should locate a CS.
   - When deciding on a CS, EV also shares its reservations about where to charge, when to appear, and how long to charge.
2. Charging Station (CS): EVs use On-Board-Units (OBUs) to determine the position of CSs. Each CS has numerous charging points that can power multiple EVs simultaneously. GC will determine the EEATC for CSs by considering various local queuing conditions. These conditions include the number of EVs currently parked, their remaining charging time at the CSs, and the charging appointments of other EVs. Through MEGEE, each CS performs the following functions:
   - To mine and collect valid information from EV charging reservations and aggregate them, report to GC alone with CSs' local queuing information.
   - To report its EEATC to EVs.
3. Global controller (GC): The cloud server computes EEATC for CSs based on information from CSs and EVs, which is then used to pick CSs. GC also sends requested energy from grids to CSs, as described in Charging Scheduling, based on the EV's charging reservations. MEGEE leverages the MEC idea by providing computing, communications, and storage capabilities to the CSs in order to manage charging demands from neighboring EVs.

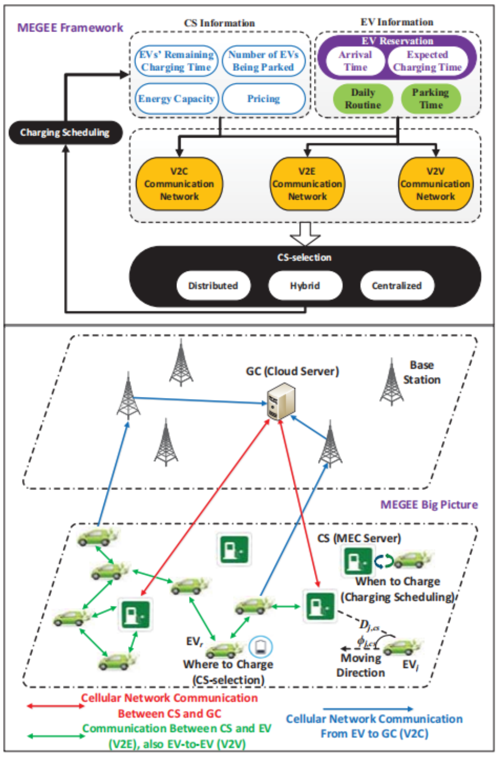

Figure 5 Network entities

Three different system architecture distributed design, centralized design and MEGEE, are presented in **Figure 6**.

1. Distributed design

   The distributed design has the benefit of enhanced privacy protection because it depends on EVs to make CS selection decisions locally. The following stages are involved in the operation of this system design:

   1) Every CS (with interval δ) periodically broadcasts its EETAC to all the EVs in the regular vicinity.
   2) Following CS selection, the EV_r charging order is communicated to GC via cellular network.
   3) Based on data from CSs and confirmed EVs' charging reservations, the EEATC of CSs is calculated and reported at β.

2. Centralized design
   1) EV_r sends its request to the GC.
   2) Based on observed CSs' local queuing data and charging reservations provided by other EVs, and the choice is communicated to EV_r.
   3) EV_r accepts decision, further notifies its charging booking to GC.
3. MEGEE design
   1) Based on cellular network, each CS periodically (with interval δ) broadcasts its EEATC to all EVs based on cellular network. Here, it is assumed that all CSs' broadcast at the same time, but MEGEE can also handle the situation

where various CSs use various information dissemination frequencies.

2) The EV, designated as EVr, creates its charging reservation after deciding where to charge. This phase could be communicated to any CSs (collocated with the MEC server) through V2V and V2E, or alternatively through the cellular network to the GC.
3) The charging reservation of an EV refers to the "one-to-any" paradigm in this stage because an EV can send information to any one of the CSs.
4) Each CS will handle the arrangements for charging reservations. Only bookings for charging for which EVs arrive later than a time bound β, will be further notified to the GC. When reaching the GC, each CS combines the charging orders for those mined EVs while time bound is approaching β. If the EVr charging reservations haven't already been provided through V2V&V2E, they will be sent straight to the GC through V2C.
5) The GC predicts the EEATC of CSs and inform as them that it will be released at β.

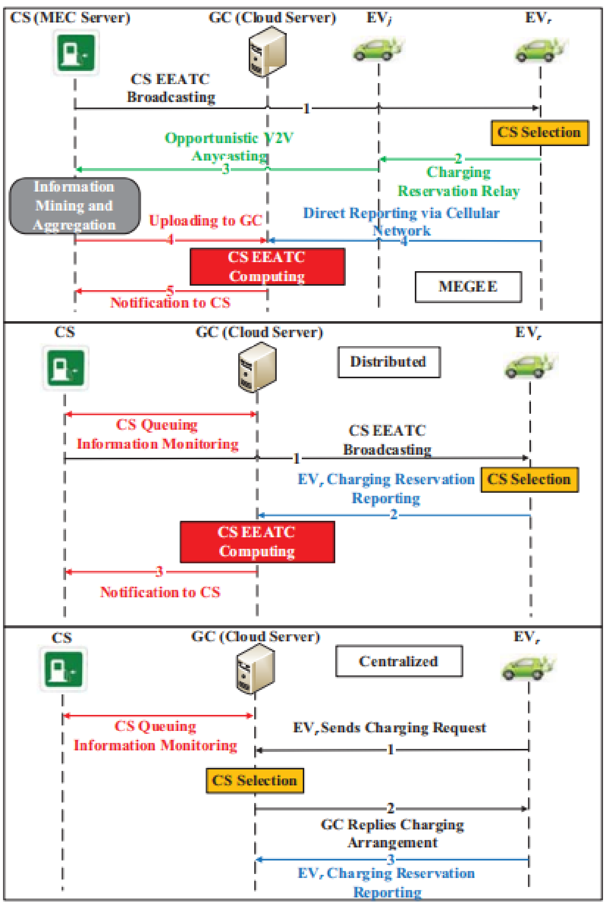

Figure 6 System architecture of distributed, centralized and MEGEE design

2.3MEGEE’s V2X Communication System Design

The EV charging cycle is described in **Figure 7**.

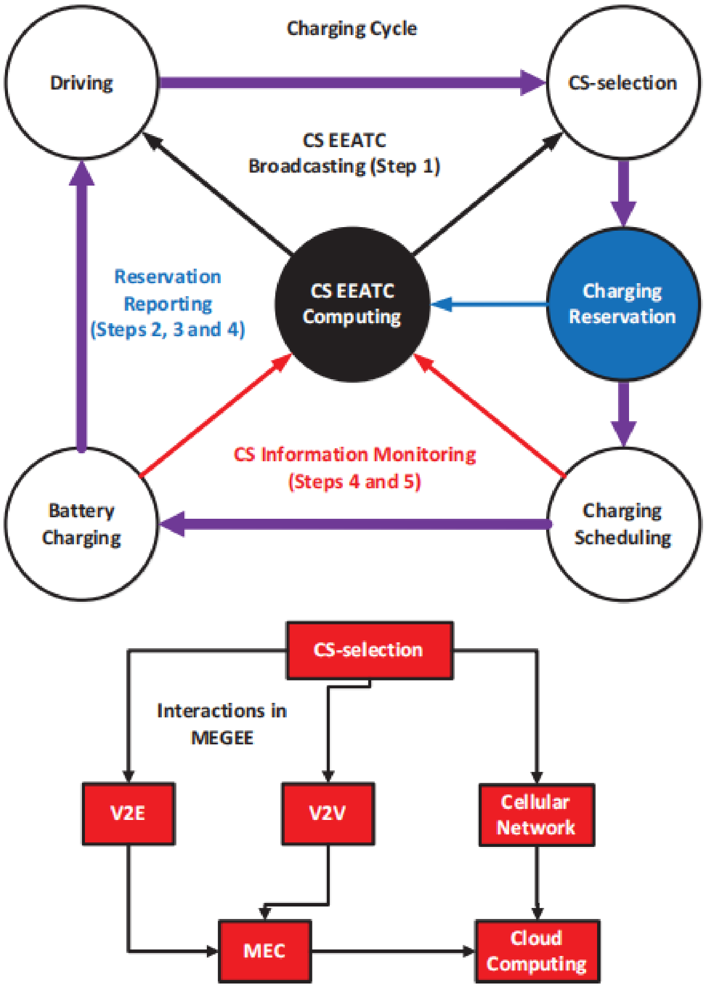

Figure 7 MEGEE charging system cycle

Driving: The EV operates on the street only if its energy level exceeds a specific limit.
CS-selection: After the EV reaches its limit, it independently determines where to recharge in the immediate area.
Charging reservation: The EV produces a charging booking and transmits it to a CS through V2E and V2V communication or to an electricity grid via V2C communication.
Charging scheduling: After the EV arrives at its assigned CS, the station determines the order of priority for charging in the queue.
Battery charging: The EV's battery is charged until full volume once it's scheduled for charging, and then it transitions to the **Driving** phase.

2.4 Methodology and Theoretical Analysis

2.4.1 V2X Technologies in Different Cases

CS-selection and Charging Scheduling basically include three techniques, V2E, V2V and V2C, which can be applied in different cases.

1. **V2E**: The mentioned approach can be utilized when a neighboring EV (denoted by EV_j) is simultaneously undergoing the process of planning for charging and is moving towards its own designated CS (with arrival time denoted by A_j, doesn't need to be the same CS selected by EV_r). Under such circumstances, certain limitations are applicable, which are outlined below:

$$A_r \geq \beta \text{ and } A_j < \beta \text{ and } A_j < A_r \quad (1)$$

   Once the above restrictions are in place, EV_r can be prompted to transmit the EV's individual charging booking (directed to its chosen CS with an arrival time of A_r) to EV_j.
   In the event EV_r's estimated time of arrival is prior to the dissemination of data during the subsequent time slot, that is $A_r < \beta$, CS will not take action to analyze this information, as a result, it will not be conveyed to the central entity (GC). The rationale behind this is that the charging reservations of EVs are critical for projecting the prospective condition of the CS. In the case of any EV that arrives prior to time slot β, it will be viewed as local information for the CS, as opposed to remote projected information that extends beyond β. This is also the explanation for why the arrival time of A_j must be prior to β.

   From a different perspective, the goal is to locate a neighboring that can swiftly transmit the charging reservation of EV_j to any CS prior to time slot β.

   Consequently, the EV_j will be chosen based on the criteria outlined by $(A_j < A_r)$.
2. **V2V**: Under such circumstances, none of the EV is moving towards their chosen CS. Therefore, a communication path using vehicle-to-vehicle technology is created to assist with delivering information. The likelihood (denoted by P) of EV_j successfully delivering the charging reservation for EV_r to any of the CSs is determined, where:

$$P = 1 - (1 - P_{cs})^{N_{cs}} \quad (2)$$

   Where $(1 - P_{cs})$ represents the likelihood that the information is not delivered, P_{cs} is described as the probability of successful delivery.
   A routing protocol is suggested that incorporates anycasting, which follows the roadmap of MEGEE in **Figure 5** and takes into account the delivery probability as defined in equation (2). It concerns the following information of EV_j, including

rate of motion S_j, the direction in which they are moving relative to any CS $\emptyset_{j,cs}$, and distance $D_{j,cs}$ between CS and EV_j. Since the charging reservation for EV_r must be delivered before β, the time remaining until β is defined as $(M = \beta - T_{current})$, where $T_{current}$ represents the current time in the network.

An earlier study on Delegation Geographic Routing (DGR)[11] with anycast functionality has been expanded. The duration of the interaction between an EV_j and CS is denoted as $\frac{D_{j,cs}-X}{\emptyset_{j,cs}\times S_j}$, where X represents the V2V communication range (also for V2E). Then it can be derived that:

$$P_{cs} = \begin{cases} \frac{M-\frac{D_{j,cs}-X}{\emptyset_{j,cs}\times S_j}}{M} & if\ \left(\emptyset_{j,cs} < \frac{\pi}{2}\right)\ and\ \left(M > \frac{D_{j,cs}-X}{\emptyset_{j,cs}\times S_j}\right) \\ 0 & else \end{cases} \tag{3}$$

Using Equation (3), only EV_j heading towards the CS will be considered for computation, subjected to $(\emptyset_{j,cs} < \frac{\pi}{2})$. Additionally, the time for an EV to reach the CS must be earlier than M to meet the deadline for the GC to collect network information. Consequently, the probability P_{cs} increases as the number of CSs that the vehicle EV_j encounters quickly increases.

Through delegation forwarding optimization, the number of hops is minimized, resulting in a decrease in the expense of communication for V2V transmission from $O(N_{ev})$ to $O(\sqrt{N_{ev}})$.

3. **V2C**: EV_r may need to use the cellular network to report its charging reservation to the GC (through the associated base station). This method can be utilized only when $(A_r \leq \beta)$, while the charging reservation has not yet been delivered to the GC through V2E&V2V communication.

2.4.2 Communication Cost among Distributed System, Centralized System and MEGEE

The communication cost among distributed, centralized and MEGEE is illustrated in **Table 2**, the number of EVs and CSs is denoted as N_{ev} and N_{cs}, respectively.

1. Distributed: The transmission solution used in the distributed system is the cellular network. The cost can be calculated as $O(R)$, where R denotes as the number of EVs' charging requests. The delivery cost of charging reservations from CSs to GC has been specified by $O(\frac{N_{cs}}{\delta})$, as GC gathers information from CSs at a regular interval δ.
2. Centralized: The expense of communication for the grid to convey charging

requests and booking s from EVs is identical and equal to $O(R)$.

3. MEGEE: The cost of communication for CS to transmit it EEATC to all EVs is $O(\frac{N_{ev}}{\delta})$, where δ denotes each CS's the EEATC broadcasting interval. The cost of communication for making a reservation depends on the available options:
   1) The expense of sending reservations for charging from CSs to GC is equivalent to a distributed approach.
   2) MEGEE experiences a communication cost of $O(\sqrt{N_{ev}})$, as the charging reservation is delivered to a CS through V2V anycast.
   3) In case the charging booking is not transmitted before β, a cellular network can also be used as an alternative solution. In this situation, MEGEE experiences a cost of $O(R)$.

Thus, the reservation cost of MEGEE is scaled among $O(\frac{N_{cs}}{\delta})$, $O(\sqrt{N_{ev}})$ and $O(R)$.

Table 2 Comparison among distributed, centralized and MEGEE

	Communication techniques	CSs' EEATC broadcasting cost	Reservation cost
Distributed	Cellular network	$O(\frac{N_{ev}}{\delta})$	$O(R)$
Centralized	Cellular network	$O(R)$	$O(R)$
MEGEE	V2V, V2E, V2C	$O(\frac{N_{ev}}{\delta})$	$[O(R), O(\frac{N_{cs}}{\delta}), O(\sqrt{N_{ev}})]$

In the actual deployed system, normally $N_{ev} \gg N_{cs}$, and $N_{ev} \leq R$, represents the demand for long-term charging, rises in a linear fashion as the quantity of EVs increases. Based on this, we can conclude that communication cost of MEGEE is significantly lower than the other two cases.

2.5 Performance Evaluation

The evaluation of effectiveness is conducted in the Opportunistic Network Environment (ONE) located in Helsinki, and this paper introduces three different design proposals: Centralized, Distributed, and MEGEE. The evaluation metrics are presented below:

- Charging waiting time: The mean time it takes for an EV to arrive at the selected CS and the duration needed for the vehicle to complete a full charge.
- Number of charging vehicles: The number of vehicles that have achieved a full battery charge at a specific CS.
- Number of reservations: The overall count of bookings received and registered by CSs, or directly reported by EVs that have arranged for charging.

We access the effectiveness from two distinct perspectives: the charging performance of EVs and the cost of communication. This is done to demonstrate that the suggested MEGEE architecture can achieve comparable charging performance while significantly reducing communication expenses.

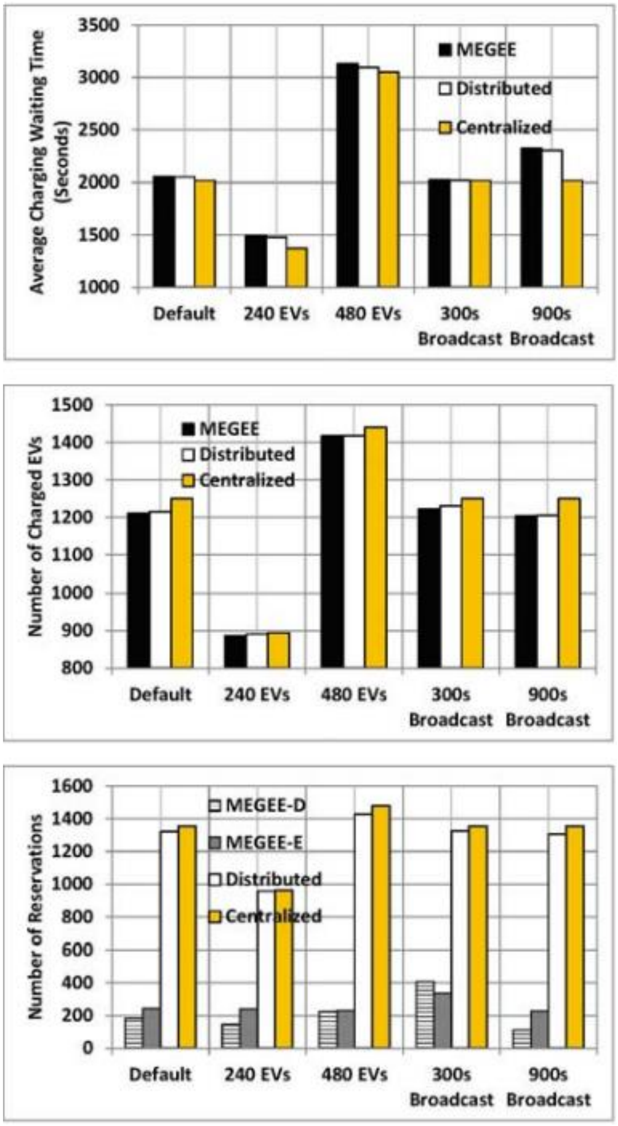

Figure 8 Result evaluation

2.5.1 Charging Performance

As shown in **Figure 8**, the rise in EV density at to charging hotspots in CSs is observed across all three proposed schemes, from 240 to 480. When the CS broadcast interval is shortened from 900s to 300s, the charging performance improves, as reflected by a reduction in wait time and an increase in the number of EVs being charged. This is due to the fact that in the decentralized designs (Distributed and MEGEE), the recorded information about the EVs is more precise, resulting in more accurate decision-making when it comes to charging reservations.

Comparing three options presented (centralized, distributed, and MEGEE), the centralized scheme is considered the most efficient because real-time information can be transmitted instantly, resulting in optimal performance. However, the distributed and MEGEE schemes can also approach the best performance, indicating that decentralized decision-making is also effective.

2.5.2 Communication Cost

EVs in both centralized and distributed frameworks utilize cellular networks to communicate with GCs and CSs, whereas MEGEE employs V2V, V2E, and V2C modes of communication. Due to this distinction, there may be variations in the broadcasting cost and charging reservation cost of their respective Expected Earliest Available Time for Charging (EEATC).

In the above theoretical analysis, it's proved that the EEATC broadcast cost of distributed and MEGEE is $O(\frac{N_{ev}}{\delta})$, which is smaller than the one of centralized case ($O(R)$). For charging reservation cost, MEGEE can be scaled among [$O(R)$, $O(\frac{N_{cs}}{\delta})$, $O(\sqrt{N_{ev}})$], which is much smaller than the one of distributed and centralized scheme ($O(R)$).

Also, MEGEE-D (V2E/V2V) and MEGEE-E (V2C) differ slightly in their charging reservations process, which is influenced by N_{CS} (the number of CS) and δ (broadcasting interval).

3. Simulation Analysis

In this part, we build centralized and distributed architecture to verify the results within the paper.

3.1 Experimental Setting

The simulation performance is evaluated via ONE, under the Helsinki city with an area of 4500*3400 m^2. Each EV is assigned with random destination in scenario throughout simulation, this repeat when the EV's energy level is larger than 50%. The driving route towards is based on Shortest Path First algorithm. There are 230 EVs in total, with each speed regulated with [8.3, 13.9] km/h. In total, 5 CSs are deployed in the city, with large enough electric energy, each CS is equipped with 5 charging slots. The CS's broadcast interval δ is 600s, the GC's broadcasting notification time bound β is 400s.

3.1.1 Centralized Architecture

In this scenario, GC acts as the cloud server, the information within every EV and CS is known at GC. Therefore, when EV's energy reaches the threshold, CS selection will be performed by GC. Detailed implementation architecture is shown in **Figure 9**.

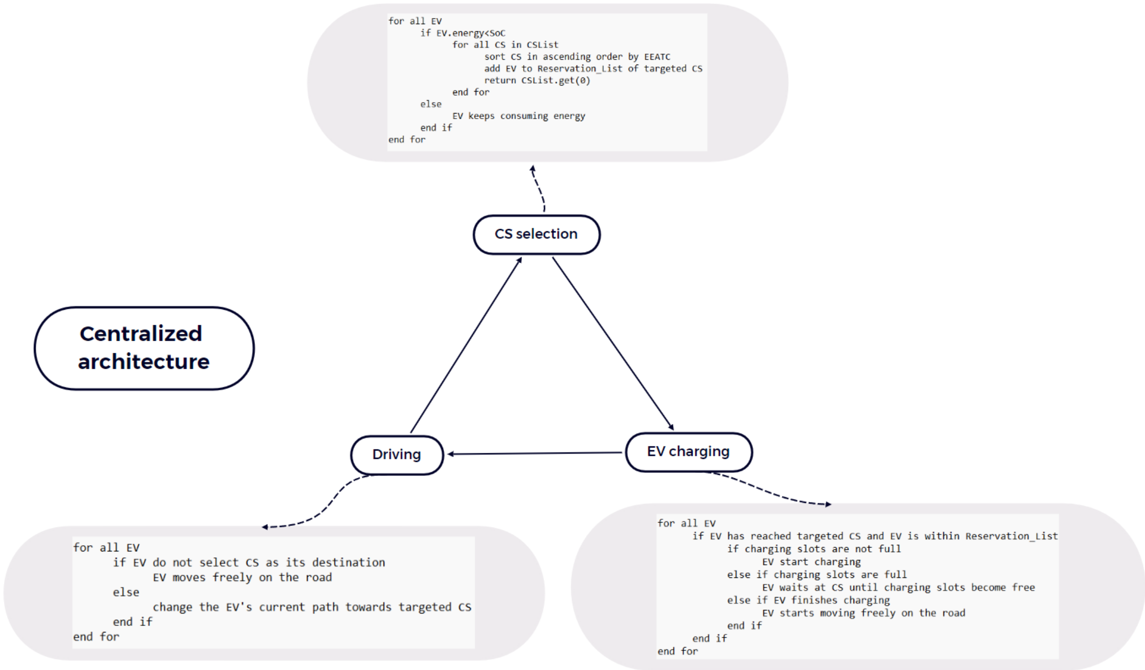

Figure 9 Centralized architecture and implementation

3.1.2 Distributed Architecture

In this scenario, GC acts as the cloud server, CS acts as the edge server. CS will broadcast the Earliest Available Time for Charging (EEATC) notified by GC last time after every interval δ, then EVs will make their own CS selection based on the broadcasted EEATC. Also, GC will compute EEATC for each CS based on the real-time information monitoring of CS. Last, GC will broadcast its computation result and notify CS after every β. Detailed implementation architecture is shown in **Figure 10**.

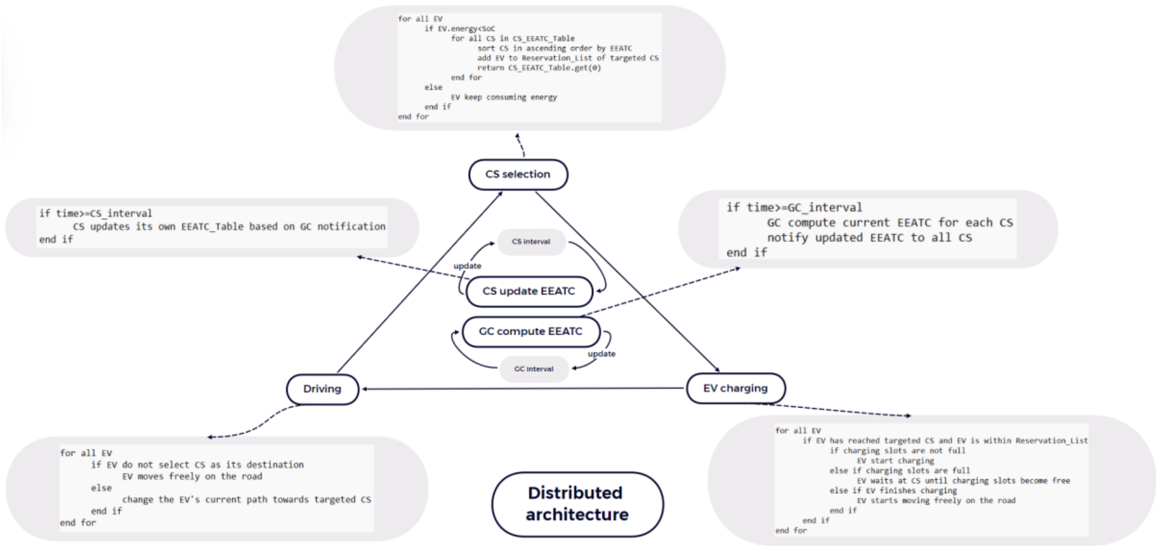

Figure 10 Distributed architecture and implementation

3.2 Result Evaluation

As shown in **Figure 11**, charging hotspots in CSs arise between the two schemes with the increase (from 130 to 190) of the number of the EVs, which makes the system busier. When the CS broadcasting interval is shortened from 1000s to 300s, only the charging performance of distributed architecture is influenced. A reduction in waiting time and an increase in the number of EVs being charged. The reason for that is the recorded information about EVs and CSs is more accurate, which leads to better decision-making.

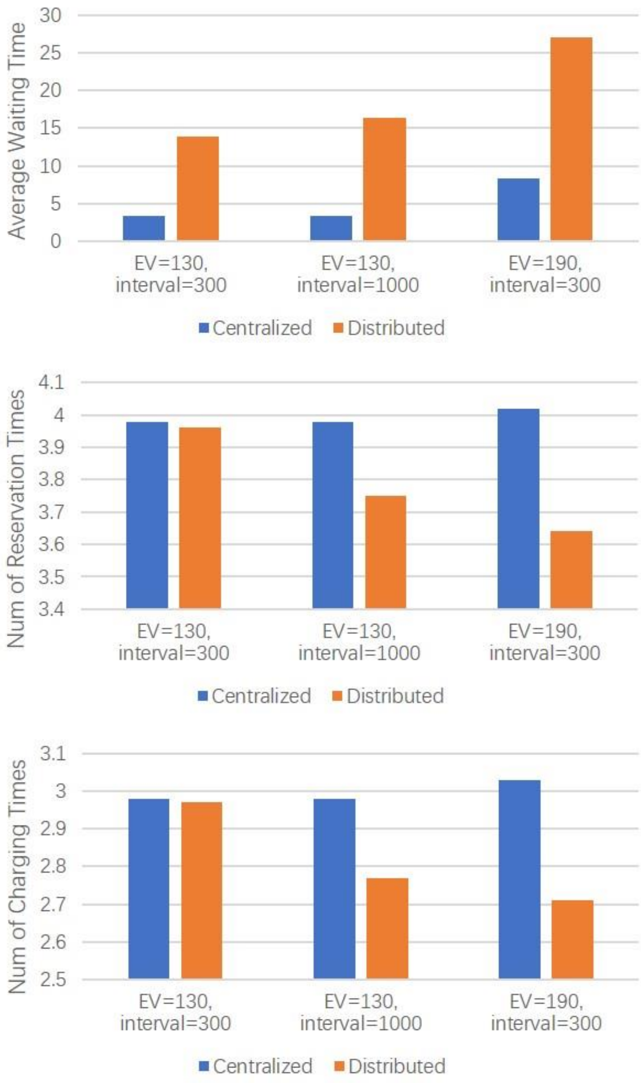

Figure 11 Simulation result

4. Conclusion

In this report, we first provided an overview of cloud computing, then focused on the trend drives from cloud to edge. The reason for that was also illustrated. Upon that, two well-known edge computing techniques Fog Computing and Mobile Edge

Computing were introduced and a detailed comparison was made between them. As vehicular energy network has become more popular these years, we addressed some application scenarios. With the implementation of MEC, delay and energy consumption of the entire network can be greatly reduced.

Then a detailed case study was made towards vehicular energy network, which includes its detailed design architecture, methodology, performance evaluation. A new architecture for EV charging called MEGEE was proposed, where introduced GC as the edge server to increase the performance over centralized and distributed framework. Furthermore, through our own simulation results, we show the implementation of centralized and distributed architecture and verify results from the case study.

On the other hand, we found the paper might lack some detail explanations when setting up the experimental environment. First, with consideration about the limitation of EV's transmission range, it seems not possible for every EV to receive CS's broadcasting message. Apparently, it would make great fluence towards distributed and MEGEE architecture, as the CS selection couldn't be done without the EEATC info broadcasting from CS. Detail about this process hadn't been discussed within the paper. Another problem is that, although the methodology about V2X techniques is addressed in detail, the paper didn't illustrate how V2X can be actually deployed. Due to the lack of some detail explanation, it was hard for us to implement the MEGEE architecture in the simulator.

5. Potential Research Opportunities

Federated Learning is a novel approach to MEC that has emerged due to the progress in personal technology and growing worries about privacy. As a result of this, a distributed vehicular network notion known as FVN (Federated Vehicular Network) has been created. This design differs from the conventional approach for it includes centralized features and employs both dedicated short-range communications (DSRC) and millimeter-wave communication. This exceptional design enables it to offer more reliable performance, resulting in many new potential applications.

There are still some potential research opportunities about the implementation of federated learning in vehicular energy network. For example, under different scenarios (parking lot, workload, users, number of vehicles), how to choose the optimal routing algorithms can be further discovered. Also, how to utilize federated learning better by adaption towards machine learning algorithms is another opportunity[12].

6. Reference

[1]. A. Ghorab, M. Abuibaid, and M. St-Hilaire, “SDN-based Service Discovery and Assignment Framework to Preserve Service Availability in Telco-based Multi-Access Edge Computing,” *IEEE 6th International Conference on Fog and Edge Computing (ICFEC)*, 2022.

[2]. P. Srivastava and R. Khan, “A Review Paper on Cloud Computing,” *International Journals of Advanced Research in Computer Science and Software Engineering ISSN: 2277-128X (Volume-8, Issue-6)*, 2018.

[3]. S. Yi, C. Li and Q. Li, “A Survey of Fog Computing: Concepts, Applications and Issues”, in *Mobidata’15: Proceedings of the 2015 Workshop on mobile Big Data*. pp. 37-42, 2015.

[4]. K. Dolui and S. K. Datta, “Comparison of Edge Computing Implementations: Fog Computing, Cloudlet and Mobile Edge Computing,” *Global Internet of Things summit (GIoTS)*, 2017.

[5]. Y. Mao, C. You, J. Zhang, K. Huang and K. B. Lataief, “A Survey on Mobile Edge Computing: The Communication Perspective,” *IEEE Commun. Surveys Tuts., vol. 19, no. 4, pp. 2322-2358, 4th Quart*, 2017.

[6]. Z. Ning, J. Huang, X. Wang, J. Rodrigues and L. Guo, “Mobile Edge Computing-Enabled Internet of Vehicles: Towards Energy-Efficient Scheduling,” *IEEE Wireless Communications., vol. 33, issue 5, pp. 198-205*, 2019.

[7]. K. Zhang et al., “Mobile-Edge Computing for Vehicular Networks: A Promising Network Paradigm with Predictive Offloading,” *IEEE Vehicular Technology., vol. 12, no. 2, pp. 36-44*, 2017.

[8]. Y. Cao, C. Wu, X. Zhang, W. Liu, L. Peng and M. Khalid, “MEGEE: Mobile Edge computer Geared v2x for E-mobility Ecosystem,” *IEEE Wireless Communications and Networking Conference (WCNC)*, 2019.

[9]. Y. Cao and N. Wang, “Toward Efficient Electric-vehicle Charging Using VANET-Based Information Dissemination,” *IEEE Transactions on Vehicular Technology, vol. 66, No. 4.* 2017.

[10]. J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, “A close examination of performance and power characteristics of 4G LTE networks,” in *Proceedings of the 10th international conference on Mobile systems, applications, and services.* ACM, pp. 225–238, 2012.

[11]. Y. Cao, Z. Sun, N. Wang, H. Cruickshank, and N. Ahmad, “A Reliable and Efficient Geographic Routing Scheme for Delay/Disruption Tolerant Networks,” *IEEE Wireless communications Letters,* col. 2, no. 6, pp. 603-606, December, 2013.

[12]. J. Posner, M. Aloqaily and Y. Jaraweh, “Federated Learning in Vehicular Networks: Opportunities and Solutions,” *IEEE Network, vol. 35, pp. 152-159,* 2021.