
FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING
DEGREE PROGRAMME IN ELECTRONICS AND COMMUNICATIONS ENGINEERING

# MASTER'S THESIS

## Performance and Efficiency Optimization of Multi-layer IoT Edge Architecture

| | |
|---|---|
| Author | Muneeb Ejaz |
| Supervisor | Dr. Erkki Harjula |
| Second Examiner | Prof. Mika Ylianttila |
| (Technical Advisor) | Tanesh Kumar |

April 2020

**Ejaz M. (2019) Performance and Resource-Efficiency Evaluation Framework for IoT Edge Computing.** University of Oulu, Faculty of Information Technology and Electrical Engineering, Degree Programme in Electronics and Communications Engineering, 62 p.

# ABSTRACT

**Internet of Things (IoT) has become a backbone technology that connects together various devices with diverse capabilities. It is a technology, which enables ubiquitously available digital services for end-users. IoT applications for mission-critical scenarios need strict performance indicators such as of latency, scalability, security and privacy. To fulfil these requirements, IoT also requires support from relevant enabling technologies, such as cloud, edge, virtualization and fifth generation mobile communication (5G) technologies. For Latency-critical applications and services, long routes between the traditional cloud server and end-devices (sensors /actuators) is not a feasible approach for computing at these data centres, although these traditional cloud provide very high computational and storage for current IoT system. MEC model can be used to overcome this challenge, which brings the CC computational capacity within or next on the access network base stations.**

**However, the capacity to perform the most critical processes at the local network layer is often necessary to cope with the access network issues. Therefore, this thesis compares the two existing IoT models such as traditional cloud-IoT model, a MEC-based edge-cloud-IoT model, with proposed local edge-cloud-IoT model with respect to their performance and efficiency, using iFogSim simulator. The results consolidate our research team's previous findings that utilizing the three-tier edge-IoT architecture, capable of optimally utilizing the computational capacity of each of the three tiers, is an effective measure to reduce energy consumption, improve end-to-end latency and minimize operational costs in latency-critical It applications.**

**Key words: Traditional Cloud, Edge, Mist, IoT, MEC.**

# TABLE OF CONTENTS

# FOREWORD

This research thesis was carried out at one of the most professional and top researcher group Center for Wireless Communication-Network and System (CWC-NS) at the University of Oulu Finland, under the Center for Wireless Communication department. It was supported and funded by the Industrial Edge (Academy of Finland) and MEC-AI (Technology Industries of Finland Centennial Foundation, and Jane and Aatos Erkko Foundation) projects. I also want to acknowledge and thank my supervisor Dr. Erkki Harjula and second examiner Prof. Mika Ylianttila in particular for their valuable insights, continuous encouragement and guidance to complete the research Master's thesis in the NSOFT progress report meetings. Furthermore, I am happy to work under professional mentor and technical supervisor help me write this thesis, Tanesh Kumar who is a PhD student at the University of Oulu research group CWC-NS.

I am honoured and thankful to both the Center for Wireless Communication (CWC) and the University of Oulu for allowing to learn in a professional research environment and enhance my skills. I am also grateful to all the members of my family for their motivation and support especially my wife Iqra Muneeb, and my father Ejaz Ahmad. Furthermore, I want to appreciate my close friends especially Asad Ullah Khan, Hassan Malik and Abdullah Malik who always there to guide and help me and provide their lovely company throughout my master's. In the last, I would like to thank to Finnish education authority for encouraging me to obtain my Master's degree.

Oulu, 25th April, 2020

Muneeb Ejaz

# LIST OF ABBREVIATIONS AND SYMBOLS

| | |
|---|---|
| 3D | Three Dimension |
| 5G | Fifth Generation mobile telecommunication system |
| AI | Artificial Intelligence |
| AMD | Advance Micro Devices (a semiconductor manufacturer) |
| API | Application programming Interface |
| BRITE | Boston University Representative Internet Topology Generator |
| CC | Cloud Computing |
| CCTV | Closed-circuit television |
| CPU | Central Processing Unit |
| CRM | Customer relationship Manager |
| DDF | Distributed data flow |
| DDoS | Distributed Denial of Services |
| DNS | Domain Name Service |
| EAaaS | Edge analytics as a service |
| EC | Edge Computing |
| ECC | Edge Computing Consortium |
| ETSI | European telecommunication Standard Institute |
| FC | Fog Computing |
| FPRBAC | Fog-based Privacy-aware Role Based Access Control |
| GUI | Graphical User Interface |
| IaaS | Infrastructure as a Service |
| IBM | International Business Machines (an ICT device manufacturer) |
| IBSG | Internet Business Solutions Group |
| ICT | Information and communication technologies |
| IEEE | Institute of Electrical and Electronics Engineers |
| IIoT | Industrial IoT |
| IoE / IoX | Internet of Everything |
| IoT | Internet of Things |
| IoV | Internet of Vehicle |
| IP | Internet Protocol |
| ISM | Industrial, Scientific and Medical |
| ITU | International and Telecommunication Union |
| M2M | Machine-to-Machine |
| MACC | Mobile Ad-Hoc Cloud Computing |
| MANET | Mobile Ad-Hoc Network |
| MC | Mist Computing |
| MCC | Mobile Cloud Computing |
| MEC | Mobile Edge Computing, Multi-access Edge Cloud |
| MI | Million Instructions |
| MIT | Massachusetts Institute of Technology |
| NAT | Network Addressing Translation |
| NIST | National Institute of Standards and Technology |
| NS-2 | Network Simulator 2 |
| NV | Network Virtualization |
| OS | Operating System |

| | |
|---|---|
| PaaS | Platform as a Service |
| PAN | Personal Area Network |
| PC | Personal Computer |
| PNNL | Pacific Northwest National Laboratory |
| PoC | Proof of Concept |
| QoE | Quality of Experience |
| QoS | Quality of Service |
| RAM | Random access memory |
| RFID | Radio-frequency identification |
| RSU | Road Side Units |
| SaaS | Software as a Service |
| SAP | Systems, Applications & Products Company |
| SDK | Software Development Kit |
| SOA | Service-oriented Architecture |
| TV | Television |
| V2I | Vehicle to Internet |
| V2V | Vehicle to Vehicle |
| VANET | Vehicular ad hoc network |
| VLANS | Virtual Local Area Networks System V |
| VM | Virtual Machine |
| VMM | Virtual Machine Manager |
| WAN | Wide Area Network |
| WiFi | Wireless Fidelity |
| WLAN | Wireless Local Area Network |
| WSN | Wireless Sensor Network |

| | |
|---|---|
| $E_T$ | Total Energy Consumption |
| $E_C$ | Energy consumption at core layer |
| $E_E$ | Energy consumption at edge layer |
| $E_L$ | Energy consumption at local layer |

# 1 INTRODUCTION

## 1.1 Motivation

In this digital world, Internet of Thing (IoT) is gaining immense attention due to its frequent use in almost each aspect of our daily life. In wireless communication, modern advance developed technologies also support and secure the concept of enabled services globally [1][7]. Modern IoT applications enabled by such emerging developments that require highly critical and demanding network requirements. Automated delay sensitive Industrial IoT (IIoT), for example, establishes stringent criteria to execute various processes safely and on the provided time for underlying network system [13].

Centralized cloud servers are considered an ideal location to process most of the IoT devices, end-devices data sent to the high computing and efficient processors in the cloud [39][40]. According to cisco in [2], the number of end-devices will reach 50 billion by 2020 and can generate 80% of the data traffic over internet open critical challenges to a traditional cloud-only-approach and it is not suitable for mass-critical applications and services [6][7][76]. That is for three main reasons. First, large distance between the end-devices and the centralized cloud for processing produce high end-to-end latency and bandwidth. Second, the openness to security threats is high as the raw data propagates far away from its source. Third, underlying networks required high capacity i.e. data delivering to and from center to internet backbone is always higher than the pace of the internet backbone.

With edge computing paradigm, the end devices sensitive data is processed to the edge of the network and close to the end-devices. It is a novel computing layer between centralized cloud and end-devices to meet these challenges in the low latency, high performance [60][70].

In mass-critical IoT applications, Mobile Edge Computing (MEC) available at mobile access network would add value to the overall network. Mobile Edge Computing (MEC) servers are available at mobile access network will provide benefit to the overall network in the context of high demanding IoT applications. However, the current MEC model has some limitations.

However, there are still some limitations with the current MEC model. In smart latency sensitive applications, continuous communication is in demand for critical process execution and also avoid the flow of the sensitive data to unauthorized servers/devices [40][64]. Therefore, it is important to investigate the possibility of utilizing local available resources to handle the most crucial data analysis and decision-making to endure a reliable operation across access-network connectivity problems. To handle the data-intensive and real-time IoT applications, compared to the traditional cloud, several various alternatives of the network architecture is required.

With the emergence of data-intensive and real time IoT applications, there is a clear need of various alternatives of the network architectures besides the traditional IoT centralized cloud. Hence, for delay-intolerant decision, it is significant to process and storage at the edge layer. Furthermore, local layer is not capable since IoT devices are capacity constraint than edge devices, so it is unfeasible to process all the data locally.

Therefore, in this context, processing and storage at the IoT edge and local/device layer is crucial for delay-intolerant decisions. It is also important to notice that devices or hardware resources at the local layer are not as capable as at the Edge, so it is therefore not feasible to bring all the functionalities to the local level. The scientific group has therefore considered the best and efficient approaches at the edge and local layer.

## 1.2 Aim of the Work

The main objective of this thesis is to study the performance and efficiency of IoT-related communication and computation on a theoretical level with varying numbers of end-nodes, real time-application scenarios, and different deployment options. The results of the thesis have also been published in the 6G summit paper [110]. The deployment types range from fully centralized cloud-based operation, through access network edge-based operation (MEC) to fully decentralized local edge-based operation.

The goal is to design a simulation model on a selected available simulators which, can handle CC , EC and Local edge computing paradigm in order to compare and evaluate the performance and efficiency between three-tier IoT models. Following are the performance indicators in this thesis:

- End-to-end latency
- Network Load,
- Power Consumption

The aim of the thesis shown in the Figure 1. In order to reduce the network workload and increase the overall efficiency of current available IoT models such as CC and Edge paradigm, Local edge IoT is an emerging paradigm that is required to overcome this problem. Furthermore, it helps to secure the user private data more securely and reduce the communication path failure between application server and end devices than CC and EC paradigm.

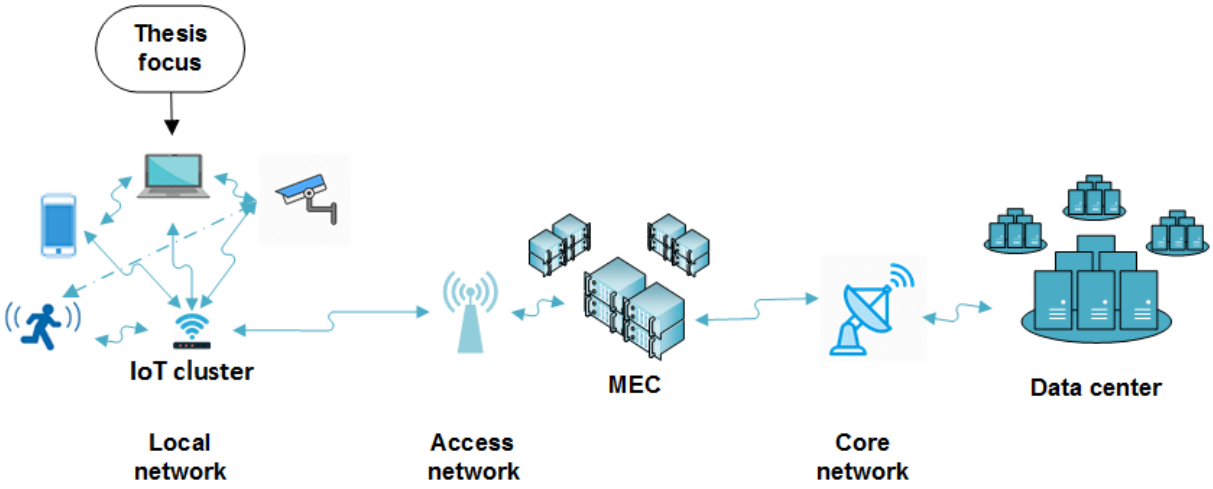


Figure 1. Emerging IoT network architecture [39].

## 1.3 Thesis Structure

The structure of the thesis is organized as follows.

In Section 2, background knowledge is presented in order to understand the full extent of the stated solution. This section first explains the fine grounds of IoT technology and its challenges. Furthermore, emerging IoT-models are reviewed in details such as Fog, Edge and Local Edge computing basics, applications, benefits, architecture, and challenges are discussed to give the reader a background about the technologies used. Section 3, covers the existing IoT models that

are designed for different less critical and critical-latency applications requirement. In the end, purposed model is designed using IIoT use-case scenario to overcome the overall performance of the network and evaluated various performance matrices. Section 4, covers the available simulation tools and their limitations used for different use-case IoT scenarios in CC, EC and LEC paradigm. Further, the most suitable simulation tool is selected after reviewing research views available research for our proposed model. The performance evaluation of traditional cloud-IoT model, a MEC-based edge-cloud-IoT model, and a local edge-cloud-IoT model with respect to their performance and efficiency is tested and results are displayed and discussed. Section 5, discussion and potential future work are highlighted. In the end, Section 6 concludes the thesis with a summary of the work done.

# 2 RELATED WORK

This section covers state of the art technologies definitions and theoretical background to provide a clear picture for the reader. Furthermore, this thesis describes existing technologies used in IoT applications, current IoT models, benefits and their challenges.

## 2.1 Internet of Things (IoT)

### *2.1.1 Background*

Kevin Ashton introduced the term Internet of Things (IoT), first time in 1999, who is the founder of MIT auto identification centre. According to Ashton "Internet of Things" brings new paradigm and it will change the world just as internet did. Ashton introduced a system in which, world devices or physical objects can be connected to the internet via sensors. He demonstrated the connectivity of the objects with radio-frequency identification (RFID) technology in Massachusetts Institute of Technology (MIT laboratory) [1][2].

The smart things in the IoT network has ability to gather distribute and analyze the raw data into useful information. The internet was a hottest trend at that moment and IoT capabilities opened a new foundation for various key applications [1]. Since then, the evolution of IoT systems have been seen quite rapidly, especially since last decade. Along with the increase in the growth of the IoT applications, the number of devices are expected to reach to 50 billion in a year 2020 reported by Cisco IBSG [2] shown in the Figure 2.

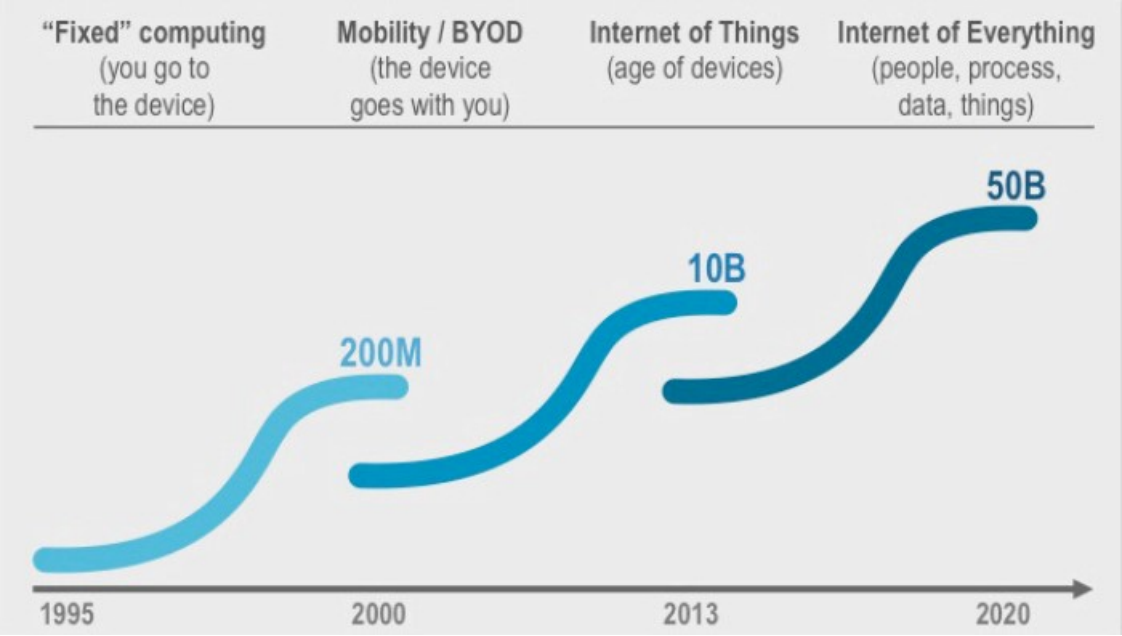


Figure 2. Continuous growth of internet connected devices [2].

IoT paradigm provides low-cost implementation of smart devices and the rise in demand had a huge impact on consumer's live and business models. IoT has become a significant business area, where traditional sensor/actuator technologies are integrated with various other enabling technologies such cloud/edge/mist, virtualization, block-chain etc to fulfil high demanding applications. The IoT is considered as vital existence in the world, comprising a number of

things, which can be linked via wireless and wired connections. Such devices have a unique devices approach that enables things to interact and cooperate with each other to generate new IoT applications and services such as smart hospitals, e-health, smart transport system, smart cities and traffic management etc. [3].

### *2.1.2 Definition and related terms*

The IoT is a novel paradigm that increases rapidly by gaining attention in modern wireless telecommunication scenarios in which , users can control and monitor the physical devices via link through internet to provide interaction and communication with each other.

According to International and Telecommunication Union (ITU), IoT is a universal architecture for digital society that enables distributed networks through the interconnection of things based on existing and evolving information and communication technologies (ICT) [5].

Currently, researchers, scientists and authorities are paying more attention on IoT as it is considered to be the next stage of advancement. Though, internet was developed in 1980s after gone through several stages it has evolved from some computers that communicate with each other to billions of computing devices and billions of phones over time [2][3][4].

The IoT can viewed as both flexible and globally network architecture that handle things in an intelligent manner. As a result, their information is shared because of the interconnection of IoT devices to create new applications and services that can improve human life and wold's economy.

### *2.1.3 IoT architecture*

The general IoT network architecture is consist of three layers: perception layer, network layer and application layer as shown in the Figure 3.

- **Perception Layer**: Perception layer is also called sensing layer. End devices such as sensors, smart phone etc. belong to this layer which sense and collect the data from the environment (such as pressure, humidity, temperature) with the help of sensors and actuators before transmitting to the network layer [6][7].
- **Network layer**: It helps in providing functions of data routing and transmission to the final destination. It is the middle layer of IoT architecture. Devices which operate at this layer are routers, switches etc. [7].
- **Application Layer:** It implements various services and applications on the behalf of the received data and information from the lowest layer. It is the top-layer of the IoT architecture. It comprises user interface, data models algorithm and everything, which is required for IoT service and applications [5][7].

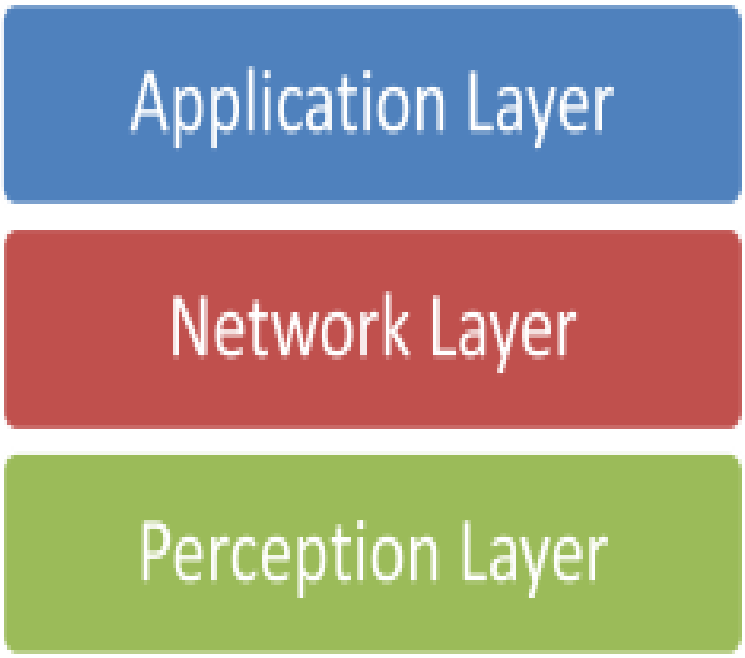


Figure 3. Three layer IoT architecture model [7].

### *2.1.4 IoT challenges*

IoT offers benefits in terms of economy and human comfort. However, because of low computational capacity and energy resources IoT devices, there are some important challenges that required to be highlighted by researchers and scientist until the IoT concept is widely acknowledged [12].

- **Big Data:** As mentioned above, the number of IoT devices, will reach 50 billion in year 2020 due to which a huge amount of raw data will be [12]. As the data arrive in real time and sometime variable, so analytics and storing these raw data according to the volume, speed and dynamic make it complex. Cloud computing provide resources to process and store the data for long terms. There are some limitations in order to handle this large amount of data in CC. Therefore, IoT applications performance totally depends on the data management algorithms and services. IoT is main source of big data, so it requires data integrity in order to provide good quality of service and privacy issues [13][14][15].

- **Networking:** Devices involves in the same IoT platform use different protocols for communicating between other devices or in the networking to maximize the network rate. Intelligent network protocols should follow the protocols for communication, already developed in machine-to-machine (M2M). It is not an easy way to develop a new protocol for networking to fulfil the needs such as overall system performance, cost, Quality of Service (QoS) and ease-of-use [15][16][19]. IoT devices creates a significant challenge to design a suitable network topology.

- **Heterogeneity:** To provide new applications that make our life easy, the IoT links massive number of objects/things/devices. Heterogeneity of devices, frameworks, operating systems (OS) and services that are already developed and might develop new applications is one of the main problems faced by IoT systems [17]. In order to handle distributed network and IoT applications, efficient services are needed to overcome this issue. Furthermore, IoT system devices from different manufacturers make connectivity and processing a very complex and difficult task [18].
- **Security and privacy:** Security and privacy are among the most challenging issues confronting most of the new technologies. IoT devices such as sensors installed in the surrounding environment to gather the data. This sensitive data, such as financial records, habits, human vitals, etc. helps researchers, business and e-health as well on which the decision takes into account for further process. The development of a stable and more secure IoT infrastructure is a necessary duty for continuing its effective implementation in our environment [22]. IoT devices in IoT networks are mostly linked to wireless networks, thus it is very difficult to protect against several attacks, such as man-in-the-middle, data sniffing, etc. The sheer volume and complexity of these devices raise the potential attack area for the hacker. Gartner predicts that more than 25 percent of all client attackers will make use of IoT by 2020 [20][21]. To overcome this issue, powerful security algorithms are needed.
- **Maintenance:** As the number of IoT devices grows and has reached almost 50 billion in 2020, so it is a challenging problem to maintain these devices, which connect to the internet. Most of the IoT devices belong to a different manufacturer who do not care about new security, privacy platforms, upgrades and other problems on regular a basis in their devices. Such IoT devices allow the hackers to use it as a weak point and effect the whole IoT network and overall system performance [17].

## 2.2 Cloud Computing

### *2.2.1 What is cloud computing*

The Cloud Computing concept has matured over the last few years. Cloud computing is used as a platform for IoT applications to process, compute, store and make decisions. The concept means that anything that can be hosted on the Internet, i.e., resources/services/data is available for use, when needed, for the composition and provision of more sophisticated services. National Institute of Standards and Technology (NIST) defines cloud computing referees, on-demand access to the network to a shared pool of configurable resources (servers, resources, software, data, etc.) that can be rapidly distributed and released with minimal management or service provider interaction [23][28].

Cloud data centers provide data management services by offering high storage and computational resources, global availability and high scalability. The cloud services can be accessed by the user from anywhere in the world and with any device, which is connected to the internet, which shows it is location-independent [31]. Furthermore, binding these raw data is a more complex process due to varies operating systems, connectivity protocols, and legacy applications compatible. IoT platform requires computing and storage, which it can get in the form of shared resources from the traditional cloud. Some of the widely known examples of cloud service providers are IBM, Google, Microsoft, and Amazon to host these cloud-based

resources [24]. Cloud computing also offers a multi-tenancy feature that enables the sharing of resources to various users over time and spatial distribution. Furthermore, it offers scalability, which provides a huge benefit for Cloud and IoT convergence.

Cloud computing is used as a platform to facilitate the IoT applications to process, compute, store and take the required decision. Cloud data centers provide data management services by offering large storage resources, computational capabilities, better security, and privacy. The main cloud features include on-demand service provisioning, resource pooling, and global access and so on. The key purpose of IoT is to connect objects, devices, and humans that generate a huge amount of raw data. Cloud computing also offers a multi-tenancy feature that enables the sharing of resources to various involved entities over time and spatial distribution. Furthermore, it offers scalability that provides huge benefits for Cloud and IoT convergence [26].

### *2.2.2 Virtualization*

Sharing extensive equipment infrastructure ideas across various application environments was introduced in 1981 by IBM. According to IBM, the virtual machine is a copy of the physical hardware of machine-like memory, storage, processors, etc. The physical resources can be shared among the virtual machine with virtualization. Goldberg and Popek introduced three requirements that satisfy virtualization to achieve efficient virtualization [28][29]. Figure 4 describes the host virtualization architecture.

- Virtualization compared to the local machine, should provide the same environment to run the program.
- It should help to control the virtual resources to protect the sensitive data in the virtual environment.
- With additional tasks of virtualization cause performance degradation but good results in managing privileged instructions should be accomplished with software or hardware support.

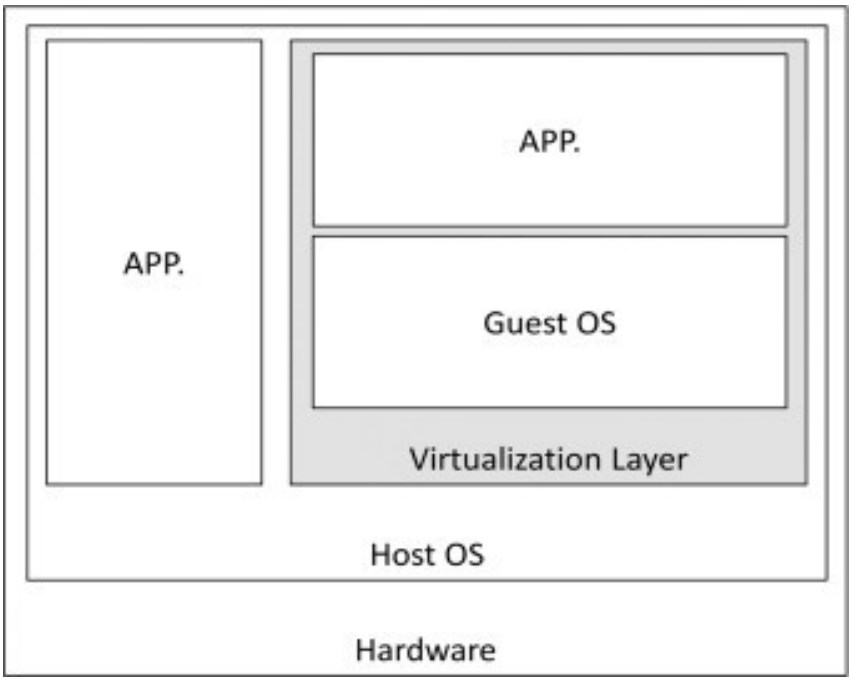


Figure 4. Virtualization Architecture [28].

Virtualization aims at optimizing devices usage, lower hardware cost through consolidating several virtualized computers into one physical unit, reducing energy use and simplifying protection and network management. To virtualize the guest Operating System (OS), five techniques can be used [28].

- **Full Virtualization**: In full virtualization environment, the guest operation system runs on the virtual machine while the host machine runs directly on hardware the guest OS does not need to know the presence of a hypervisor in full virtualization. Guest OS that belongs to its virtual machine operated independently. Furthermore, using direct execution and binary translation, multiple guest operating system run in an isolated way on a single host operating system [27][28]. In full virtualization, system degrades as the translation between physical and virtual resources is so continues, which can lower the system performance. Full virtualization is shown in the Figure 5.

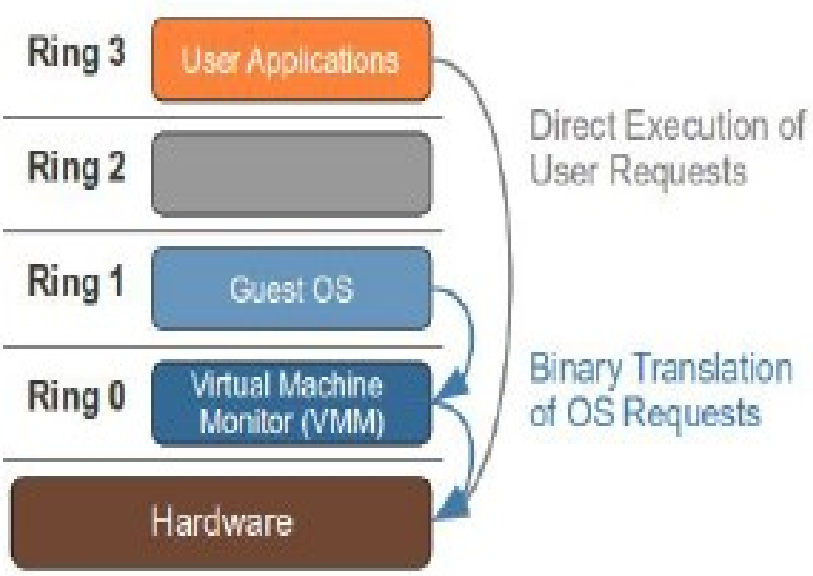


Figure 5. Full virtualization [27].

- **Paravirtualization:** In paravirtualization, the guest host can communicate with the hypervisor. Instead of the hypervisor in the host machine, it is installed in the guest host to achieve better system performance than full and hardware-assisted virtualization. It helps to modify the guest operating system using hypervisor API calls.On paravirtualization, performance operations by operating system reduced the execution time. Figure 6 illustrated paravirtualization [28][29][30].

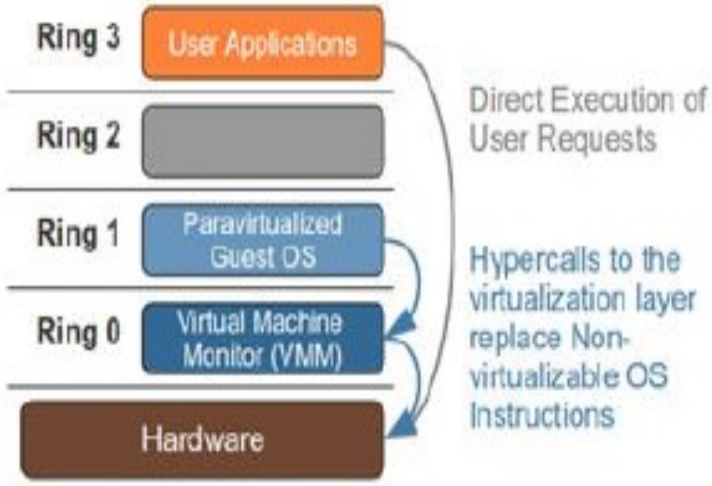


Figure 6. Para virtualization Adopted [27].

- **Hardware-assisted virtualization:** Hardware-assisted virtualization has been described in Figure 7. To achieve better results, Intel and advance micro devices (AMD) introduced hardware-assisted virtualization technology. Guest OS runs at Ring 0 and the hypervisor runs at Ring 1. So, Para virtualization is nor more required, so Virtual machine Manager (VMM) does fewer operations and the system performance increased [26][27][28].

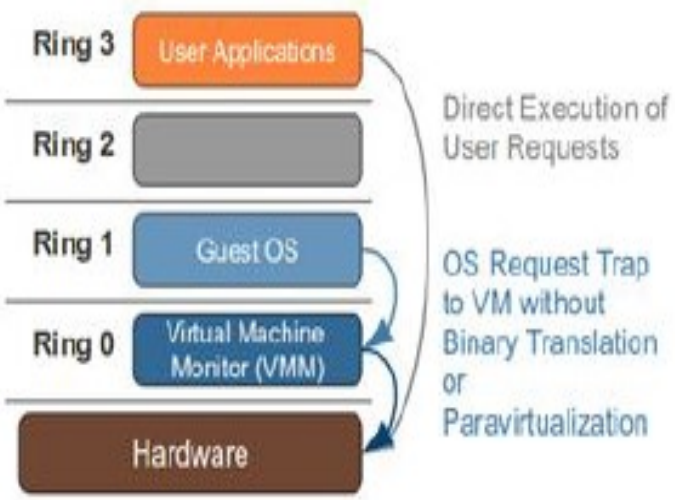


Figure 7. Hardware-assisted virtualization [27].

- **Network Virtualization (NV):** Network virtualization separates the network from underlying network hardware. It combines the available physical network resources or part of network resources into one virtual unit. Network virtualization components offer routing and network addressing translation (NAT) by network media as Ethernet and fibre channel network elements such as laptop, personal computer(PCs), virtual local area network system (VLANS), network hardware, routers, switches [31][32].

- **Server virtualization:** It allows dividing a physical server into multiple virtual servers that can be installed on random hardware. These are also called private servers or virtual servers. In server virtualization, each virtual machine runs independently. Full virtualization, paravirtualization, are common approaches of server virtualization [33].

### *2.2.3 Cloud Service Models*

Cloud offers there key service models, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) that are discussed in this section.

#### *2.2.3.1 Software-as-a-Service*

The First business model used in cloud computing is the most limited option called SaaS. This approach allows the customer to use the applications running on the cloud infrastructure. Users can access these applications only via web browsers like Google Chrome or Internet Explorer or programmable Graphical User Interface (GUI). User does not have to think about how the service is controlled or how the underlying network is maintained using the SaaS model. An example of the SaaS platforms is Google Docs, Office 365, Zoho, Oracle Customer relationship manager (CRM), Adobe Creative Cloud and web-based email. End-users cannot manage the cloud resources of the service by themselves [34][35].

#### *2.2.3.2 Platform-as-a-Service*

Cloud platform services are also called as PaaS, where developers are considered as customers, allowed to develop, test, and run applications in the cloud environment. The user can manage the deployed applications but the underlying network such as servers; the vendor or third party manages VM's, storage, and operating system. OpenShift, System Application and products (SAP), Force, Mosso, Google App Engine and Window Azure are examples of PaaS model. PaaS model is identical to SaaS, except that PaaS offers a forum for application development instead of distributing the software via internet like SaaS. User does not need to have expensive machines to run their applications on it. PaaS offers a great deal of scalability by design as it is built on cloud computing. PaaS allows Google is using PaaS model to provide the platform for them [34][36][37].

#### *2.2.3.3 Infrastructure-as-a-Service*

Infrastructure-as-a-Service (IaaS) are made of extremely distributed and digital computing capabilities. Instead of buying hardware, IaaS helps service providers to purchase on-demand, as needed services. Configuring Virtual Machine (VM) running on the cloud is allowed to the clients. The VM resources such as storage, RAM, operating system and Central processing Unit (CPU) speed that can be managed by users only. It is also called self-service model. The customer (service providers) can run number of applications on the given VM but cannot have full control over physical resources in the cloud. Vendors are responsible for providing security, firewalls and physical data center. Examples of IaaS are Microsoft Azure, Amazon E2C, GoGrid, Rackspace etc. Figure 8 shows diagram of three cloud computing service models for further illustration [37][38].

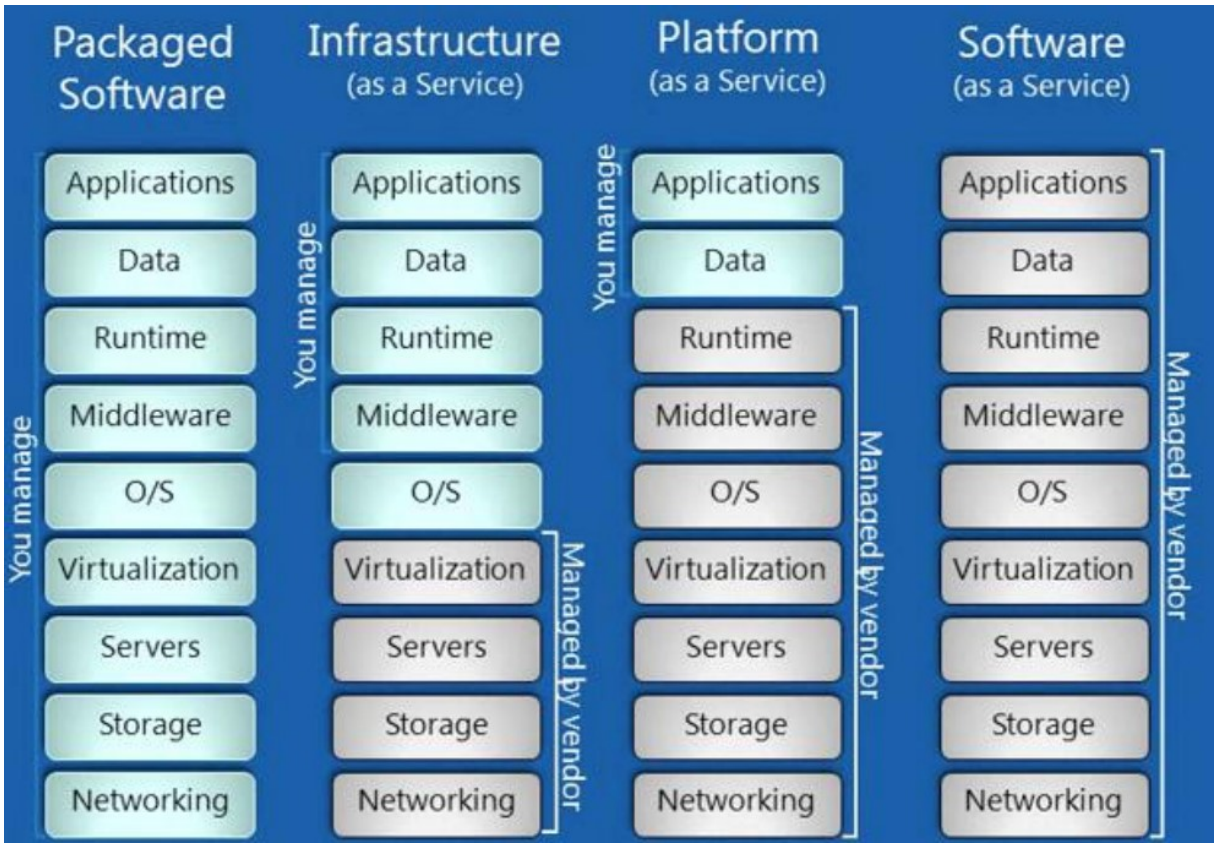


Figure 8. Cloud Computing Service Models [34].

#### *2.2.3.4 Microservices*

In the last few years, microservices have achieved significant attention over the industry. The cloud services have evolved from traditional to microservices architecture in recent years, a number of microservices form a service, which can execute limited tasks is considered to be a refined and simplified architecture [40]. With an only centralized system, one can achieve many benefits by having multiple individual services that work together in concert. It gives benefits by including simplified codebases for specific services, adopted scalability and allows upgradation, able to code in multiple programming languages if required, and data layers for various services as well. Microservices are adopted by some companies like Amazon, Sound Cloud, and Netflix to create a complicated, a scalable system comprising tiny autonomous services that use application programming interface (APIs) for communicating with each other to scale their applications and product [39].

Microservices have various benefits over monolithic applications architecture such as: 1) it reduces the complexity by introducing individual tiny services; 2) easy to deploy in the system and removes from the system; 3) increase the system flexibility; 4) scalability. Instead of attempting to make feature calls in a process, that method often include cost like operational and computational complexity of operating an application in different processes and network connectivity cost [41].

The best approach for building microservices can be achieved through Docker containers. They are easy to use due to lightweight, start quickly and can cover within itself dependencies and introduce complexities. A developer, for example, can launch hundreds of containers on a

reasonable laptop. An essential feature of Docker containers from IoT perspective is that they can be distributed where only one or few processors run within a single container [39]. Docker Swarm, Kubernetes, and Mesos are commonly are widely deployed container orchestration systems and offer automatic support like load balancing, software upgrade, and service discovery. Using microservices within IoT domain provides a promising approach to accommodate IoT functions locally by keeping it small enough. Butzin et al. [40] examined this concept, concluded that microservice solution and IoT have the same framework target, and would thus be a good combination [40].

### *2.2.4 Deployment Approaches*

Earlier the availability of various cloud computing services models is described in the previous subsection. In this section, deployment models are enlisted on which cloud computing service models are hosted.

#### *2.2.4.1 Public Cloud*

The services in the public cloud can be accessed publicly as it is not restricted to any organization or community. Consumers do not have to bear the expenses for the maintenance of this cloud, as it is free to use the applications provided by the service providers in it. Service providers have to pay the cost of the infrastructure and bandwidth deployed in it. It is not suitable for the organizations operating sensitive information exposed to the public as the security rules are not applied according to the organization [37][38].

#### *2.2.4.2 Private Cloud*

A private cloud is used as a dedicated cloud to help small or large organizations use it for confidential business data. The infrastructure of the private cloud is operated and managed by the organisations, institutions and the government itself. A Private cloud provides more control over on-demand scalability, flexibility, and on-demand security as well. The sensitive information will remain in the private cloud, which is less exposed to the other environment-related organizations, which is one of the most beneficial parts [37].

#### *2.2.4.3 Hybrid Cloud*

According to the name, hybrid means it is a combination of both public and private clouds. It offers some cloud resources that can be owned controlled and managed which can be operated in the private cloud while others are used via the open accessible public cloud. For Example, a private cloud is used to secure the organization’s sensitive information while the public cloud is used for generally less critical information [24][35][38].

### *2.2.5 Challenges in Cloud Computing*

Cloud computing paradigm allows the users/customers to access the Ubiquitous Computing Resource pool. Although, cloud computing has various benefits, on the other hand, it has some limitations such as security, privacy, latency, and network load as well.

- **Privacy:** Privacy is going to be more of a major issue with centralized cloud computing paradigm; we will have in the future. Data privacy and security will be a problem on the cloud-server side. The traditional cloud exposed to a huge number of potential hostile users, resulting in concerns of user data privacy. The main issue while using the cloud resources, the data is moved from the local device through numerous network hops and finally reached the cloud allows the hackers to view the sensitive data easily. According to The Independent, stating United States of America (USA) authorities spied the sensitive data in British internet users on various major cloud storage services regularly on Feb 01, 2013 [42][43].

- **Latency:** The delay increase as the physical distance between both the origin point as well as the endpoint increases. Number of routing hops has bigger effect, as routers generate delay, particularly under congestion. The time required to access the traditional cloud-based application is too high according to the geographical distance between the user and the cloud. Now, a day's developers are developing applications, which require low latency, or latency sensitive application. Therefore, the cloud is not practical for low-latency application or latency sensitive applications such as smart transport, e-health application, etc. require high performance and high reliability, if any delay in the patient data occurs could result in patient's life. So, cloud computing is not suitable for such type of applications [43][44].

- **Network Load:** The number of end devices is increasing day by day which results in creating a huge amount of data at the edge of the network requires cloud resource it will cause network congestion at the edge of the network. It is impractical for the cloud to handle network load. A new platform solution is required, which is impractical for many use cases used as a cloud-only solution [42][43][44].

- **Security and Data Confidentiality:** With cloud computing, the user cannot have absolute control of their data when accessing the cloud. In addition, the sensitive data is stored in the cloud, data location or policy for handling data are changed without permission etc. The updated data can be then recovered and analyzed to critical decisions by the user. In this case, the authenticity of the user data is very important, and thus must be assured. Common standards, however, do not take place for guaranteeing data security. It is impossible for an end-user to even decide which authentication protocols and security mechanisms are applied to data in the cloud is a complex, non-transparent chain [44][45][46].

- **Resource Allocation:** As the number of connected devices (IoT devices) increases unexpectedly, cloud resource is required for each entity for computing the data. This

could be a challenge because it would be very difficult to decide how many resources may be a system, entity or IoT device needs [43].

- **Quality of Service (QoS):** With the cloud-computing paradigm, providing Quality of service (QoS) is a major challenge as the volume of information, type and complexity increases. Any type and amount of data can be induced at any given moment. This can also be emergency data. QoS calculated in terms of Latency, jitter, bandwidth [47][48]. According to [2], the number of IoT devices will reach 50 billion by 2020. Therefore, it requires processing and storage services which is not easy for the cloud to fulfil all IoT applications requirement for cloud computing architecture, due to distributed in vast geographical areas. To solve this problem, a new layer between end devices and centralized cloud is needed.

## 2.3 Overview of the Edge Paradigm

In this section, fog, edge, and local edge computing paradigm concepts are discussed theoretically along with definitions, architecture, benefits, and challenges.

### *2.3.1 Fog Computing*

Bonomi [44] proposed the idea of using the computational resources provided by the fog devices located on the edges of the network, a concept called Fog Computing (FC).

Currently, researchers are continuously investigating the process of using edge capabilities to support IoT needs in a better way. Cisco proposed a fog-computing framework, which pushes the centralized cloud services close to the edge devices, which generates the data such as sensors and actuators [45]. Fog computing supports the latency-sensitive applications. Fog computing is different from edge computing and provides tools for distributing, orchestrating, managing and securing resources and services across networks and between devices that reside at the edge. Edge architecture places servers, applications, and small clouds at the edge. Fog jointly works with the cloud [54].

Fog computing adds extra layer between end devices and centralized cloud and provides security and privacy for private data such as healthcare, vehicle communication, user location information [44][54]. Fog nodes consist of network devices that perform computational tasks and data storage capabilities in the same way as a centralized cloud.

Fog computing is emerging as attractive solutions to the problem of data processing in the IoT. Many IoT use-cases e.g healthcare, video browsing requires low latency and real-time decisions, it is not possible to get such output using a traditional cloud platform [55]. Rather than outsourcing all operations to the traditional servers, they also use devices on the edge of the network that has more processing power than the end devices, thus reducing latency and network congestion.

Notably, this approach improves IoT application development, combining FC and IoT dedicated software platforms, harnessing available resources (processing and storage) present on nearby devices. Among the benefits introduced by FC is the reduction in communication latency between nearby devices sharing computational resources on the network edge.

#### *2.3.1.1 Characteristics*

According to Cisco, there are some of the key features for fog computing which allows the necessary extension of cloud computing. The main characteristics of the fog-computing model, which support the IoT, exploit its potential are the following [44][52].

- **Low Latency and Location-Awareness:** Fog nodes bring computation closer to the end-devices, which reduce the physical between the data source and fog server. This helps to reduce the end-to-end latency. Besides, it also helps to provide location-aware services such as a cache of location dependent content as it offers location awareness [51][52].

- **Large-Scale Sensor network:** One of the key scenarios for the fog-computing paradigm is the large-scale sensors network that communicates with the fog nodes. Instead of task million instructions (MI) request to centralized cloud, now sensors can send this request to the fog node. Fog node either use its resources to process the request itself or send it to other nearby fog nodes for further processing which depends on the fog nod availability [49][50].

- **Support IoT Devices in Mobility:** Also, to the distributed IoT devices in the vicinity, the mobility of the interacting end-devices must be taken into account. Fog nodes are not geographically static in the network; end device identity is decoupled from the host location and IP [49]. End devices (wearable devices, static cameras, smart vehicles, smartphones, etc. are widely distributed at the local layer. Fog devices in the fog layer can be used both as mobile and static computing resource platform. Fog nodes can be installed in parks, highways, football grounds, etc. [50][51][52].

- **Real-time interaction:** Real-Time interaction is required for latency-sensitive fog applications instead of batch processing. Fog node significantly reduces data traffic across the internet and also provide high speed services which help fog to meet the demands of real-time interaction for low latency IoT applications.

- **Heterogeneity:** Different hardware resources are used in fog nodes and are implemented in a wide variety of scenarios [49].

- **Bandwidth reduction:** As fog, nodes bring close to the low edge to compute, store and process the end devices data. Fog devices are capable to analyze the generated data from IoT devices in terms of data cleaning, filtering, processing and decision making close to the edge. This helps in the reduction of the network bandwidth due to this computing offer close to the edge, important data is forwarded to the cloud, and most data need not forward over the internet.

- **Interoperability:** Fog node should offer seamless communication between different IoT devices and service providers (video streaming) and resource virtualization.

- **Geographical distribution:** As IoT devices are dynamic, these devices remain in geographically distributed, so instead of centralized computing, fog computing is needed for the processing [49].

#### *2.3.1.2 Standardization*

OpenFog Consortium introduced to the standardization of fog computing, their mission is to control standard bodies to establish specifications so that end devices can communicate safely with other edge nodes and cloud services in a friction-free environment. There are six working groups created by OpenFog Consortium namely communication, security, infrastructure, testbed, manageability and architecture. The responsibilities of such groups are analyzed, recommend standards, practices, and technologies suitable for the design of OpegnFog to overcome the related challenges [50].

In February 2017, OpenFog Consortium releases OpenFog reference architecture. OpenFog reference architecture is a universal technical framework designed to meet the data-intensive needs of 5G, Artificial Intelligence (AI) and IoT. To develop an open architecture fog computing environment, OpenFog architecture is considered as a start point that provides a roadmap and it will be an initial phase in creating standards of fog computing. Figure 9, illustrates the reference architecture introduced by OpenFog Consortium which, is used as a common baseline for achieving a multi-vendor interoperable fog computing ecosystem [50][51].

OpenFog architecture is a compound of multiple perspectives for highlighting participants in the market of fog computing, such as software view, view of node and view of the system. The lowest level view is a node view that contains the abstraction layer of protocols and actuators, sensors and control. To create a system view, a couple one or more node views combined with another component to create a platform. The software view contains the top three layers are above the hardware layer. This paper introduces agility, flexibility, security and other pillars of OpenFog architecture. OpenFog Consortium reference architecture is adopted by IEEE standards for fog computing through the Institute of Electrical and Electronics Engineers (IEEE) 1934 shown in Figure 9.

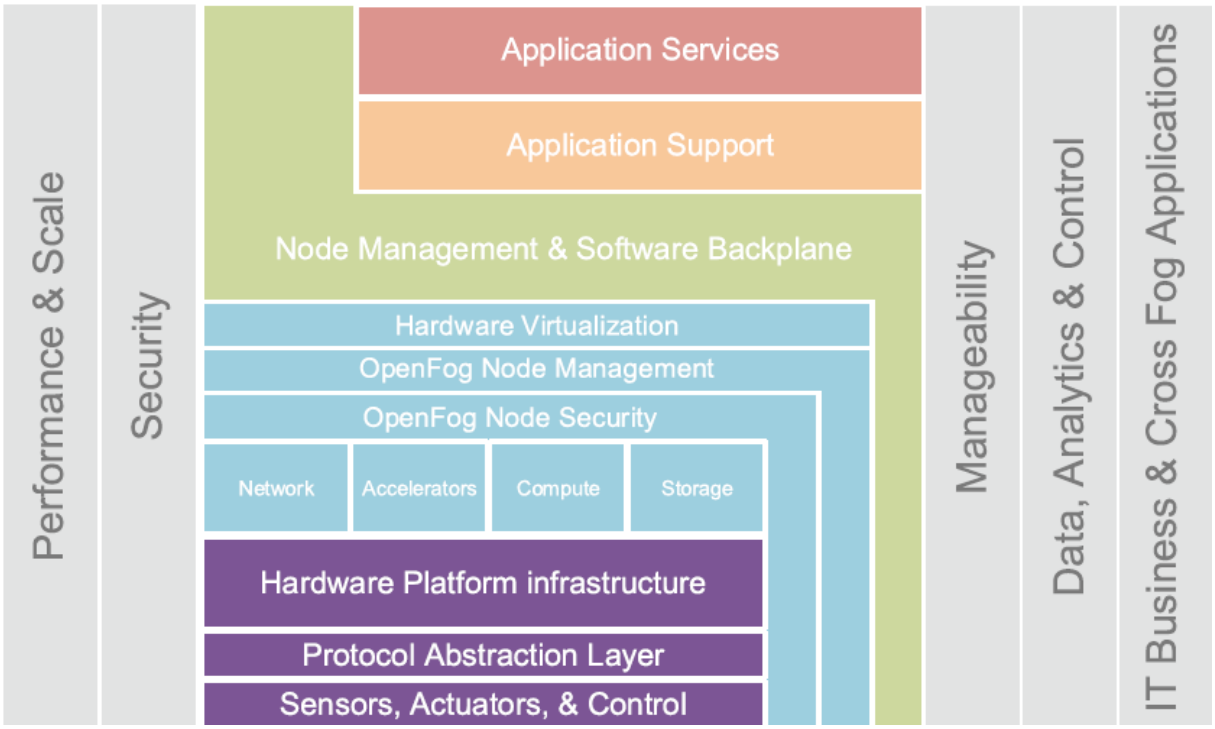


Figure 9. OpenFog reference architecture [50].

### *2.3.1.3 System Architecture*

Several fog computing architecture has been proposed in recent years. Fog computing architecture contains three layers local edge layer, fog layer, and cloud layer. Figure 10 illustrates the hierarchical architecture of fog computing.

End devices layer comprised of end devices such as IoT devices (smartphone, smart vehicle, etc.), sensors, actuators, etc. These nodes have less computational and storage resources. End devices such as smart phone are used to sense the data through surroundings and forward to the fog layer for storage and computing. These end devices are widely distributed in general [53].

Fog layer comprises of fog nodes. Nodes in fog layer are also called fog nodes generally routers, switches, gateways, access points, base stations fog servers, etc. These nodes have more power, computational resources, and storage. Fog nodes can store temporarily, compute, networking and control as a functional point of view. These nodes distributed among clod and end devices, such as train stations, highways, recreation areas, shopping malls, etc. On a moving carrier, they can be at a fixed location or mobile. Fog nodes interact with the end devices and provide their services. Fog nodes in fog layer, latency-sensitive application and real-time data analysis can be carried out. These nodes also connected with the cloud data center via the internet protocol (IP) core network [51][53].

Cloud Layer is the uppermost layer of fog computing architecture. This layer comprised of highly powerful servers and storage devices. For intensive computing and an enormous amount of data storage, the cloud is used. Cloud offers multiple applications services. Cloud resources are efficiently controlled by cloud core modules through control strategies according to the demand-load [49][51][53].

Depending on the network speed and server loads, the processing in cloud computing might take longer execution time. In mobile devices, the delay could be higher because the wireless network capacity is comparatively low. Fog architecture is proposed by some of the researchers to support global mobile devices. This paradigm of computation improves performance and decreases energy consumption in the mobile environment. IoT and mobile internet can get support from fog computing architecture in terms of efficient processing and storage facilities [51].

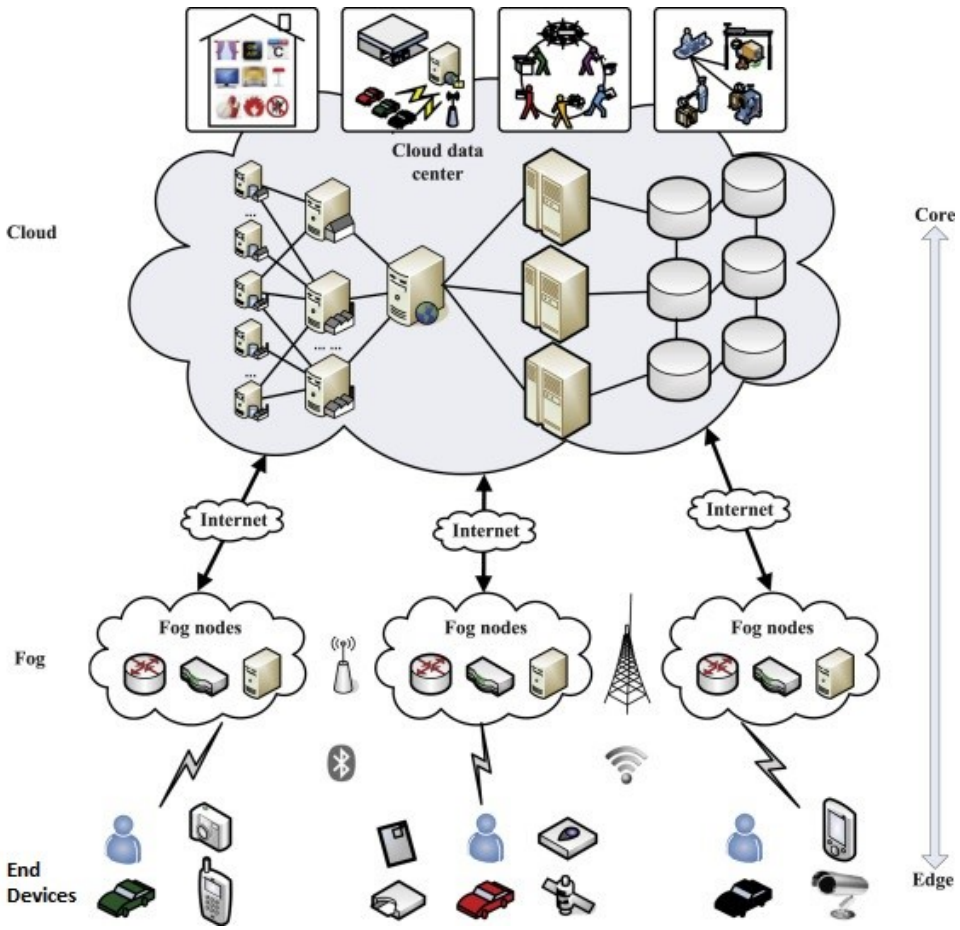


Figure 10. Fog System Architecture [53].

#### *2.3.1.4 Fog platform for IoT applications*

Researchers found fog computing has plenty of interesting applications in multiple aspects compared with centralized cloud computing architecture. Next, we will describe some case studies of new applications scenarios.

- **Video Analytics:** The number of end devices increasing rapidly, traditional camera surveillance system deployed today are not able to process dynamic analysis of complex events in massive cameras. The video stream sent from millions of security/ Closed-circuit television (CCTV) cameras to traditional cloud for processing is not considered

as an efficient way due to large distance and privacy concerns. On the other hand, using fog computing platform, video stream can be processed close to the end devices instead of moving all the traffic to the cloud. It brings better results in terms of latency and provide real time video analytics in a distributed manner [45].

- **Smart Grid:** With smart grid technology, millions of customers, service providers, and manufacturers can smartly manage electricity across the world. It is a fusion of both electrical grid and electrical system, accompanied by telecommunication technologies [47]. Every smart grid network/infrastructure comprises of integrated functions, such as communication gateways, management centers and individual user, which is geographically distributed connected with centralized cloud computing. Companies used every user's private data obtained by the smart meter, such as information about power consumption daily/weekly/monthly, which is used for system monitoring or for pricing. The centralized cloud load can be managed by computing and processing the power consumption data close to the edge of the individual smart grids [49].

- **E-health**: Remote health monitoring services helps serious patients that enable real-time data exchange. As some bad situation occurs, the wearable devices such as smart watch, smart belt, etc. attached to the patient's body or close to the patient can react or alarm the human intervention to a healthcare professional such as hospital ambulance etc. [49]. Real-time diagnosis applications for critical patients requires a reliable connection and provide continuous data with low latency, otherwise, the patient will face death or serious harm in high delay network infrastructure such as centralized cloud computing. Fog computing platform is a most suitable to fulfil these requirements as it reduce the processing and interaction time between the patient and the healthcare infrastructure [50].

- **Smart Home and Cities:** IoT devices such as smart television (TV), phones, Air-conditioner, etc. are available almost in every house and cities these days. To manage and control the data of heterogeneous devices needs distributed intelligence that can be scaled up by a number of devices without degrading the system performance and functionality of the system. To reduce the response time by processing and data buffering, fog computing paradigm can be vital to this kind of requirements [44][49].

- **Connected Vehicles:** In automobile infrastructure, roadside units (RSU) to networks (V2I) and other automobile/vehicle interfaces (V2V) can link automobile, The RSUs offers real-time vehicle connectivity to many moving vehicles through distributing computation tasks. There are few smart vehicles on the road now using Internet of Vehicle (IoV) development, individual automobile contains a processing unit for smart traffic applications. With IoV models, two-way communication can be achieved between the vehicles by installing edge servers on the RSUs and dragging the cloud services to the edge of the RSUs by combining of processing and communication mechanism. With fog computing, smart IoV applications such as self-driving cars, mobility-aware computation and real-time data computing could be efficiently promoted [44][51].

#### *2.3.1.5 Open Issues and Challenges in Fog*

FC have some challenges due to the dynamic behaviour of the network environment, thus becoming vulnerable to standard threats that can exploit the fog computing framework. Researchers have to focus on these challenges in order to realize the full potential of fog computing. Some major challenges are discussed as follows.

- **Secure data storage:** Fog computing face same security threats as the user private data is outsourced to fog node for controlling. Unauthorized parties can abuse the uploaded user data for their interests. Also, the outsourced data could be modified incorrectly, so it is difficult to ensure the data reliability. Auditable data storage service techniques such as homomorphic encryption together with searchable encryption to provide data reliability, integrity for fog storage servers should be proposed to overcome these threats [52].

- **Man-in-the-middle:** With limited fog resources, it is unable to deploy secure communication protocols; attacker can interrupt the data packets between the nodes. Also, hackers could replace the original node with the fake fog node and get sensitive information [52][53][54].The definite approach remain an open challenge as main-in-the-middle has been shown in other studies to be a stealthy threat on fog computing.

- **Distributed denial of services:** Websites and online services in this digital era are facing most challenging security threat called DDoS nowadays. Due to large number of irrelevant services request simultaneously, it is hard for resources constraint fog nodes to deal with it. As a result, fog nodes busy for a long period of time and legitimate services are unavailable for hosting these resources as seen in the Figure 11. Furthermore, DDOS attack can be carried out by fog nodes themselves. Introducers or hackers conduct DDOS attacked on the popular websites such as PayPal, Spotify, and YouTube etc. by accessing the home appliances, which were connected to the internet such as CCTV cameras, printers, Smart Tv's etc. Smart IoT devices have some computational power, which helps to process some tasks in fog computing, results more severe DDOS attacked may take place as compared to traditional DDOS. Researchers and scientists need to work on this security threat and bring new solutions for this solution in fog computing [53].

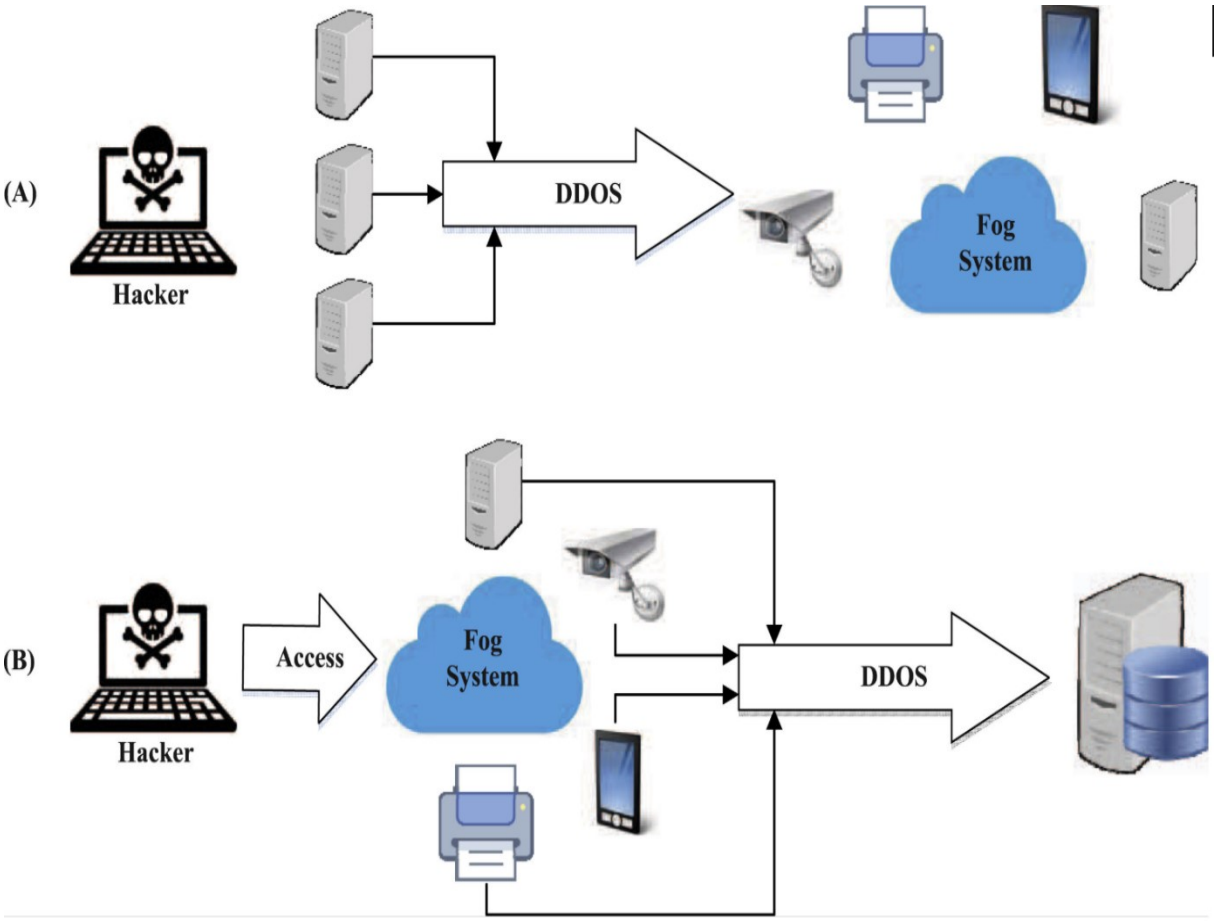


Figure 11. (A) Launch DDOS attack to stop fog device; (B) Send DDOS attacks from Fog device [53].

- **Fault tolerance:** When any specific sensors, networks, applications, and service platforms stop working, fog computing are still capable to provide services normally as number of fog nodes are geographically distributed, users should connect to the adjacent node using corresponding mechanism when the service in a particular area is abnormal [53].

- **Access Control:** System security is ensured by using access control tool. In fog computing, it is not easy to design end device-fog-cloud in order to meet different level resource constraints. In centralized cloud, several encryption techniques have been introduced in order to achieve the efficient data access control [54].

- **Authentication:** Trust and authentication issues may be faced in fog nodes like gateways whereas in cloud scenarios no such issues persist. It is not a preferable choice to rely on cloud central authentications servers even when the remote authentication server communications are down, authentication still have to work constantly to access personal devices locally [54]. Some researchers discussed the authentication and trust issues in the fog, but none of them provides a systematic solution [53].

- **Energy Management issues:** Fog devices are widely distributed in the fog layer as compared to centralized cloud due to this they may consume high energy. Researcher and stakeholders are required to bring new techniques and protocols in order to manage and optimize the energy consumption in the fog paradigm [53][54].

- **Program platform issue:** Edge devices are used to compute at their end in the fog computing, these edge devices runs heterogeneous platform and requires different program which is not easy for fog-computing while on the other hand, program is written in specific program language that runs in the cloud for computational work [53].

- **Fog resource management issue:** Fog computing brings the computation and processing close to the edge network from centralized cloud. Sharing and discovery For Instance fog resource management is critical for application performance. As fog node is handling heterogamous traffic between cloud and end devices in terms of RAM, CPUs, power, bandwidth and supported services [54].

### *2.3.2 Edge Computing*

#### *2.3.2.1 Definition*

The Pacific Northwest National Laboratory (PNNL) introduces the edge computing [56] as "*an approach to move the applications, data and services to logical extremes of the network and it allows information and analytics to occur at the source of the data".*

The Edge Computing Consortium (ECC) defines the edge computing [57] *as an open platform deployed on the edge of the network that is close to the source of the data, and provides intelligent services to meet the requirements of real-time processing, data optimization, security and privacy by mobile edge network infrastructure* [58].

OpenEdge Computing defines "*edge computing as computation done at the edge of the network through small data centers that are close to users"[59]. "The original vision for edge computing is to provide compute and storage resources close to the user in open standards and ubiquitous manner"* [60].

#### *2.3.2.2 Where is Edge?*

As discussed above, the generated data by end devices is computed at the edge or close to the edge of the network in edge paradigm. Here, the core networks equivalent is the edge of the network where end devices directly generate the data from surroundings.

Edge computing (EC) adds a new tier of connectivity at the edge of the network between centralized cloud and end-devices. Edge computing enhance the cloud services efficiently such as computations, processing and management close, up to one hop away from IoT devices in the local network such as the WiFi access points or gateways Instead of depending on the cloud hundreds of centralized cloud data centres [59][60]. It allows the services to utilize the devices available in the vicinity e.g. by offering real-time communication, high data rate and ultra-low latency and also has the capacity to control and limit the private user data.

European telecommunication Standard Institute (ETSI) proposed Multi-access Edge Computing (MEC) a standard solution for forth coming 5G networks. Instead of transferring all the data from the end devices to the centralized cloud, MEC offload the data to the edge of the network for processing and data storage from mobile and IoT devices [61][62][63][64][65]. Figure 12 shows the proximity of end devices in the edge layer.

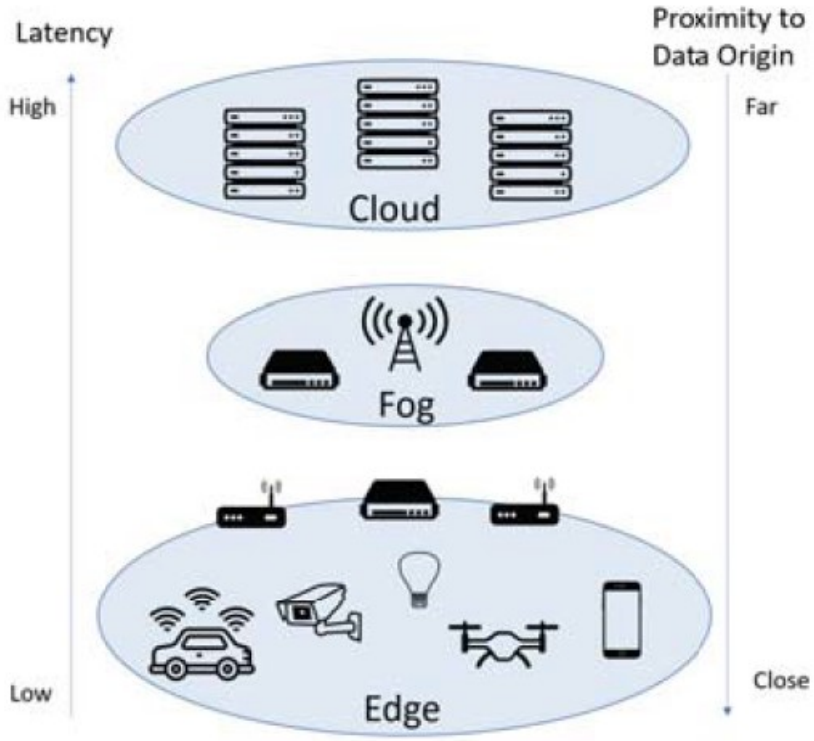


Figure 12. Edge/Fog devices placement in the network system [66].

Comparing with edge computing, MCC (Mobile Cloud Computing) also move the capabilities of the mobile devices and enhance management, storage, computing of end devices generated data. Edge computing is dissimilar with MCC, as it provide computing, processing and analysing at the edge of the network close to the end devices. Edge paradigm offers pre-processing, data filtering IoT data via cloud services installed close to the IoT devices by integrating IoT devices with cloud [59].

#### *2.3.2.3 Edge vs Fog*

Fog jointly works with the cloud, while edge is defined by the exclusion of cloud [54]. Although, the term used as FC is somewhat close to edge computing. With various overlapping definitions described in the literature for both Fog and EC computing, it is still unclear to differentiate between them [61][67][68]. OpenFog Consortium also distinguish that fog computing works in a hieratically manner and it offers storage , offloading , processing ,

computing control anywhere between cloud and things whereas edge computing appears to be restricted for computing to the edge of the network [69].

Chiang et al. [70] fog computing comprises cloud, core, metro, edge, clients, and things. Fog architecture distributes orchestration, managing and securing the resources and functions in the cloud, anywhere cloud-to-end-devices continuum, and support end-to-end services and applications on the things. Instead of treating edge network as isolated computing platforms, it pursue seamless computing services from cloud to the end-devices.

Harjula et el. assumes fog computing is mainly used to provide platform for services which is above of edge network and local end devices network while edge computing primarily refers to the operational edge network/infrastructure. For better system performance at the edge of the network and minimize the core and cloud consumption/load and increase the network durability, fog include pre-process , cache and analytics the IoT devices generated data before send to the cloud [61][68].

#### *2.3.2.4 Benefits of EC*

Edge computing is used to reduce the core network load and is not used to eliminate cloud computing, but it a new addition layer in the network system for processing. Because of its cutting-edge software capabilities various business services have transitioned from cloud to edge, there are various advantages of using edge-computing paradigm for IoT solutions. Few of these are discussed as follows [72].

- **Trust:** With edge computing, the data privacy of local user is safer than cloud and fog computing as the user data remains in the lowest layer and it is easy to manage and control from intruders [72].

- **Proximity**: Communicating and sharing information between the close nodes is more effective than using distant traditional cloud servers. In 19080s and 1990s, peer-to-peer networks gained popularity in this context [72].

- **Intelligence**: As mentioned above, new edge devices have more power capacity and can offer more tasks/instruction to be processed on the edge. This opens the door to automated decision making on the edge, such as distributed crowd-sensing applications or agents that can respond to incoming information flows [72].

- **Control:** The application is controlled and managed in the devices at the edge. Such devices can allocate or delegate to other peers or to the cloud computing, scheduling or storage [72][73].

- **Latency:** In EC, the response time in computation services is counted in milliseconds and supports various SaaS schemes. EC can perform data analytics, predictive analysis and virtualization on edge servers. Relying on its lower latency, EC enables ubiquitous computing in smart applications, where the user can interact with the system in real time and have a better Quality of Experience (QoE). Smart applications, which requires low latency where a local user can communicate with the system in real time and have good Quality of Service (QoS) in edge computing (EC) [71].

- **Human**: User' sensitive information should be computed and storage close to them in order to keep humans in charge of their knowledge [72].

- **Bandwidth and Scalability:** By 2020, 50 billion end-devices produce a huge amount of data, which, send to the cloud using MANET applications such as video streaming, online games, e-commerce, etc. Hence, increase the overall load on the network. EC enables the processing and computing at the edge server can reduce the amount of data to the upper layers of the network, improve the energy efficiency, and reduce the bandwidth utilization in MANET applications. In addition, EC offers low latency for critical applications which requires a prompt reaction for lifesaving events such as in VANET and IoV's. Therefore, the transport network prevents from frequent accidents and it is a strong bend towards EC paradigm for many smart city projects across the globe such as e-health, smart transport [71].

- **Cost Effective**: Centralized cloud servers are cheap for data storage but expensive to get it out. It is reverse in edge.

#### *2.3.2.5 Limitations*

In edge computing, processing nodes are geographically distributed. In fact, edge-based services have to cope with different aspects of constrained environment. This section identifies and address the potential issues in the context of edge computing.

- **Security and privacy:** The most critical services such as data security protection and privacy should be provided at the networks edge. For Example, in a smart home, private data can be analyzed easily through sensor usage data. Intruder/hacker can easily speculate whether or not the house is vacant through the usage of electricity or water usage reading. In this context, it is a problem how to provide service without harming the user's privacy in CC. To keep the data in the edge network for computing, which implies at home, may be an optimal solution to protect the data security and the privacy. The traditional security and privacy mechanism used in CC is not a better solution for edge paradigm. There should be some new security algorithm introduced by the researchers according to the capacity-constraint edge devices. To offer protection against data security issues, it is important to model a lightweight authentication mechanism wherein EC servers authenticate the IoT devices without a time delay. In order to handle this issue, there should be a reliable trust management system incorporate in the edge servers which is capable enough to manage the end nodes and edge servers [73].

- **Trust issue:** As edge servers are geographically distributed over the network, the trust estimate from one EC server cannot headlong the confidence to the other EC servers. In the distributed networks such as VANETs and MANETs, end-devices are mobile and requires time-to-time authentication. An appropriate trust mechanism needs to be deployed in the EC servers, which are capable enough to manage the trust both from servers and from end nodes [74].

- **Programmability:** Users program their code and deploy it on the centralized cloud server. In the cloud, service provider is in charge to decide on which computing device this computation will conduct. Customer/users have limited information of how the application runs, as the cloud infrastructure is transparent to the user. The code is usually written in one programming language and optimized for different target platforms, as the application only operates in a cloud. However, computation is offloaded from cloud in the edge computing, and the edge nodes comprised heterogeneous platforms. It is very difficult for programmers to write an application and deploy in the edge computing as edge device manufacture varies from each other [73].

- **Naming:** As the number of end-devices are large and there are many applications which, runs the services according to the application's requirement on the edge nodes, need for naming scheme in the edge computing like all computer systems for programming, addressing, and data communication is very important [73]. Hence, an effective naming scheme for the edge computing model is yet to be developed and standardized [75]. To link with the heterogeneous objects, typically edge operators require learning specific communication and network protocols within their network system. The main aim of the naming scheme is to cope the dynamic network topology, end-devices mobility, security and privacy. Most of the current networks are well managed using traditional Doman Name Service (DNS) and uniform resource identifier. Due to dynamic edge network and mobility of end-devices, this scheme is not flexible handle these network.

### *2.3.3 Local Edge Computing/Mist computing*

A new term introduced after the Fog/Edge computing is 'mist computing', which is more distributed than fog. Some authors called this layer as 'extreme edge' or 'local edge' [76]. According to [77], mist computing moves the computing closer to the network edge, which involves sensors and actuators devices. Author's supports in [77] that mist computing reduce the latency and increase subsystems' independence. Under such situations, devices self-awareness is crucial, because computation and actuation rely on the device's understanding of the environment. This is considered as the bottom layer of IoT consist of resource constrained IoT devices.

Authors in [78], proposed a new idea of using a mobile device as a cloud-computing environment for processing the data. According to the author, video distribution applications in WiFi infrastructure can reduce network and server load significantly using mist computing/local edge. To be simplified, the edge is one hop away from the end devices, such as WiFi or gateway. In the mist-computing paradigm, IoT devices can be utilized to process the generated data and make decisions locally [76] [78]. Figure 13 clearly illustrates the local edge paradigm.

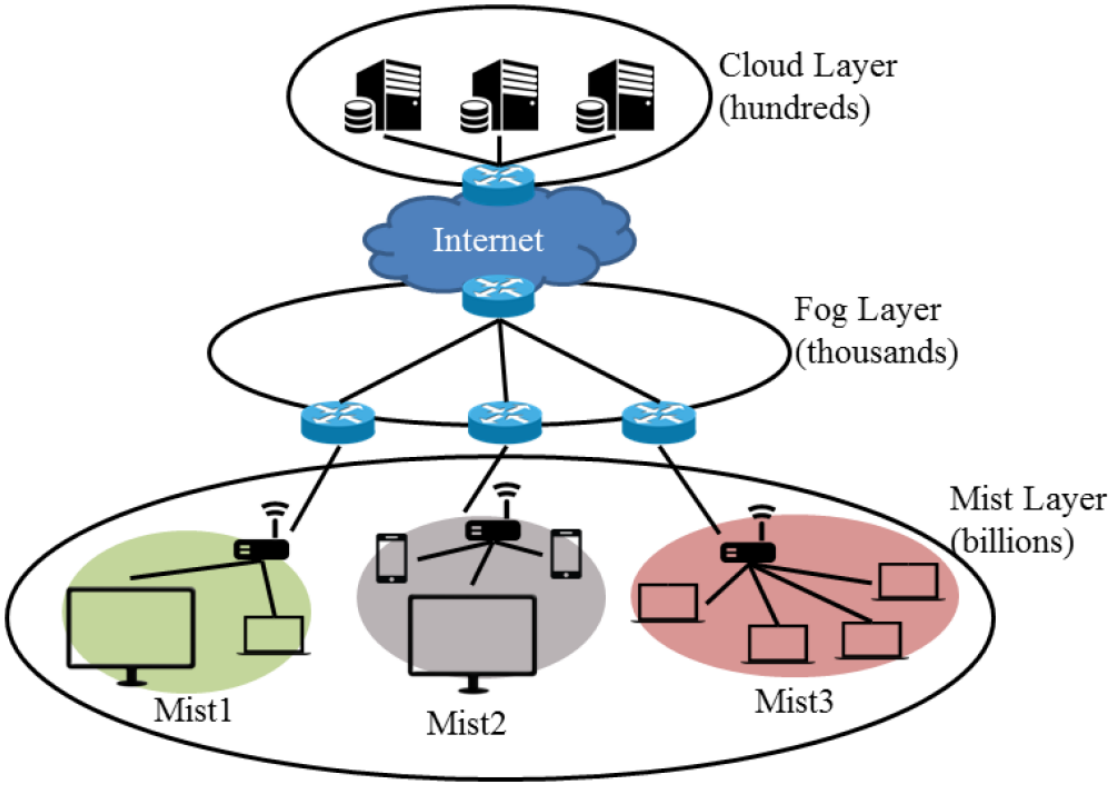


Figure 13. Cloud, Fog and Mist layers [80].

#### *2.3.3.1 Emergence of local edge/mist computing*

The IoT's bottom layer takes operations from the surrounding. It is carried out by small devices made up of sensors, low power micro-controller, a radio module, and a battery. In many cases, these devices build a connection between each and other creating a short-range Personal Area Network (PAN) or Wide Area Network (WAN) using Industrial, Scientific, and Medical (ISM) band. Some researchers called it a mist computing where, processing and computing takes place in sensors and actuators further close to the network edge. Instead of computing, the generated data at edge or traditional cloud servers, sensors are available such as laser range finders, surveillance cameras, and 3D scanners and actuators like servo motors that are equipped with a micro-controller unit that can be utilized to process certain tasks locally [79][80].

End-devices send huge amount of data to process at fog, centralized cloud servers propagate through various routers/switches, and network links between fog and core nodes consume high network bandwidth, which cause network congestion. The application's data is computed and processed at the gateway/router and it is responsible of the operation of the network can reduce the network performance [80][83]. A new paradigm emerges to overcome this challenge called mist/local edge computing. Local edge IoT/mist computing brings the computation and processing at the local layer or with in the end devices clearly seen in the Figure13. Rubio et el. [76] presented local edge/mist computing paradigm to utilize the lowest layer where end devices/points are present (such as smart phones, wearable device, camera). Local edge computing pushes the storage, processing, computing from cloud and fog computing to the end-devices/things.

The mobile devices as a cloud computing environment idea in the vicinity for caching, processing and storage. Also called it a Mobile ad hoc cloud Computing (MACC). According to their research, using mist-computing paradigm, the load of WiFi architecture is reduced for video dissemination applications. In their research , a group of people gathered in a sport event where the users can only use WiFi Direct and exchange video replays with each other without using centralized server and WiFi access points [78][79][80]. With mist computing model, user's data are more secure by keeping the processing in the local layer and deploy virtualized setup on a single board computer sensors where end devices act like a thin server [81][82]. Preden et el. called it Mist [91], where Mist move the processing, computing from centralized cloud to the extreme edge of the network which, comprises sensors and actuators.

#### *2.3.3.2 Benefits of local edge/mist computing*

Real-time application cases require ultra-low latency for processing and computing at the nearest or into the end devices extreme. Some benefits are discussed below.

- **Computation offloading:** Fog and traditional cloud servers can be overloaded by processing a huge amount of end devices data for processing. It is very significant to find a location for offloading in order to determine what appropriate algorithms and trade-offs need to be executed. To reduce the end-to-end latency, offloading is required to the nearby device. To fulfil the application needs, both fog and cloud can be used together in a distributed manner. For critical application that requires real-time data analysis, low latency is required that cannot be achieved by distant locations, results offloading move to the near devices. To overcome this issue, local edge computing/mist computing can be used as it pushes the computation from the centralized nodes to local layer where IoT devices are located. These devices reduce storage and latency of the fog and cloud and increase the autonomy of a solution by providing computation offloading. Zhou et el. presents algorithm [84] to encourage vehicular cloud computing. This algorithm helps to offload to a nearby vehicle in order to reduce the delay. Vehicular cloud computing work can also be seen in [85]. Nearby devices splitting large computation into multiple smaller tasks is presented in [86]. Articles presented in [87][88], the appropriate methods to offload application tasks between smart phones.

- **Security and privacy:** Nowadays, data can be used as a lethal weapon and it requires high data security and privacy to avoid any attack from outside. Local edge computing paradigm offer high data security and privacy to handle user's sensitive information than FC and CC. With LEC, the data processed locally and so, it remains in the local layer. However, computing at cloud and fog servers are not suitable for sensitive user's data, as it requires to pass through various routers/switches to reach the destination for processing. Data can be altered or easily theft between the nodes in access and core network [79][85].

- **Network load reduction:** As the amount of end-devices reached 50 billion by 2020, [45] it will generate zettabyte (ZB) raw data. As the distance between the end-devices and CC is so large and require high bandwidth, (limited routers processing load and communication link load) CC is not an optimal solution to process such huge amount

of data. Therefore, it may cause network congestion/overload. On the other hand, mist/local edge computing reduced the overall network traffic by pre-processing the raw data and only send the reliable data to the cloud for future processing and storage [84].

- **Managing massive distributed resources**: In traditional cloud IoT model, cloud owners control the cloud resources where developers pay for these resources in order to deploy services. On the other hand, in local edge model, users can manage and control their own data information [84].

#### *2.3.3.3 Local Edge Computing Applications*

Researchers have found many interesting applications that can be used efficiently in local edge computing model in multiple aspects than in centralized cloud, fog and edge computing architecture.

- **Smart traffic signal:** When processing end-devices (smart-traffic signal) data on traditional cloud, increase the processing time cause traffic congestion due to the large distance between end nodes and the cloud server. On the other hand, surveillance cameras at each traffic signal feed the video to the local serving node for processing and take decisions locally can reduce the traffic (vehicle and pedestrian) congestion even in the peak hour [84].

- **Autonomous vehicles**: Smart vehicles have a processing unit, which comprises sensors (such as temperature, heat, air pressure, humidity, etc.) and actuators. The generated data can be processed at these computational units to reduce the response time [84][85].

- **3D Bin-picking:** 3D random bin-picking task is a long-standing issue that was identified by using advance tools integrated with ROS-based software. Many vendors investigated and widely adopted the robotic pick-and-place of assorted parts; however, human operators still surpass robotic solutions, particularly with the small parts. The processing time is the main aspect that requires to identify the pieces using traditional cloud servers, test the 3D target position, and measure a collision-free robot trajectory. Although it requires high computational and processing efficiency, a consolidated system conducting bin-picking as a service may bring significant benefits in terms of technology and maintenance costs at the cost of adding a single point of failure in a factory that uses multiple machine-tending workstations [92][93].

# 3 PROPOSED MODEL

The existing computing architectures for IoT system in this section, such as traditional cloud-IoT model and Egde-IoT model along with a proposed model considered as a solution for mass-critical low latency applications and also compare the attributes of dissimilar IoT models are discussed.

### *3.1.1 Traditional Cloud IoT Model*

The conventional IoT architecture comprise three key layers i.e. core, access, and local (device) layers. The local layer contains the low power IoT devices/sensors, the access layer provides a gateway (routers, ISP gateways) to move/transfer the generated data to core layer, and core layer contains the high-capacity routers and switches to route the data to the server-cloud layer. Cloud offers elasticity of network and computational resources and thus IoT device data is offloaded, analyzed, processed, computed and stored for further analysis [110].

The centralized cloud IoT model is shown in Figure 14. This model has for long been utilized successfully for IoT applications. Such IoT applications might not be very restricted in terms of latency requirements and therefore centralized cloud is well suited in such cases where data can be stored for long-term analysis. However, with the evolution of technology, there is a clear need for IoT application with delay-critical requirements such as healthcare, smart vehicles, and industrial (IIoT). Therefore, cloud-IoT based architecture faces various limitations and vulnerabilities in terms of latency, bandwidth congestion and privacy.

End-devices generate massive amount of sensitive data which is computed in the centralized cloud server. This generated data remains at traditional server for future analysis in the distant data centres attracted for various attacks such as man in the middle, service denial, data sniffing etc. Furthermore, IoT devices increase the bandwidth due to data propagation through various hops in the network and reach the centralized server lower the overall network system. Edge paradigm has been introduced recently to address and overcome some of the major limitations in the traditional cloud IoT paradigm [110].

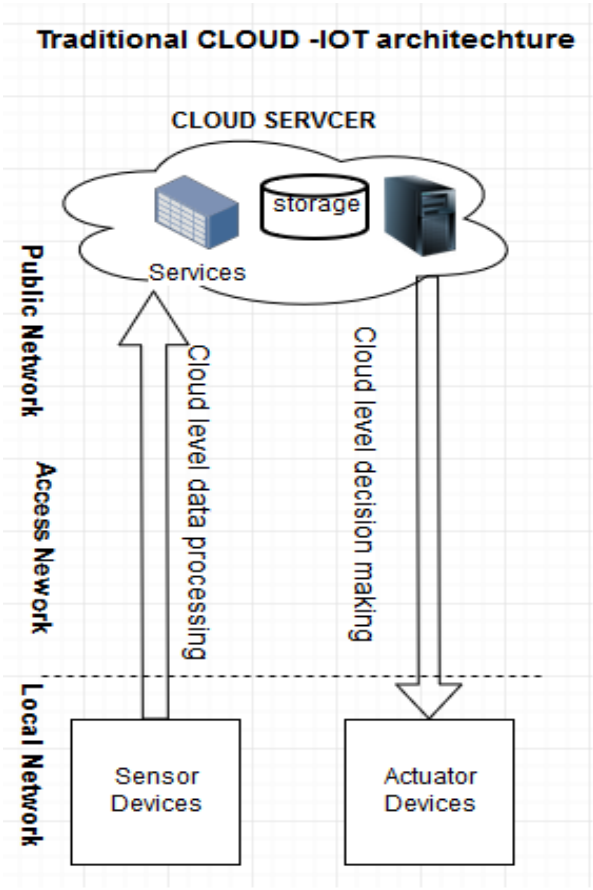


Figure 14. Traditional Cloud-IoT Model.

### *3.1.2 Edge Cloud-IoT Model*

The collaboration between edge network and IoT architecture reduces the size of required physical infrastructure and virtual distance, supports scalability and is more secure as compared to traditional cloud computing architecture. Figure 15 illustrates the Edge IoT model, which allows to reduce the processing burden on traditional cloud by performing some of the data processing and computations at access layer and therefore closer to the end-users. In addition to this, edge also contains cloud server computing, which means it is also convenient to store the data, which is frequently used and highly crucial for decision-making at the edge layer [12].

Edge network is considered as the middle tier connecting centralized cloud to the device/local layer. Therefore, this model is very efficient as compared to the traditional IoT model to handle the end IoT devices data by providing part of cloud services and functionalities at the edge. This model is very much suitable for real-time latency sensitive IoT applications such as video streaming, online gaming, e-health applications etc [12][25].

However, this model has some challenges in terms of connectivity, latency and user data privacy as the generated data moves from local layer to access layer where user data could be exploited. In the case, the connection with edge is lost; it can interrupt the process requiring highly critical information from the edge.

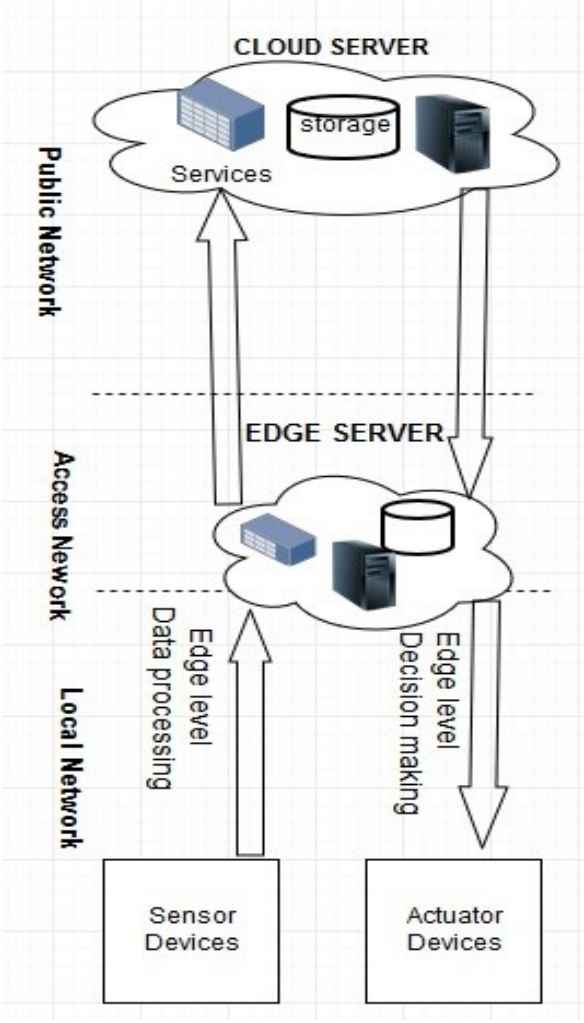


Figure 15. Edge-IoT Model.

### 3.2 Proposed Model: Local Edge IoT Model

A path initiated towards the paradigm shift is presented in our previous project [40], in the relationship between people and digital world. The intelligent environment formed by the digital world provides all the information, tools, and services around users need in their daily life. To achieve the above-mentioned goal, three-tier model is presented in Figure 16 to overcome the issues such as high-latency, network load, vulnerabilities to network occurs between distant computation devices and data sources end-devices. This model integrates the cutting-edge ideas and models discussed in the previous section in a novel way. However, bringing EC capacity within local IoT nodes is useful in several case scenarios. As, IoT nodes cannot be used to support full-function MEC host, due to hardware-constraint. Therefore, for better IoT environment, it is important to research alternate decentralized solutions [45].

Our model aims to reduce computational and the network load at EC and CC by moving computation at locally at the IoT device/local layer. This model is more reliable in terms of providing end-to-end connectivity with the local serving node. We have used real-time video-based automated controlling of wood harvester as use-case where video feed are processed at local layer, edge layer and core layer according to the application complexity to optimize and evaluate the overall network performance. In this model, local layer in IoT will provide the

flexibility/feasibility to scale at the local networks, which makes it easier to add or remove any microservice functionalities in the network. In order to take local decisions and quick response time, virtualized-based microservices, which composed of limited set of functions, utilize the local node resources. In order to compute complex application services, sources data is moved to the more computational capacity edge or traditional server for quick data analysis.

However, later in section, to examine the feasibility of the model, this thesis present a PoC implementation utilizing iFogSim simulator.

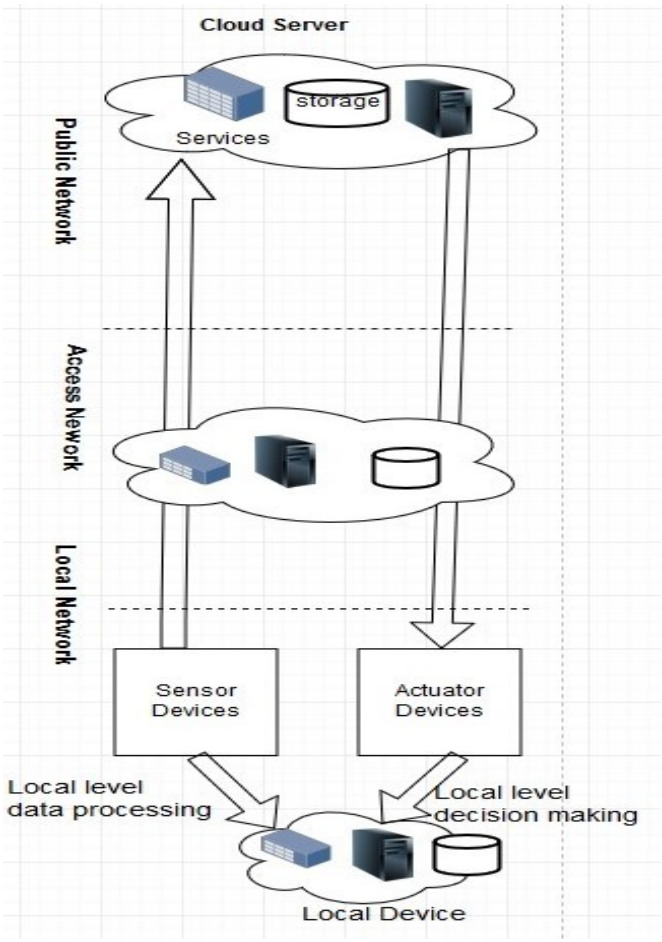


Figure 16. Local Edge Computing Model.

Figure 17 illustrates the application processing among various components in the local edge IoT model discussed above. First, in physical component, fog devices act like cloud data center in CC, edge server in EC, and local node such as sensors and actuators in local edge computing by offering computing, storage, and network resources. In each network layer, fog device is designed with different attributed of instruction execution rate and power consumption (busy and idle resources) which represent its hardware capacity and energy efficiency. Lower fog device as sensors generate tuples (task in MI) is guided by events and the time between creating two tuples is fixed by deterministic distribution which is referred to its sensing interval.

In the next phase, AppModule, AppEdge and AppLoop are created. Logical components contain applications, modules (AppModules) and application edges (AppEdges). Distributed

application is promoted by considering a set of interdependent AppModules features defined the dependency between two modules. AppModules is mapped with VMs, and AppEdges defines the logical flow (task complexity, CPU, network length, and source and destination module direction) between two VMs. Each AppModule (VM) execution depends on a specific type of tasks coming from dataflow predecessor AppModule (VM). In the meantime, with given specification on AppEdge objects different types of tuples are created.

Finally, management component comprised Controller and Module Mapping objects are initiated. According to the AppModules requirement, the Module Mapping entity defines and identifies available resources required for a particular application task and place them in the fog device using scheduling and AppModule placement policy in each network layer. The module is placed in the processing fog device in each network layer. Controller used application placement policy information followed by Module Mapping object and launch AppModules on the fog devices in local edge, access layer and core layer upon submission of application. Later, Controller is responsible to gather performance parameters results in terms of network usage, power consumption, and end-to-end latency from the fog devices and submit the whole system to the CloudSim engine for simulation.

Control algorithm allows the monitoring module to record the resources used by fog devices in each network layer and forward to the resource management entity to meet application level QoS.

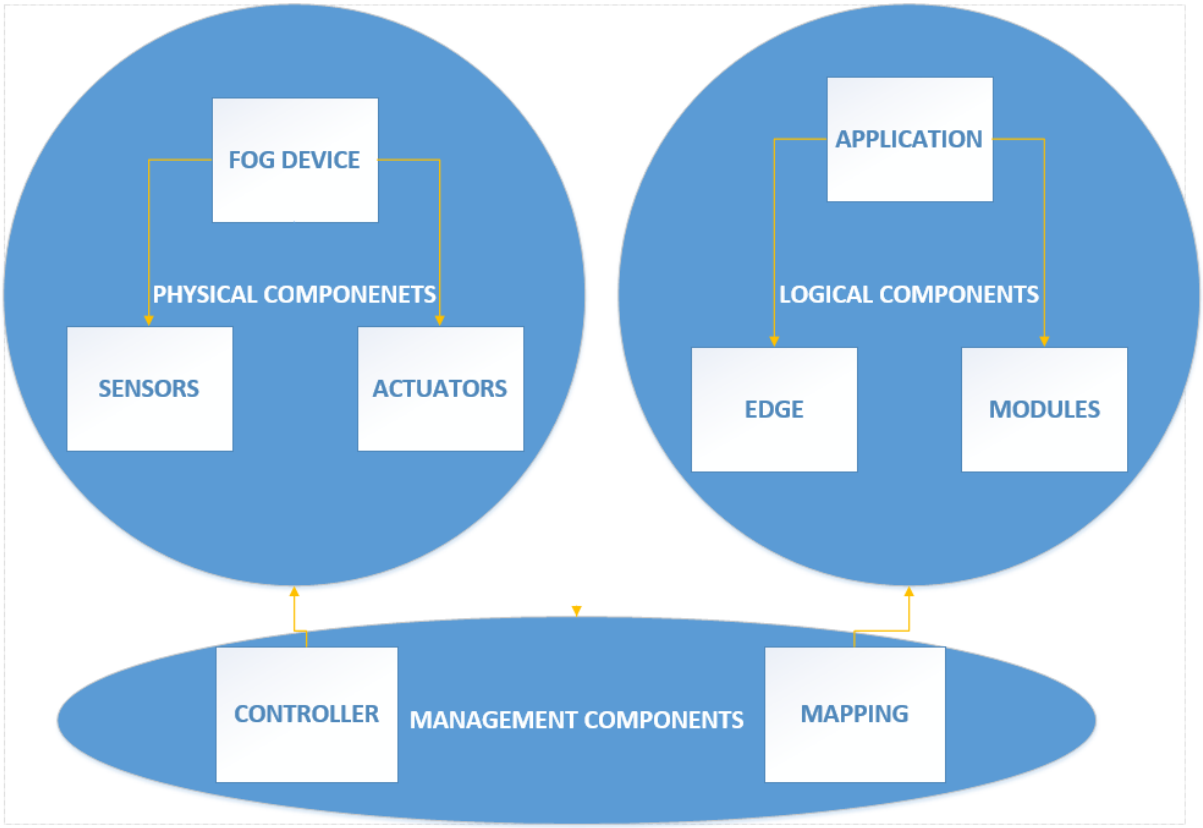


Figure 17. High-level view of component interaction.

Table 1 shows the comparison of relevant IoT paradigm related to the thesis models. Local edge computing model is proposed, after analyzing the attributes of CC, FC and EC.

Table 1. Attributed of CC, FC, EC and LEC paradigm [60].

| Attributes | Cloud Computing | Fog Computing | Edge Computing | Local Edge Computing |
|---|---|---|---|---|
| Hardware | Large-Scale data centers with devices with virtualization capacity | Devices with virtualization capacity (switches, servers etc.) | Edge devices with computing capability | IoT devices (e.g. smart phones, home appliances devices, sensors etc.) |
| Service Type | Global | Less global | Between global and Local | Local |
| Standardization | NIST, OCC, CSA etc. | OpenFog Consortium, IEEE | _ | _ |
| Type of Application | Ample Computation | High Computation with low latency | Low latency computation | Distributed processing on IoT devices |
| Architecture | Centralized/ Hierarchical | Decentralized/ Hierarchical | Localized/ distributed | Localized/ distributed |
| Availability | High | High | Average | Low |
| Latency | Relatively high | Low | Low | Moderate |
| Security | Must be provided along cloud-to-things continuum | Must be provided on participant nodes | Must be provided on edge devices | must be provided on IoT devices |
| Power Consumption | High | Medium | Low | Low |
| Hardware Connectivity | WAN | WAN, LAN, WLAN, Wifi, Cellular | Wan, LAN, WLAN, Zigbee | LAN, Bluetooth, Wi-Fi, cellular, Zigbee |
| Internet Connectivity | Must be connected to the internet for the duration of services | Can operate autonomously with no or intermittent Internet connectivity | Can operate autonomously with no or intermittent Internet connectivity | Can operate with low or intermittent Internet connectivity |
| Available computing resources | High | Moderate | Moderate | Limited |

# 4 PERFORMANCE EVALUATION

This section discussed the simulation tool to model the different IoT model for real-time and non-real-time applications. The selected simulator is proposed in order to evaluate the performance and efficiency of three different IoT based models that are defined in section 2 and 3. These models include: i) traditional cloud-IoT model, ii) a MEC-based edge-cloud-IoT model, and iii) a local edge-cloud-IoT model.

## 4.1 Simulation Environment

To realize the full potential of local edge computing together with IoT networks for real-time analytic, several key performance factors, such as energy consumption, latency and network usage need to be considered. The details about each of these performance metrics have been presented in the previous section.

### *4.1.1 Relevant simulation tools*

In this section, different relevant simulations tools are discussed that can model and evaluate cloud-fog-edge-IoT computing architectures for various network parameters.

Dastjerdi et el [94] suggested resource management and scheduling techniques which include load balancing, resource distribution and migration for fog and edge computing at the software-level to enable real-time analytics. To evaluate and understand fog and edge systems and remove counterproductive policies and tactics , low cost simulation provide best solution as commercial providers/network providers do not share the network infrastructure to third parties to above techniques and build a prototype/test bed is both challenging, expensive, resource-intensive and time-consuming [95].

NS-2, TOSSIM, EmStar, OMNeT, and Avrora are the simulators to simulate traditional network infrastructure in the early stage, in order to develop and test network protocols. These simulators are not suitable for fog and edge computing environment [97].

To evaluate the performance, there are a number of simulators that to evaluate cloud-IoT based computing, but fewer are available for evaluating fog and edge computing scenarios [96]. To simulate large fog networks, FogNetSim++ provides users with detail configuration options. OMNeT++ is an open source tool on which FogNeTSim++ is designed; using OMNeT++ extensive library user can simulate network characteristics using discrete event simulations. FogNeTSim++ also provide mobility model solutions, handover mechanisms and fog node-scheduling algorithms. The Article evaluated traffic management system as a use case using FogNetSim++ simulator to show the scalability in terms of CPU and memory usage, execution time, latency, packet error rate. There are some limitations in FogNetSim++ that it does not support VM migration between fog nodes [98][99].

FogTorch allows fog infrastructure presented by Brogi et al. [100]. FogTorchII is an extension of FogTorch [101]. It offers fog computing paradigm deployments, resource management modelling in terms of Hardware capabilities (RAM CPU cores and storage) and also models software abilities, frameworks, OS etc.). In addition, it offers infrastructure and network level modelling and quality of service (QoS) attributes like bandwidth and latency. To implement variation in communication links used as inputs, Monte Carlo simulations is used in FogTorchII. RAM consumption and Storage shows the percentage indicator as an output in

terms of QoS-assurance and fog resource consumption. It has limitations in terms of scalability and node mobility.

There are various simulations tools for fog and edge computing dynamic infrastructure that may not always hold true. To overcome these limitations, few emulation frameworks were introduced. EmuFog framework is designed for fog computing scenarios. Fog computing infrastructure design can be evaluated by allowing the developers to implement real large-scale applications and induced workloads in the network topology using EmuFog [102].
There are four steps required to implement fog Infrastructure.

- Network topology is generated through XML file, supporting real-world topology datasets.
- To define a network topology, graph representation is presented where graph nodes considered as routers devices and links as a connection between nodes.
- Fog and edge devices are determined and placed according to the placement policy.
- Docker containers are running in each fog node for real applications.

Mayer et al. [102] has proposed EmuFog fog environment emulator. It emulates the switches and routers in the Fog infrastructure. However, emulator does not allow mobility and scalability between clients and fog nodes.

Another emulator Fogbed is introduced in [104]. It is an extension of network emulator Mininet [70]. Fog and cloud testbed can be built using Fogbed. In, Fogbed emulator, API containers can be easily removed, added, and connected from the network topology. Furthermore, resource limitation for a container like CPU time and memory available can be improved at run-time. However, there are some limitations using this emulator like scalability, mobility, security etc. in fog computing aspects.

Lera at el. [54] proposed a simulator called YAFS for different IoT architecture in Fog computing. Author compared the fog nodes performance evaluation in terms of latency, cloud and edge policies, network infrastructure with other available fog computing simulators.

Aazam at el. Illustrate the concept of fog cloud computing and its architecture for low latency IoT applications. Author compared the performance metrics such as processing delay, processing cost, processing capability and task length using cloud and fog computing architecture. CloudSim toolkit and Boston University representative internet topology generator (BRITE) network topology presented by researchers in [96].

Puthal at el. proposed a novel load balancing technique to authenticate the EDCs and find less loaded EDCs for task allocation. The proposed load balancing technique is more efficient than other existing approaches in finding less loaded EDCs for task allocation. The proposed approach not only improved the efficiency of load balancing, it also strengthens the security by authenticating the destination EDCs [71]. Khakimov at el. proposed edge computing network model structure for fog applications to measure the workload of the network nodes in terms of latency, distribution-computing power etc. [105].

Edge analytics as a service (EAaaS) is proposed in [106] for IoT devices to promote latency, scalability etc. EAaaS introduced to overcome required real-time data analysis , an expensive pay-as-you-go model and , traditional cloud IoT analytics services related data privacy concerns which is absent at edge side. For external applications, a gateway-side edge analytics agent and an edge analytic SDK EAaaS offers RESTful interfaces to enable user to develop node integration methods. As a part of Existing RESTful services, authors describe software upgradation capabilities and also for utilizing existing analytic models, machine learning is used. Everywhere Software framework is a new developed commercial edge-computing

platform for IoT gateways by Eurotech. Another source project for edge computing paradigm is EdgeX [107].

EdgeCloudSim is proposed in [108] for edge computing environment. With EdgeCloudSim, both computational, network resources and simulation modelling can be covered. As iFogSim, EdgeCloudSim is also reliant on CloudSim. It also helps to simulate and compare the output using three tier IoT models. One of the main advantage of EdgeCloudSim is support mobility between end devices. However, it does not support scalability.

### *4.1.2 Selected simulation tool*

Bhuyya et el. [109] introduced iFogSim simulation tool where resource management modelling and scheduling techniques can be implemented in different models of IoT computing paradigm. It is based on Java as a tool for simulation of fog, edge and local networks. iFogSim use Distributed data flow (DDF) models.

The key reason to choose iFogSim for simulation is that it offers a hierarchical structure, which helps to place the application at different layers in the network [109]. It is the further extension of CloudSim simulator by including various additional features/ options to place the application at fog/edge layer or even on the local level with some modification using iFogSim. iFogSim allows to simulate real-time IoT based applications, processing in Fog/Edge environment and measure resource and network management metrics such as latency, cost, networking congestion, energy consumption. Simulation is carried out by using IIoT use-case (video-based remote control) on the simulators and compare various computational resources for three IoT models.

### *4.1.3 Hardware Specifications*

This thesis has simulated hardware setup for the entire system to fulfil use-case scenario. Network design setup for three-tier IoT model requires traditional cloud data center, edge node, access point, and surveillance camera. Hierarchal level describes the directly connected devices from cloud to surveillance camera. Table 2 shows the configuration of the devices to design a fully functional scenario.

Table 2. Tentative physical requirements in PoC implementation.

| Devices | Hierarchal level | CPU(GHz) | Power (max-idle) Watt | Up/Down Link |
|---|---|---|---|---|
| Cloud datacentre | 0 | 3 | 650-150 | 100 Gbps |
| Edge Device | 1 | 3 | 350-112 | 10 Gbps |
| Access Point | 2 | 3 | 12.5-2 | 1 Gbps |
| Surveillance Camera | 3 | 1.2 | 4.1-0.66 | 500 mbps |

Figure 18 illustrates the network design topology for three tier IoT models using iFogSim simulator. To implement the use-case, the application is placed and processed at each layer to test the overall performance of the network system. With iFogSim, physical entities are created and their capabilities and configuration are specified which include sensors, actuators, nodes

and cloud VM. In addition, the links connected between the entities are described in the table 2.

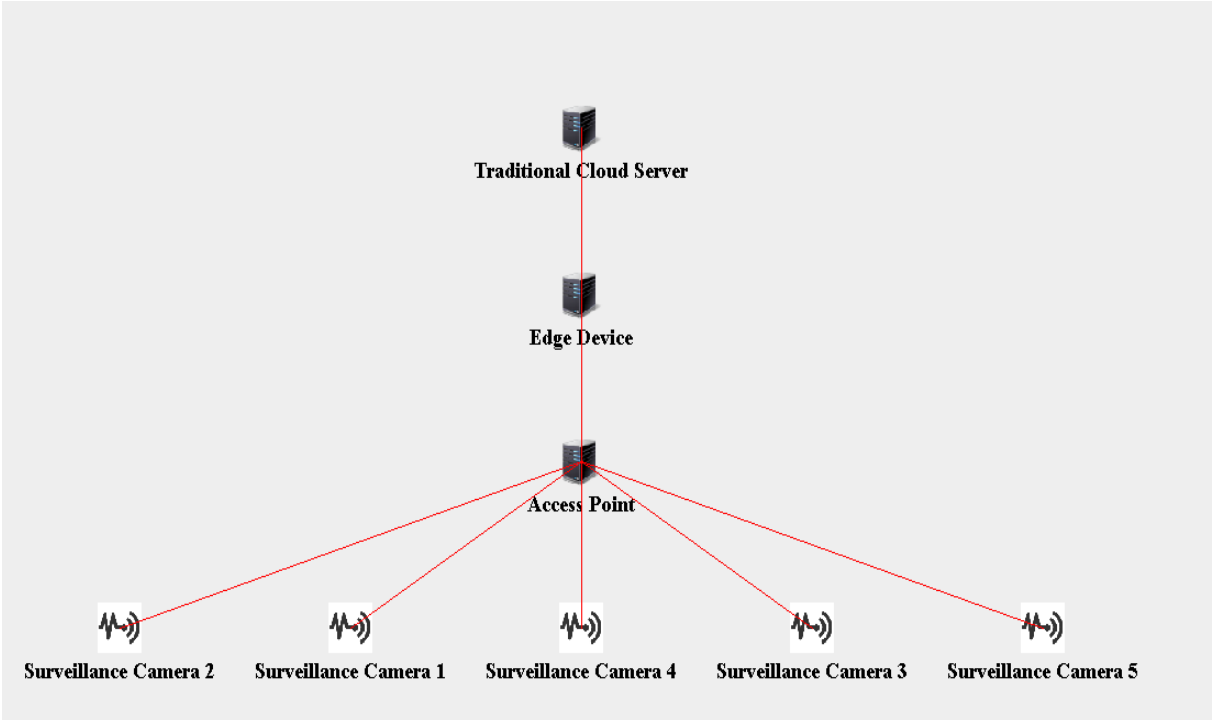


Figure 18. iFogSim design topology for three tier-IoT models using JSON [110].

### *4.1.4 Performance Metrics*

This thesis measures the performance evaluations and resource efficiency for each of three IoT models. Following are some of the key performance factors, which should be considered while analysing the performance of overall application.

- **Latency:** Latency refers to the degree of end-end delay between the time a transfer of a data stream is requested and the actual time when the requestor starts to receive data [110].

- **Energy consumption:** The energy consumption due to the effects of data forwarding, computation, and data storage at each network layer [110]. The power consumption of overall network can be expressed as:

$$E_T = E_C + E_E + E_L \quad (1)$$

Hence, $E_C$, $E_E$ and $E_L$ belongs to energy consumption at local, edge and core layers plus energy of the links between the nodes respectively.

- **Network usage:** The network usage can be referred as the utilization of each of three network layers in the application. It can also referred to the number of packets (KB) that are transmitted across the communication network layers. The network usage increases with the increase of the number of data processing and network devices [110].

### *4.1.5 Use-case: Video-based vehicle remote control*

The real-time video based automated remote controlling of wood harvester is used in our scenario. Video cameras are installed on each side of the wood harvester to record capture and send the video for processing to the node where video feed control algorithm and intelligent video recognition is deployed. The control algorithm is stored on a cloud data center, on a MEC server or on a local computer, depending on the architecture model used.

One of the main considerations in the use case situation is latency, i.e. the video stream and control messages will be transmitted in less time. Local and edge computing mainly regards to delay aspect. Therefore, in video-based vehicle remote control use-case, we are analyzing three-tier IoT model in this thesis to compare and evaluate on various performance metrics in terms of end-to-end latency, power consumption and network usage. Figure 19 illustrates the video-based vehicle remote control. There are five surveillance cameras are used which feed video and process the data locally than it can move to the edge IoT model for processing and in the last it is being computed and processed at the traditional IoT model.

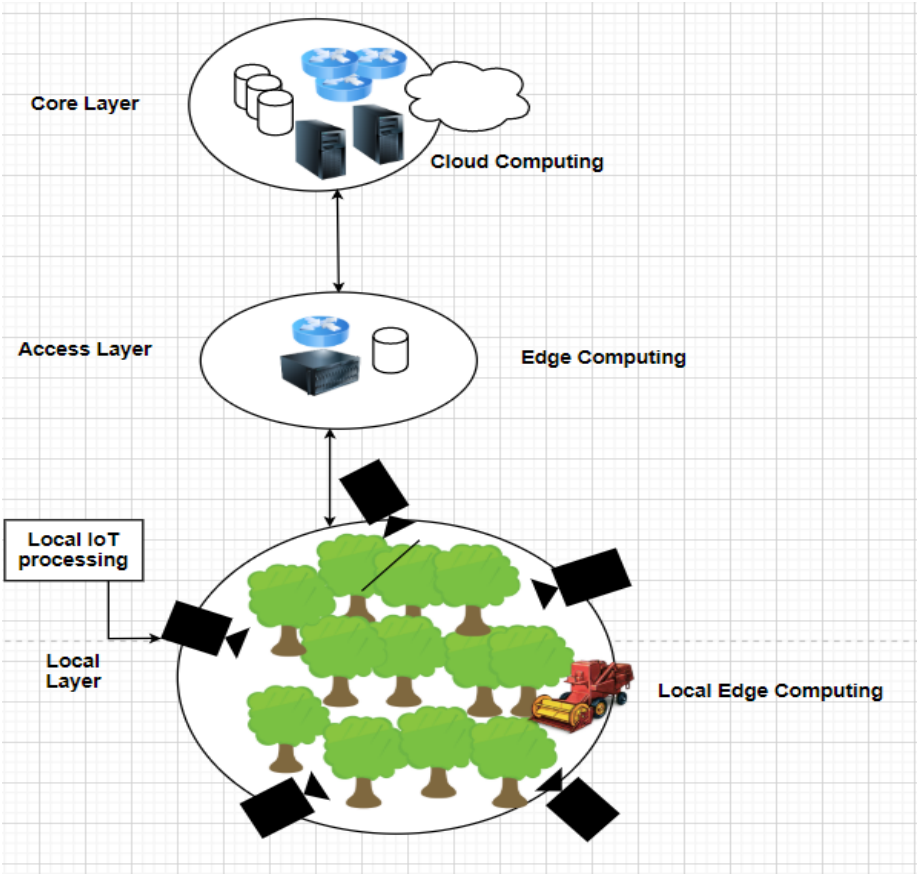


Figure 19. Use Case: Harvester Wood Cutting Video Surveillance.

## 4.2 Results

To verify the presented local edge computing paradigm model in the previous sections, the use-case application “Automated Harvester Wood Cutting Video Surveillance” was implemented

and evaluated in three tier IoT models by running simulations using iFogSim tool. To evaluate and compare the performance of each tier model, some parameters such as end-to-end latency, power consumption, and network usage are considered. Following sections illustrate the results.

### *4.2.1 End-to-End Latency*

End-to-end latency is measured (millisecond, ms) vs task (million instructions, MI) as shown in the Figure 20 by placing control algorithm for different models at each layers of the IoT edge cloud architecture. The complexity of the control algorithm increases as the number of MI increases along other simulation parameters. With a less complex task (complexity below 100000MI) processed at each IoT model, local-edge-model shows promising results compared to other cloud and edge-IoT model. This can be seen where the green and red curves intersect in Figure 20. Edge IoT model provide lowest end-to-end latency as the complexity increase between 100000 MI and 600000 MI, placing control algorithm at edge IoT models is more suitable.

Core layer server shows better results in a term of end-to-end latency with complexity above 600000 MI. In Figure 20, the intersection of blue and red curves shows where core server end-to-end latency surpasses the edge server. The algorithm is most optimal to be run at the cloud IoT model. As the core server have more computational capacity than edge and local servers Figure 20. In general, the algorithm's ideal location depends on its complexity, i.e. the computing resources it requires computing the task at hand [110].

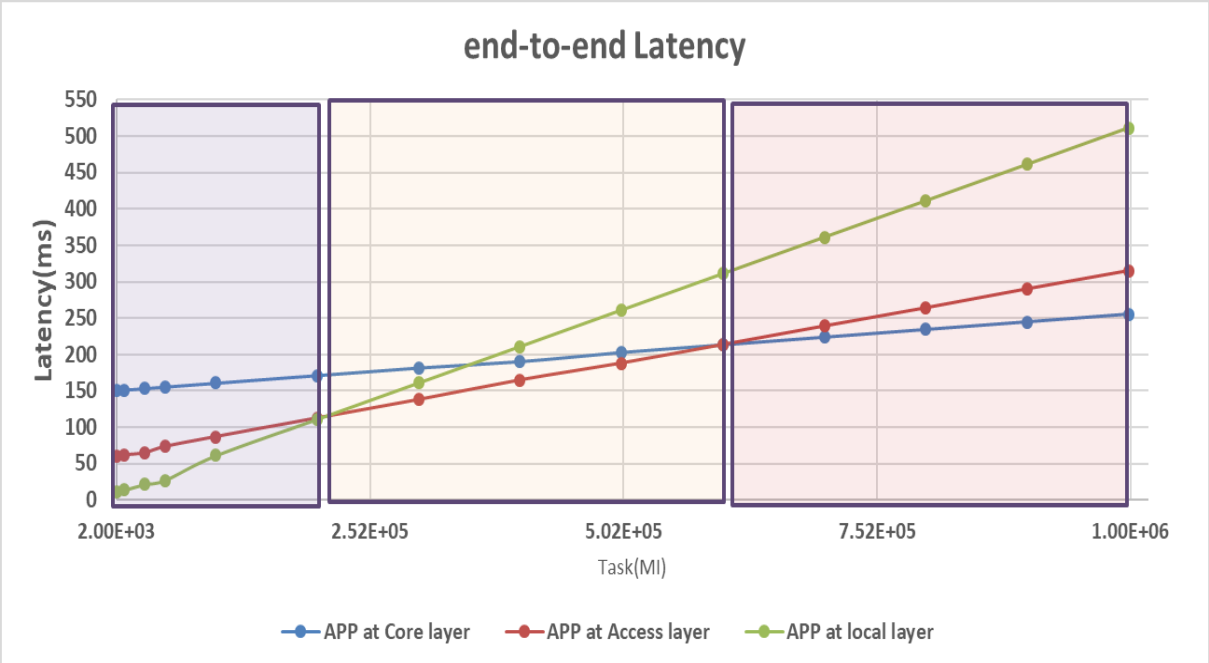


Figure 20. Comparing end-to-end Latency in three-tier models [110].

### *4.2.2 Power Consumption*

The power consumption for different IoT models vs the processing/computation where the algorithm is placed on different layers of the IoT edge cloud architecture described in the Figure

21.Cloud IoT model consumed total power 719.6W and Edge-Cloud-IoT model total consumed 568.9W, which, is 28.9%, less than cloud IoT model. In Local Edge IoT model total consumed 308.2W, which is 57 percent lesser than cloud IoT model.

The network activity and computational load imposes 478 W power consumption to the core layer, when the processing takes place at a cloud server, When the computation takes place at a cloud server, the network activity (including network load on the core network infrastructure) and computational (including network and computational load on the server) load imposes 478 W power consumption to the core layer, edge layer and local layer consumed 219.3 W and 22.3 W power which, comprise only network load on the access network and local network infrastructure.

The capturing node is the same local node in all scenarios connected to the same local router in each scenario results excluding the power consumption of capturing the video and thus its consumption remains integral between the scenarios. When processing takes place at the edge server, including network load and edge server computation load, access layer consumed 397 W, local layer consumed 21.9 W (including local layer network load) and core layer devices remain idle, consume 150 W. Local server consume 46.2W power, when processing takes place at local layer which comprised network and processing load on local layer components. Core and edge layer consume 150 W and 112 W in idle mode.

The impact of running the algorithm on different layers can be clearly seen in results. The simple rule tends to be: The more the video feed is evaluated from the capturing node, the more power is used. This is clarified more by the scenario's data-intensive design, where the raw video feed needs to be sent to the processing node and the longer the path, the more power is consumed. With low capacity-constrained local nodes and networks, high capacity-nodes and network devices are more power hungry [110].

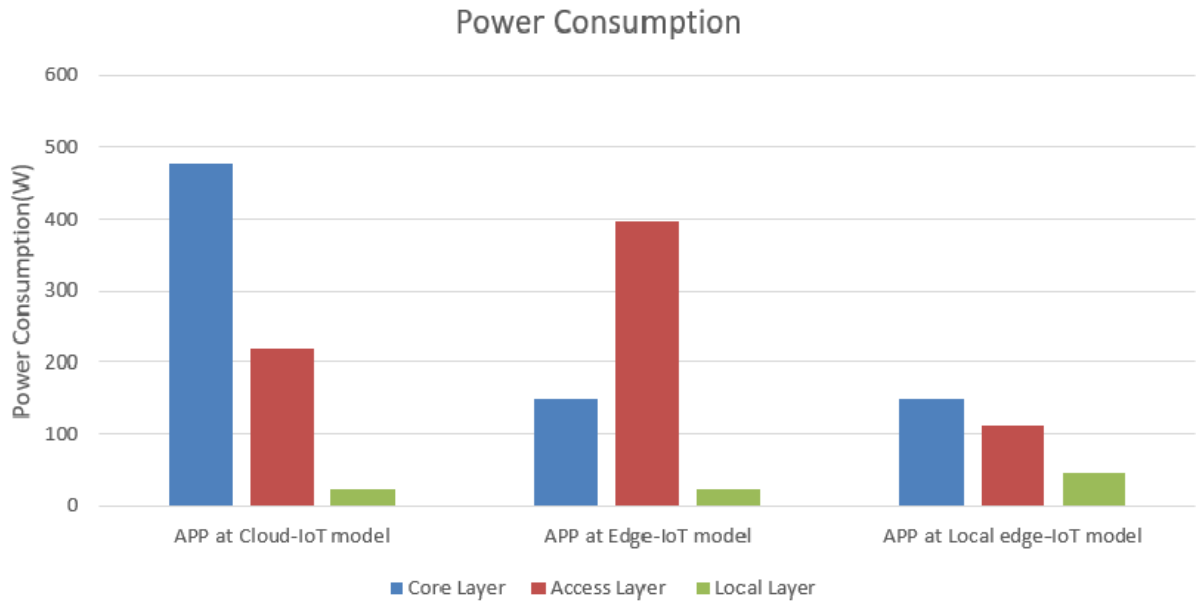


Figure 21. Comparing power consumption in three-IoT models [110].

### *4.2.3 Network Usage*

Figure 22 describes the network usage for three different IoT models, when the number of transferred bytes per send is evaluated. We used full HD video in our scenario resulting in approximately 3.47 MB/s network usage through network devices along the route including

control traffic. In Figure 22, local edge model consume less network usage as compared to cloud and edge IoT model. Local node process the video locally, so the video remain between capturing and processing node results only control traffic is used by core and access layers. On the other hand , video processing at edge IoT model, only control traffic takes place at core layer , while video delivery use the network load to both local and access layers. In the last, analyzed video feed needs to be delivered all the way to the traditional server, when the processing takes place at the cloud IoT model [110].

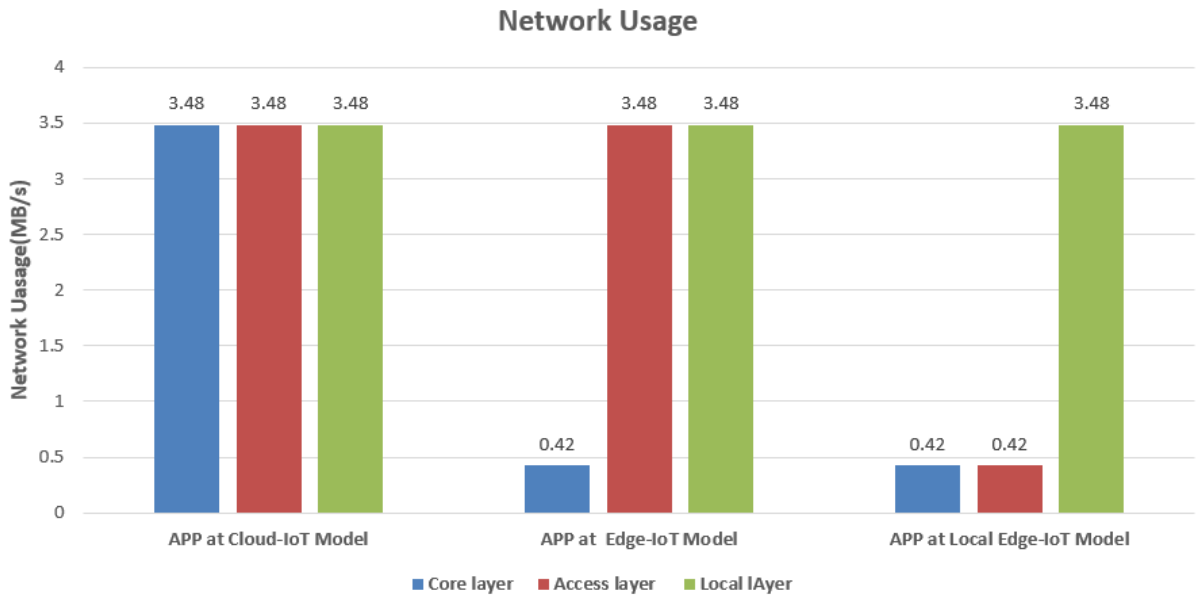


Figure 22. Comparing Network Usage in three-IoT model [110].

### *4.2.4 Comparison with existing work*

Table 3 illustrates the comparison of existing work with this thesis using three-tier IoT models. Muhammad et al. [77] simulated smart city scenario with a different number of end-devices using iFogSim simulator. Author measured and compared performance metrics such as RAM utilization and simulation time using cloud-fog network topology application placement policy. Muhammad et al. [62] evaluated the performance of the use-case scenario online game called EEG by considering cloud and fog based IoT models using metrics time, energy, cost, and network usage. Sarkar et al. [59] used intelligent security surveillance application using a surveillance camera. This work compared and computed various performance parameters such as overall latency, execution cost, and energy consumption at traditional cloud and fog IoT models. None of the above papers have considered the computation/processing at very extreme edge/mist. Furthermore, this thesis utilized the capabilities of local tier along with others. Therefore, a new tier of network layer is focused to enhance the overall efficiency of the network system, which was not considered by the authors in Table 3 using iFogSim Simulator called local/mist computing. The same performance metrics are evaluated as highlighted in the above paper, however, we have considered the novel IoT edge layer as well [110].

Table 3. Comparison of existing work with this thesis using three-tier IoT models.

| Network | This Thesis [110] | Muhammad [77] | Muhammad [62] | Sarkar [59] |
|---|---|---|---|---|
| Traditional cloud only | Yes | Yes | Yes | Yes |
| Edge computing | Yes | Yes | Yes | Yes |
| Mist Computing | Yes | No | No | No |

# 5 DISCUSSION AND FUTURE WORK

The findings of our simulation evaluation using video-based automated wood harvester use-case provide valuable aspects on the usage of local edge computing for IoT-applications of the next decade. For modelling complex IoT domain, iFogSim simulator tool is chosen. iFogSim meet several design objectives: attributes of cloud, edge and sensor customized configuration, policies, and application placement during the simulation.

This thesis introduces a three tier IoT model where local layer including sensors and actuators is mainly focused for processing the end-devices data locally. However, mist computing or extreme edge is not yet considered as a computing layer in the existing research papers, therefore, unfiltered generated data is propagated to the fog and centralized cloud server to make-decision. However, this thesis brings the computation at extreme edge which is considered as a novel approach.

To achieve satisfactory results, hosting computational resources on end-devices, likely in the local layer, is an optimal solution with regard to end-to-end latency, resources consumption and performance. For less complex application tasks, edge and traditional cloud-IoT models provide more than 100ms end-to-end latency, because of a large distance between end-nodes which, is not tolerable for fast-pace applications, however, the proposed model is most suitable in this situation to provide less than 100ms end-to-end latency which can be clearly seen in the Figure 20. In Figure 21 and Figure 22, proposal local edge-IoT computing, reduce the network usage and power consumption compared with two existing IoT models. Our experiment involves five IoT devices (surveillance camera), however, adding more end-devices, the primary results remain the same. Thus, just to mention a few, potential end-devices e.g. inside factory building, surveillance cameras, home and vehicles including plane, private car and local trains, smart TVs can be used at enterprise premises for local computation. This model provides low latency, on device processing, data offloading, as well as storage for trusted-computing and data privacy. Local edge model can enable virtual reality (VR) and online-gaming applications will utilize the enhanced user experience by sending reliable data to edge computing and reduce the network usage and power consumption.

This thesis should be viewed as the foundation as Local Edge IoT is a new and emerging area of study for the researchers concerned to further address numerous obstacles to the centralized cloud IoT model, such as data security and privacy. This work can also be helpful in recognizing and describing the criteria while actual proof-of-concept (PoC) implementation of local Edge-IoT platform. In addition, considering node mobility in the simulation will be more useful to see the system's performance by analysing the impact on end-to-end latency, execution time, number of services executed and cost (network and computational).

In addition, there is another open challenge for Local edgeIoT due to limited capacity devices; researchers should introduce new resource management policies to find how to get extra bit of battery life by considering the migration based on device battery life. Artificial Intelligence is also an emerging technique which should be used to explore the local edge paradigm using dynamic and automated processing and data computing policies in order to enhance the network system overall efficiency.

This thesis research topic can be considered as the base for advanced research in the direction of delay-critical IoT applications, where only traditional IoT models are not sufficient to achieve these goals.

# 6 CONCLUSION

The primary goal of this thesis was to compare and analyze our novel three-tier EdgeIoT model, introduced in our previous articles [39], [40] against the traditional Cloud IoT and MEC-based two-tier Edge IoT models, with respect to latency, energy-efficiency and communication-efficiency. To simulate these three-tier IoT models for the IIoT video surveillance application use-case, iFogSim simulation tool is used. With the assumption that the nodes at the local layer have sufficient computational capacity, tasks handling locally shows optimal results regarding energy and communication efficiency can be seen clearly in the results. iFogSim simulation platform is used to simulate these three IoT models for the video surveillance application.

As local devices are normally very capacity-constrained in most IoT scenarios, it is unfeasible to perform all the computation locally. Access network level is also called MEC where edge computing take place in the next phase for operation. MEC can handle relatively complex computation easily than local node will take for processing and provide low end-to-end latency with relatively more computing capacity as shown in the results. MEC computing models shows optimal results in data sensitive situation than traditional cloud server model in terms of both communication and energy efficiency. MEC can handle delay and mission-critical tasks to avoid the overload situation at MEC level, as the demand in real world is extremely fluctuation and MEC restricted capabilities.

However, due to limited processing capacities nodes of local and edge layers, traditional cloud IoT model provide low energy and network efficiency particularly in computation intensive scenarios for less latency sensitive tasks should be performed on centralized cloud servers. For extremely complex computation activates, traditional cloud model is considered most performing tier with rest to end-to-end latency, as lower-tier nodes takes more time in computation latency than the centralized cloud server end-to-end latency.

The advantages of our three-tier EdgeIoT model are, according to the findings, undeniable. The benefits of all three available computational tiers allow optimization into account, and, therefore, application based requirements and system resources, allows optimized results with rest to reliability, performance and efficiency [110].

# 7 REFERENCES


[1] K. Routh and T. Pal, “A survey on technological, business and societal aspects of internet of things by q3, 2017,” in 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoTSIU), Feb 2018, pp. 1–4.

[2] Bradley, J., Barbier, J., & Handler, D. (2013). Embracing the Internet of everything to capture your share of $14.4 trillion. *White Paper, Cisco*, *318*

[3] Pattar, R. Buyya, K. R. Venugopal, S. S. Iyengar and L. M. Patnaik, "Searching for the IoT Resources: Fundamentals, Requirements, Comprehensive Review, and Future Directions," in IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2101-2132,thirdquarter2018.
doi: 10.1109/COMST.2018.2825231

[4] Ericsson, Ericsson Mobility Report, November2016, availableat: Mobility Report

[5] Recommendation ITU-T Y.2060 Overview of the Internet of Things,document, International Telecommunication Union, Jun. 2012, Art. no. E 38086.

[6] M. U. Farooq, M. Waseem, A. Khairi, and S. Mazhar, ``A critical analysis on the security concerns of Internet of Things (IoT)," Int. J. Comput. Appl., vol. 111, no. 7, pp. 1_6, Feb. 2015.

[7] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, “Internet of Things (IoT) security: Current status, challenges and prospective measures," in Proc. 10th Int. Conf. Internet Technol. Secured Trans. (ICITST), Dec. 2015, pp. 336_341.

[8] K. Sonar and H. Upadhyay, An Approach to Secure Internet of Things Against DDoS. Singapore: Springer 2016, pp. 367_376

[9] R. Ravindran, J. Yomas, and J. E. Sebastian, ``IoT: A review on security issues and measures," Int. J. Eng. Sci. Technol., vol. 5, no. 6, pp. 348_351, Dec. 2015

[10] Q. Gou, L. Yan, Y. Liu, and Y. Li, ``Construction and strategies in IoT security system," in Proc. IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber, Phys. Social Comput., Aug. 2013, pp. 1129_1132

[11] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, ``A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," IEEE Internet Things J., vol. 4, no. 5, pp. 1125_1142, Oct. 2017.

[12] Gartner, Forecast: IoT Security, Worldwide, 2016, available at: https://www.gartner.com/doc/3277832/forecast-iot-security-worldwide.

[13] A. Zaslavsky C. Perera, and D. Georgakopoulos, “Sensing as a Service and Big Data,” Proc. Int. Conf. Adv. Cloud Comput., pp. 21–29, 2012.

[14] H. F. Atlam, A. Alenezi, A. Alharthi, R. Walters, and G. Wills, “Integration of cloud computing with internet of things: challenges and open issues,” in 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2017, no. June, pp. 670–675.

[15] C. Liu, C. Yang, X. Zhang, and J. Chen, “External integrity verification for outsourced big data in cloud and IoT: A big picture,” Futur. Gener. Comput. Syst., vol. 49, pp. 58–67, 2015. International Journal of Intelligent Computing Research (IJICR), Volume 9, Issue 3, September 2018 Copyright © 2018, Infonomics Society 937.

[16] M. Chen, J. Wan, and F. Li, “Machine-to-Machine Communications: Architectures, Standards and Applications.,” KSII Trans. Internet Inf. Syst., pp. 480– 497, 2012

[17] K. Xu, Y. Qu, and K. Yang, “A tutorial on the internet of things: From a heterogeneous network integration perspective,” IEEE Netw., vol. 30, no. 2, pp. 102–108, 2016.

[18] H. F. Atlam, G. Attiya, and N. El-Fishawy, “Integration of Color and Texture Features in CBIR System,” Int. J. Comput. Appl., vol. 164, no. April, pp. 23–28, 2017.

[19] H. F. Atlam, A. Alenezi, R. K. Hussein, and G. B. Wills, “Validation of an Adaptive Risk-based Access Control Model for the Internet of Things,” I.J. Comput. Netw. Inf. Secur., 2017.

[20] D. Bubley, “Data over Sound Technology: Device-to-device communications & pairing without wireless radio networks,” 2017.

[21] A. Gupta, R. Christie, and P. R. Manjula, “Scalability in Internet of Things: Features, Techniques, and Research Challenges,” Int. J. Comput. Intell. Res., vol. 13, no. 7, pp. 1617–1627, 2017.

[22] F. Atlam, A. Alenezi, R. J. Walters, and G. B. Wills, “An Overview of Risk Estimation Techniques in Riskbased Access Control for the Internet of Things,” in Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS 2017), 2017, pp. 254–260.

[23] Mell P., Grance T., 2011 NIST Special Publication 800-145: The NIST Definition of Cloud Computing. Availableat:http://csrc.nist.gov/publications/nistpubs/800-45/SP80045.pdf.

[24] Amazon Web Services "Amazon Elastic Compute Cloud", http://aws.amazon.com/ec2/, February 2011.

[25] Czernicki, B. "IaaS, PaaS, and SaaS Terms Clearly Explained and Defined." http://www.silverlighthack.com/post/2011/02/27/IaaS-PaaS-and-SaaS-TermsExplained-and-Defined.aspx, February 27, 2011.

[26] M. Capra, R. Peloso, G. Masera, M. Ruo Roch, and M. Martina, “Edge computing: A survey on the hardware requirements in the internet of things world,” Future Internet , vol. 11, no. 4, p. 100, 2019

[27] VMware (2007) Understanding Full Virtualization, Paravirtualization, and Hardware Assist. VMware, white paper nov 10, 2007.

[28] Lim, S., Yoo, B., Park, J., Byun, K., & Lee, S. (2012). A research on the investigation method of digital forensics for a VMware Workstation’s virtual machine. Mathematical and computer modelling, 55(1-2), 151-160.

[29] Lee, H. (2014). Virtualization basics: Understanding techniques and fundamentals. *School of Informatics and Computing, Indiana University 815 E 10th St. Bloomington, IN 47408*.

[30] Goldberg, R. P. (1974) Survey of virtual machine research. IEEE Computer Magazine, 7(6):34–45, 1974

[31] Natarajan, S., Krishnan, R. R., Ghanwani, A., Krishnaswamy, D., Willis, P., Chaudhary, A., & Huici, F. (2017). An analysis of lightweight virtualization technologies for NFV. In *Proc. IETF Draft* (pp. 1-16).

[32] L. Vaquero and L. Rodero-Merino, "Finding your Way in the Fog", ACM SIGCOMM Computer Communication Review, vol. 44, no. 5, pp. 27-32, 2014. Available: 10.1145/2677046.2677052.

[33] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar and J. Ott, "Consolidate IoT Edge Computing with Lightweight Virtualization," in IEEE Network, vol. 32, no. 1, pp. 102-111, Jan.-Feb. 2018.

[34] Czernicki, B. "IaaS, PaaS, and SaaS Terms Clearly Explained and Defined." http://www.silverlighthack.com/post/2011/02/27/IaaS-PaaS-and-SaaS-TermsExplained-and-Defined.aspx, February 27, 2011.

[35] Gray, M. (2010). Cloud computing: Demystifying iaas, paas and saas. *Retrieved July*, *17*, 2011.

[36] Williams, A. (2009). The Feds, not Forrester, Are Developing Better Definitions for Cloud Computing. Retrieved July, 3, 2013.

[37] Christian Baun, Marcel Kunze, Jens Nimis, and Stefan Tai. Cloud Computing:Web-Based Dynamic IT Services. Springer Publishing Company, Incorporated, 1st edition, 2011.

[38] Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing (Technical report), National Institute of Standards and Technology: US Department of Commerce.

[39] J. Islam, E. Harjula, T. Kumar, P. Karhula and M. Ylianttila, "Docker Enabled Virtualized Nanoservices for Local IoT Edge Networks," 2019 IEEE Conference on Standards for Communications and Networking (CSCN), GRANADA, Spain, 2019, pp. 1-7.doi: 10.1109/CSCN.2019.8931321

[40] Harjula, E., Karhula, P., Islam, J., Leppänen, T., Manzoor, A., Liyanage, M., ... & Ylianttila, M. (2019). Decentralized iot edge nanoservice architecture for future gadget-free computing. IEEE Access, 7, 119856-119872.

[41] Amaral, M., Polo, J., Carrera, D., Mohomed, I., Unuvar, M., & Steinder, M. (2015, September). Performance evaluation of microservices architectures using containers. In 2015 IEEE 14th International Symposium on Network Computing and Applications (pp. 27-34).

[42] Aazam, M., Huh, E. N., St-Hilaire, M., Lung, C. H., & Lambadaris, I. (2016). Cloud of things: integration of IoT with cloud computing. In Robots and sensor clouds (pp. 77-94). Springer, Cham.

[43] Rong Chunming, Nguyen Son T. Cloud trends and security challenges. In: Proceedings of the 3rd international workshop on security and computer networks (IWSCN 2011); 2011.

[44] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM, 2012, pp. 13–16.

[45] OpenFogConsortium, Openfog reference architecture for fog computing, 2017. [Online]. Available: https://www.openfogconsortium.org/ra/, February 2017.

[46] Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, ACM (2012), pp. 13-16

[47] Mohammad Aazam, Adeel M. Syed, Eui-Nam Huh, “Redefining Flow Label in IPv6 and MPLS Headers for End to End QoS in Virtual Networking for Thin Client”, in the proceedings of 19th IEEE Asia Pacific Conference on Communication, Bali, Indonesia, 29-31 August, 2013.

[48] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami, “Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions”, Technical Report CLOUDS-TR2012-2, July 2012

[49] Bachmann, K. (2017). Design and implementation of a fog computing framework (Doctoral dissertation, Master’s thesis, Vienna University of Technology (TU Wien), Vienna, Austria).

[50] Ai, Y., Peng, M., & Zhang, K. (2018). Edge computing technologies for Internet of Things: a primer. Digital Communications and Networks, 4(2), 77-86.

[51] OpenFog Consortium Architecture Working Group. (2017). OpenFog reference architecture for fog computing. OPFRA001, 20817, 162.
[52] Butterfield, E. H. (2016). Fog Computing with Go: A Comparative Study.
[53] Hu, P., Dhelim, S., Ning, H., & Qiu, T. (2017). Survey on fog computing: architecture, key technologies, applications and open issues. Journal of network and computer applications, 98, 27-42.
[54] M. I. Bala and M. A. Chishti, "Offloading in Cloud and Fog Hybrid Infrastructure Using iFogSim," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2020, pp. 421-426.
[55] Fernández, C. M., Rodríguez, M. D., & Muñoz, B. R. (2018, May). An edge computing architecture in the Internet of Things. In 2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC) (pp. 99-102). IEEE.
[56] "Edge computing," Paci_c Northwest Nat. Lab, Richland, WA, USA, White Paper, Jan. 2013.
[57] ECC, ``White paper of edge computing consortium," ECC, Beijing, China, White Paper, Nov. 2016.
[58] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, ``A survey on mobile edge networks: Convergence of computing, caching and communications," IEEE Access, vol. 5, pp. 6757_6779, 2017.
[59] Sarkar, I., & Kumar, S. (2019, July). Fog Computing Based Intelligent Security Surveillance Using PTZ Controller Camera. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-5).
[60] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji,J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms," 2018.
[61] E. Harjula, P. Karhula, J. Islam, T. Lepp¨anen, A. Manzoor, M. Liyanage, J. Chauhan, T. Kumar, I. Ahmad, and M. Ylianttila, "Decentralized iot edge nanoservice architecture for future gadget-free computing," IEEE Access, vol. 7, pp. 119 856–119 872, 2019.
[62] M. I. Bala and M. A. Chishti, "Offloading in Cloud and Fog Hybrid Infrastructure Using iFogSim," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2020, pp. 421-426.
[63] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, ``Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," IEEE Commun. Mag., vol.
[64] A. Reznik, R. Arora, M. Cannon, L. Cominardi, W. Featherstone, R. Frazao, F. Giust, S. Kekki, A. Li, D. Sabella, C. Turyagyenda, and Z. Zheng, ``Developing software for multi-access edge computing," ETSI, Sophia Antipolis, France, ETSI White Paper 20, Sep. 2017.
[Online].Available:https://www.etsi.org/images/_les/ETSIWhitePapers/etsi_wp20_MEC_SoftwareDevelopment_FINAL.pdf
[65] P. Mach and Z. Becvar, ``Mobile edge computing: A survey on architecture and computation offloading," IEEE Commun. Surveys Tuts., vol. 19, no. 3, pp. 1628_1656, 3rd Quart., 2017. doi: 10.1109/COMST.2017.2682318.
[66] Caprolu, M., Di Pietro, R., Lombardi, F., & Raponi, S. (2019, July). Edge computing perspectives: architectures, technologies, and open security issues. In 2019 IEEE International Conference on Edge Computing (EDGE) (pp. 116-123). IEEE.
[67] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," IEEE Internet of Things Journal, vol. 3, no. 6, pp. 854– 864, 2016.

[68] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, “Fog computing: A platform for internet of things and analytics,” in Big data and internet of things: A roadmap for smart environments. Springer, 2014, pp. 169–186.
[69] OpenFogConsortium. Openfog reference architecture for fog computing, 2017.
[70] Mung Chiang, Sangtae Ha, I Chih-Lin, Fulvio Risso, and Tao Zhang. Clarifying fog computing and networking: 10 questions and answers. IEEE Communications Magazine, 55(4):18–20, 2017
[71] H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty, and C.-T. Lin, “Edge of things: The big picture on the integration of edge, iot and the cloud in a distributed computing environment,” IEEE Access, vol. 6, pp. 1706–1717, 2017.
[72] Argerich, M. F. (2018). Learning based Adaptation for Fog and Edge Computing Applications and Services.
[73] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing:Vision and challenges. IEEE Internet of Things Journal, 3(5):637–646, 2016.
[74] Bo Li, Yijian Pei, HaoWu, and Bin Shen. Heuristics to allocate high-performance cloudlets for computation offloading in mobile ad hoc clouds. The Journal of Supercomputing, 71(8):3009–3036, 2015.
[75] De Donno, M., Tange, K., & Dragoni, N. (2019). Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog. Ieee Access, 7, 150936-150948.
[76] E. Rubio-Drosdov, D. D. Sánchez, F. Almenárez and A. Marín, "A Framework for Efficient and Scalable Service Offloading in the Mist," 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 2019, pp. 460-463.
[77] M. I. Naas, J. Boukhobza, P. Raipin Parvedy and L. Lemarchand, "An Extension to iFogSim to Enable the Design of Data Placement Strategies," 2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC), Washington, DC, 2018, pp. 1-8.
[78] P. M. P. Silva, J. Rodrigues, J. Silva, R. Martins, L. Lopes, and F. Silva, “Using edge-clouds to reduce load on traditional wifi infrastructures and improve quality of experience,” in 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), May 2017, pp. 61– 67.
[79] Portilla, J., Mujica, G., Lee, J. S., & Riesgo, T. (2019). The extreme edge at the bottom of the Internet of things: A review. IEEE Sensors Journal, 19(9), 3179-3190.
[80] Rubio-Drosdov, E., Sánchez, D. D., Almenárez, F., & Marín, A. (2019, April). A Framework for Efficient and Scalable Service Offloading in the Mist. In 2019 IEEE 5th World Forum on Internet of Things (WF-IoT) (pp. 460-463). IEEE.
[81] Ahmed Salem and Tamer Nadeem. Lamen: leveraging resources on anonymous mobile edge nodes. In Proceedings of the Eighth Wireless of the Students, by the Students, and for the Students Workshop, pages 15–17. ACM, 2016.
[82] Roberto Morabito. Virtualization on internet of things edge devices with container technologies: a performance evaluation. IEEE Access, 5:8835–8850, 2017.
[83] M. Vogler, J. M., Schleicher, C. Inzinger, and S. Dustdar, “DIANEdynamic IoT application deployment,” in 2015 IEEE International Conference on Mobile Services, pp. 298-305, June 2015.
[84] Y. Sun, X. Guo, S. Zhou, Z. Jiang, X. Liu, and Z. Niu, “Learning-Based Task Offloading for Vehicular Cloud Computing Systems,” in EEE Int. Conf. Commun. (ICC) 2018, accepted.

[85] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, “A survey on vehicular cloud computing,” in Journal of Network and Computer Applications, vol. 40, pp. 325-344, April 2014.

[86] A. Dou, V. Kalogeraki, D. Gunopulos, T. Mielikainen, and V. H. Tuulos, “Misco: a map reduce framework for mobile systems,” in Proceedings of the 3rd international conference on pervasive technologies related to assistive environments, June 2010.

[87] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, “Serendipity: enabling remote computing among intermittently connected mobile devices,” In Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing, pp. 145- 154, June 2012.

[88] N. Fernando, S. W. Loke, and W. Rahayu, “Honeybee: A programming framework for mobile crowd computing,” in International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services, pp. 224-236, December 2012.

[89] I. Zhang, A. Szekeres, D. Van Aken, I. Ackerman, S.D. Gribble, A. Krishnamurthy, and H. M. Levy, “Customizable and Extensible Deployment for Mobile/Cloud Applications,” in Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation, vol. 14, pp. 97-112, October 2014.

[90] R. Kemp, N. Palmer, T. Kielmann, and H. Bal, “Cuckoo: a computation offloading framework for smartphones,” in International Conference on Mobile Computing, Applications, and Services, pp. 59-79, October 2010.

[91] J. S. Preden, K. Tammemäe, A. Jantsch, M. Leier, A. Riid, and E. Calis, “The benefits of self-awareness and attention in fog and mist computing,” Computer, vol. 48, no. 7, pp. 37–45, Jul. 2015.

[92] Galambos, P. (2020). Cloud, Fog, and Mist Computing: Advanced Robot Applications. IEEE Systems, Man, and Cybernetics Magazine, 6(1), 41-45.

[93] RoboEarth Consortium, “RoboEarth project website.” Accessed on: Nov. 21, 2019.[Online]. Available: http://roboearth.ethz.ch/

[94] Dastjerdi, A.V.; Buyya, R. Fog computing: Helping the Internet of Things realize its potential. Computer 2016, 49, 112–116.

[95] Buyya, R.; Ranjan, R.; Calheiros, R.N. Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In Proceedings of the International Conference on IEEE High Performance Computing & Simulation (HPCS’09), Leipzig, Germany, 21–24 June 2009; pp. 1–11.

[96] M. Etemad, M. Aazam, and M. St-Hilaire, “Using devs for modeling and simulating a fog computing environment,” in 2017 International Conference on Computing, Networking and Communications (ICNC). IEEE, 2017, pp. 849–854.

[97] Yu, F.; Jain, R. A Survey of Wireless Sensor Network Simulation Tools; Washington University in St. Louis, Department of Science and Engineering: St. Louis, MO, USA, 2011.

[98] Qayyum, T.; Malik, A.W.; Khattak, M.A.K.; Khalid, O.; Khan, S.U. FogNetSim++: A Toolkit for Modeling and Simulation of Distributed Fog Environment. IEEE Access 2018, 6, 63570–63583.

[99] Varga, A.; Hornig, R. An overview of the OMNeT++ simulation environment. In Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, Marseille, France, 3–7 March 2008; p. 60.

[100] Brogi, A.; Forti, S.; Ibrahim, A. How to best deploy your Fog applications, probably. In Proceedings of the 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), Madrid, Spain, 14–15 May 2017; pp. 105–114.

[101] Brogi, A.; Forti, S. QoS-aware deployment of IoT applications through the fog. IEEE Internet Things J. 2017,4, 1185–1192.

[102] Mayer, R.; Graser, L.; Gupta, H.; Saurez, E.; Ramachandran, U. EmuFog: extensible and scalable emulation of Large-Scale fog computing infrastructures. In Proceedings of the 2017 IEEE Fog World Congress (FWC), Santa Clara, CA, USA, 30 October–1 November 2017; pp. 1–6.

[103] Coutinho, A.; Greve, F.; Prazeres, C.; Cardoso, J. Fogbed: A rapid-prototyping emulation environment for fog computing. In Proceedings of the 2018 IEEE International Conference on Communications (ICC),Kansas City, MO, USA, 20–24 May 2018; pp. 1–7.

[104] De Oliveira, R.L.S.; Schweitzer, C.M.; Shinoda, A.A.; Prete, L.R. Using mininet for emulation and prototyping software-defined networks. In Proceedings of the 2014 IEEE Colombian Conference on Communications and Computing (COLCOM), Bogota, Colombia, 4–6 June 2014; pp. 1–6.

[105] A. Khakimov, A. Muthanna, and M. S. A. Muthanna, "Study of fog computing structure," in 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). IEEE, 2018, pp. 51–54

[106] Xiaomin Xu, Sheng Huang, Lance Feagan, Yaoliang Chen, Yunjie Qiu, and Yu Wang. Eaaas: Edge analytics as a service. In Web Services (ICWS), 2017 IEEE International Conference on, pages 349–356. IEEE, 2017.

[107] Marc Körner, Torsten M Runge, Aurojit Panda, Sylvia Ratnasamy, and Scott Shenker. Open carrier interface: An open source edge computing framework. In Proceedings of the 2018 Workshop on Networking for Emerging Applications and Technologies, pages 27–32. ACM, 2018.

[108] Cagatay Sonmez, Atay Ozgovde, and Cem Ersoy. Edgecloudsim: An environment for performance evaluation of edge computing systems. In Fog and Mobile Edge Computing (FMEC), 2017 Second International Conference on, pages 39–44. IEEE, 2017.

[109] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in internet of things, edge and fog computing environments," Softw., Pract. Exper., vol. 47, pp. 1275–1296, 2016.

[110] Ejaz, M., Kumar, T., Ylianttila, M., & Harjula, E. Performance and Efficiency Optimization of Multi-layer IoT Edge Architecture