A Lightweight Location Aware Fog Framework for QoS in Internet of Things Paradigm

Thesis Submitted to

The Superior College, Lahore

In Partial fulfilment of the Requirement for the Degree of

Doctor of Philosophy in Computer Science

By

Qaisar Shaheen

Registration No: PHCS-F14-002

Session: Fall 2014

Faculty of Computer Science & Information Technology

The Superior College, Lahore, Pakistan

Author's Declaration

I hereby state that my Ph.D. thesis "A Lightweight Location Aware Fog Framework for QoS in Internet of Things Paradigm" is my own work and has not been submitted previously by me for taking any degree from this University or anywhere else in the country/word.

At any time if my statement is found to be incorrect even after my Graduation, the University has the right to withdraw my Ph.D. degree.

Students Signatur ________________

Date: 23-06-2021

The Superior College, Lahore

Plagiarism Undertaking

I solemnly declare that research work presented in the thesis title "**A Lightweight Location Aware Fog Framework for QoS in Internet of Things Paradigm"** is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged that complete thesis has been written by me.

I understand has zero-tolerance policy of the HEC and University.

Toward plagiarism, Therefore, I as an Author of the above title thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of PhD Degree, the University reserves the rights to withdraw/revoke my PhD degree and that HEC and the University has the right to publish my name on the HEC/University Website on which names of students are placed who submitted plagiarized thesis.

Students/Author Signature: _______________

Name: Qaisar Shaheen

The Superior College, Lahore

Certificate of Approval

This is to certify that the research work presented in this thesis, entitled **"A Lightweight Location Aware Fog Framework for QoS in Internet of Things Paradigm"** was conducted by Qaisar Shaheen under the supervision of Dr. Muhammad Shiraz.

No Part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the Faculty of Computer Science and Information Technology, The Superior College, Lahore in partial fulfillment of the requirements for the degree of Doctor of Philosophy in field of Computer Science and Information Technology at The Superior College, Lahore.

Student Name: **Qaisar Shaheen** Signature: Shaheen

EXAMINATION JURY/BOARD APPROVAL:

Sr. No	Name	Role	Signature
1	Dr. Arfan Jaffar	Chair	
2	Dr. Muhammad Shiraz	Thesis Supervisor	
3	Dr. Usman Hashmi	Co-Supervisor	
4	Dr. Saba Bashir	External Examiner-1	Saba Bashir
5	Dr. Qamar Abbas	External Examiner-2	
6	Dr. Tehreem Masood	Internal Examiner	

Dedication

THE HOLY PROPHET MUHAMMAD (P.B.U.H), WHO LAID THE FOUNDATIONS OF MODERN HUMAN CIVILIZATION AND PAVED THE WAY TO SOCIAL, POLITICAL, MORAL, SPIRITUAL, ECONOMIC, CULTURAL, PHYSICAL AND META PHYSICAL REVOLUTION.

Acknowledgement

This thesis is the culmination of my journey of PhD which is just like climbing a high peak step by step accompanied with encouragement, hardship, trust, and frustration.

At this moment of accomplishment, I am greatly indebted to my research guide, Dr. Muhammad Shiraz, who accepted me as his PhD student and offered me his mentorship, fatherly love and care. This work would not have been possible without her guidance and involvement, his support and encouragement on daily basis from the start of the project till date. Under his guidance I successfully overcame many difficulties and learnt a lot. His own zeal for perfection, passion, unflinching courage and conviction has always inspired me to do more. He has taught me another aspect of life, that, "goodness can never be defied and good human beings can never be denied". For all these, I sincerely thank his from bottom of my heart and will be truly indebted to his throughout my life time.

Thank you Dr. Muhammad Usman Hashmi (my Co-Research Supervisor) for their support and help in the thesis.

I acknowledge the people who mean a lot to me, my parents, Abu (Father) and Ammi (Mother), for showing faith in me and giving me liberty to choose what I desired. I salute you all for the selfless love, care, pain and sacrifice you did to shape my life. Although you hardly understood what I researched on, you were willing to support any decision I made. I would never be able to pay back the love and affection showered upon by my parents.

I owe thanks to a very special person, my wife, for her continued and unfailing love, support and understanding during my pursuit of PhD degree that made the completion of thesis possible. You were always around at times I thought that it is impossible to continue, you helped me to keep things in

perspective. I greatly value her contribution and deeply appreciate her belief in me.

I am sorry to my loving wife and Muhammad Mustafa Qaisar (Son). I am sorry because the time, which I was supposed to spend with you, was given to research. I appreciate my better half for abiding my ignorance and the patience showed during my thesis writing.

Words would never say how grateful I am to both of you. I consider myself the luckiest in the world to have such a lovely and caring family, standing beside me with their love and unconditional support. I love you all unconditionally.

My heart felt regard goes to my father-in-law, mother-in-law for their love and moral support.

I thank the Almighty for giving me the strength and patience to work through all these years so that today I can stand proudly with my head held high.

Author

Qaisar Shaheen

Publications

Journal Publication:

1. **Q. Shaheen** et al., "Towards Energy Saving in Computational Clouds: Taxonomy, Review, and Open Challenges," in *IEEE Access*, vol. 6, pp. 29407-29418, 2018. doi: 10.1109/ACCESS.2018.2833551. ***[IMPACT FACTOR: 4.098]***

2. **Shaheen, Qaisar**; Shiraz, Muhammad; "A lightweight Location Aware Fog Framework (LAFF) for QoS in Internet of Things paradigm" in *Mobile Information Systems*. ***[IMPACT FACTOR: 1.508].***

Supplementary Contributions

During the PhD candidature, I have also contributed to the following collaborative works (this thesis does not claim them as its contributions):

3. Ullah, Irfan ; Shiraz, Muhammad; **Shaheen, Qaisar**; yubo, TIAN ; Ehsan, Kamran; Akhtar, Rizwan; Changda, Wang; Khan, Muazzam "An Improved Intrusion Detection System for Gray Hole Attack in Mobile Ad-hoc Network". *International Journal of Distributed Sensor Networks. (Accepted 20-03-2020).* ***[IMPACT FACTOR: 1.614]***

4. Awan, I; Shiraz, Muhammad; Usman, Muhammad; **Shaheen, Qaisar**; Akhtar, Rizwan; Ditta, Allah; "Secure Framework Enhancing AES Algorithm in Cloud Computing". *Security and Communication Networks.* ***[IMPACT FACTOR: 1.288].***

5. Muhammad K. Khan, Muhammad Shiraz, **Qaisar Shaheen**, Shariq Aziz Butt, Rizwan Akhtar, Muazzam A. Khan, Wang Changda, "Hierarchical Routing Protocols for Wireless Sensor Networks: Functional and Performance Analysis", Journal of Sensors, vol. 2021, Article ID 7459368, 18 pages, 2021.https://doi.org/10.1155/2021/7459368. ***[IMPACT FACTOR: 1.595]***

6. Muhammad Fahad Mukhtar, Muhammad Shiraz, **Qaisar Shaheen**, Kamran Ahsan, Rizwan Akhtar, Wang Changda, "RBM: Region-Based Mobile Routing Protocol for Wireless Sensor Networks", *Wireless Communications and Mobile Computing*, vol. 2021, ArticlID 6628226, 11 pages, 2021. https://doi.org/10.115 5/2021/6628226. ***[IMPACT FACTOR: 1.819]***

Abstract

The world is moving to an era of technologies to make life easier and better. These emerging technologies automate daily tasks to facilitate humans. The large number of connected Internet of Things (IoT) devices have increased the data production. Realization of (IoT) has revolutionized the scope of connectivity and reachability ubiquitously. Under the umbrella of IoT, every object which is smart enough to communicate with other objects, leads to the enormous data generation of varying size and nature. Cloud Computing (CC) employs centralized datacentres for the provisioning of remote services and resources. However, for the reason of being faraway from user devices, CC have their own limitations especially for time and resource critical applications. The data production makes data filtration, storage and processing a difficult task and ultimately burden on cloud is increased. This heavy data traffic and cloud's inability to manage users by tracing their exact location, decreases the efficiency of cloud. The remote and centralized characteristics of CC often results in creating bottle necks, being latent, and hence deteriorate the Quality of Service (QoS) in the provisioning of services. Here, the concept of Fog Computing (FC) emerges that tends to leverage CC and end devices for data congestion and processing locally in a distributed and decentralized way. However, addressing latency and bottleneck issues for time critical applications are still challenging. In this work, a lightweight framework is proposed which employs the concept of fog head node that keeps track of other fog nodes in terms of user registrations and location awareness with addition of load balancing, cloud agent, service management and offloading management modules. Therefore, fog head helps to reduce service time to fulfil user's request by communicating to the fog nodes. The proposed lightweight Location Aware Fog Framework (LAFF) persistently satisfies QoS by providing an accurate location aware algorithm. A comparative analysis is also presented to analyse network usage, service time, latency, RAM and CPU utilization. The comparison results depict that the LAFF reduces latency, network use and service time by 11.01 percent, 7.51 percent and 14.8 percent respectively in contrast with state-of-the-art frameworks. Moreover, considering RAM and CPU utilization, the proposed framework supersedes IFAM and TPFC targeting IoT applications. The RAM consumption and CPU utilization is reduced by 8.41 percent and 16.23 percent as compared to IFAM and TPFC respectively making the framework lightweight. Hence, the proposed LAFF improves QoS while accessing remote computational servers for the outsourced applications in fog computing.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

IoT	Internet of Things
CC	Cloud Computing
QoS	Quality of Service
FC	Fog Computing
LAFF	Location Aware Fog Framework
RAM	Random Access Memory
CPU	Central processing Unit
IFAM	Intelligent FC Analytical Model
TPFC	Task Placement on FC
C-RAN	Cloud Radio Access Network
WAN	Wide Area Networks
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
MCC	Mobile Cloud Computing
DCs	Data Centers
F2F	Fog-2-Fog
FRMS	Fog Resource Management Scheme
VM	Virtual Machine
(RTES)	Real Time Efficient Scheduling
SDCFN	Software Defined Cloud/Fog Networking
PSOCO	Particle Swarm Optimization Constrained Optimization
BFS	Breadth First Search
WSN	Wireless Sensor Network
QDFSP	QoS-aware Dynamic Fog Service Provisioning
SOA	Service Oriented Architecture
LBS	Location based services
MMD	Multi Media Data
TD	Textual Data
DSE	Dynamic Service Execution
NDN	Named Data Networking
GA	Genetic algorithm
SMM	Service Management Module
CDCs	Cloud Data Centers

Fog-DC	Fog Data Center
Fog-MMD	Fog Multimedia Data
Fog-TD	Fog Textual Data

CHAPTER 1

INTRODUCTION

1.1 Overview

This chapter highlights theoretical frameworks and motivation behind the proposed research. It also discusses the statement of problem and objectives of the proposed research. The chapter is divided into six sections. Section 1.2 presents the brief introduction. Section 1.3 highlights motivation behind this research by explaining the importance of the proposed work and significance of the proposed solution. Section 1.4 summarizes the statement of problem. Section 1.5 highlights the objectives of the research. Section 1.6 summarizes the research methodology, section 1.7 summarizes the thesis contributions to the existing body of knowledge and section 1.8 sketches the layout of the thesis.

1.2 Introduction

In most recent couple of years, there have been significant improvements in IoT, and computing advances bring about immensely expanding number of connected IoT gadgets. To connect huge number of physical items and accumulate information from different applications, IoT is the best decision. IoT is spatially dispersed free sensors to screen natural or physical conditions. These conditions may incorporate weight, temperature and sound and so on to send the collected data through the framework to the primary area [1].

The concept of Internet of Things (IoT), supported by computational intelligence has revolutionized in almost all domains of life. Basic infrastructure for such environment comprises of *things* (computing devices) which have computing, communicating and storing capabilities. Based on current trends, it is expected that by 2025 such smart environments will incorporate over 1 trillion IoT devices with 50% increased demand for latency sensitive applications [2]. With every passing day, many new applications and domains in IoT and computational intelligence are emerging to help mankind in one way or the other. On the other hand, providing such applications to general public has opened new horizons of business as well. In IoT, such businesses and applications mostly rely on the sensory data that has to be gathered for effective decision making. For fusion or manipulating big data, (that may be some streaming data or in shape of batches) there are some requirements such as distributed processing capability, effective communication and uncompromised network so that decision making process may yield better accuracy. Cloud being service provider tend to solve these problems.

In early 2000s, Cloud computing (CC) emerges as a new paradigm offering infrastructure services, platforms and software over the internet [3]. In CC, computing resources are managed by datacenters and these datacenters are operated by centralized systems. These centralized systems ensure the on-demand access of CC resources to users. CC is a service-oriented paradigm and possess significant benefits such as: location independence, high resource availability, assured services to users and scalability. However, due to these offered benefits, CC is widely adopted in various domains including education, industry, research and healthcare. In past couple of years, many extensions of CC have been developed. There has been an abrupt increase in the number of smart phone users during 2008-2012. This significant proliferation cause increase in the number of applications, service requirements of users and the practice of IoT technologies. However, cloud services are merged with sensor and traditional cellular networks to handle the foresaid situation. As a result of this mergence, the concept of Cloud Radio Access Network (C-RAN), Mobile Cloud computing (MCC) and Fog computing are evolved [4, 5].

Among the different extensions of the CC, FC offers more services to IoT applications [6]. As FC is extensively distributed in vast geographical areas, providing the computational resources near to the users. This near to the edge distribution reduces the service delivery time and network congestion for IoT applications. FC architecture is a three-layered execution paradigm to provide services between IoT and cloud datacenters through data processing within the communication channels. The deployment of Fog nodes at multiple networking levels is highly scalable and convenient to enhance device-to-device interaction. Hence, it is broadly accepted that FC not only endorses the basic functionalities of CC but also complements the IoT. FC is a framework [3] where vast number of heterogeneous IoT gadgets work together with one another to perform different networking tasks within the same domain, without the interference of other devices and vice versa. The main goal of FC is to decrease the cloud participation by cleaning and pre-processing data generated by the increasing amount of IoT and other sensor devices linked to the FC environment.

Cloud datacenters contain data and services to provide users the storage and computing facilities [3]. These datacenters are located far from the end users and IoT devices. Hence, the communication and data transfer require more time to be executed. This increases the service delivery time. Huge amount of data generation from large number of IoT devices increase network use and cause computational overhead on cloud data centers [8]. To address the foresaid issues of CC, an extension of CC named FC is introduced in 2012 [8, 9]. FC refers to a hierarchically distributed computing paradigm that bridges cloud data centers and IoT devices. The fog environment offers both infrastructure and a platform to run diversified software services. At different hierarchical levels of the fog environment, the physical devices are commonly called fog nodes. This technology overcomes the limitations of cloud computation by enabling data acquisition, processing, and storage at decentralized and locally available fog nodes [12].

FC architecture incorporates a layer between IoT devices/users and cloud datacenters [12]. This fog layer contains computing components such as: gateways, PCs, micro datacenters and nano servers. These computing components are known as fog nodes. Hence, FC resists the transfer of huge amount of data from end users to cloud data centers to reduce the service delay [12]. Moreover, it also reduces the network use as a result of conservation of network bandwidth. Furthermore, the burden of data is transferred from cloud data centers to network edge. Fog nodes lowers the operational cost as they are less expensive [14]. These foresaid significant features make FC a promising paradigm to provide application service requirements to users for better QoS provisioning. However, ensuring rich user experience QoS is the main concern to be addressed specially for time-sensitive applications such as health care IoT [15], web-based gaming [16] and video streaming applications [17]. Hence, a system is required for mitigating resource limitations & ensuring QoS for edge devices. Therefore, the increased number of requests for accessing server’s nodes and provisioning of services results longer response time. Which is a main hurdle in provisioning of QoS.

1.3 Motivation

The standard technique of transferring all the information from the system edge to the cloud adds a great deal of latency and the measure of information, produced from endless gadgets, consume large amount of transfer speed increasing network use and service time. Most of the recently referenced issues, while partner edge devices to the cloud, are additionally featured by the fog computing [17]. Million devices are connected to the cloud producing data at an exponentially growing rate which will be the challenge for the network regarding congestion and network performance. Cloud solution is not feasible for some utilization cases due to high latency and network use. Fog Computing (FC) is a framework [6] where vast number of heterogeneous gadgets work together with one another to perform different networking tasks within the same domain, without the interference of other devices and vice versa. The main goal of FC is to decrease the cloud participation by cleaning and pre-processing data generated by the increasing amount of IoT and other sensor devices linked to the FC environment. The motivation behind conducting this research is to develop a lightweight framework that can provide better QoS to users by reducing latency, network use and service time.

The main goal of this research is to provide a QoS provisioning lightweight framework which can be adopted in FC environment. This study aims to develop a novel and lightweight framework to provide QoS to users. This framework maintains less network use, service time and reduces latency and compared with other state-of-the-art frameworks. The framework designed in this research has been verified and tested in simulation environment.

1.4 Statement of Problem

IoT present a novel way of data streaming of sources through which data is gathered from various sensors over a short period of time. In this modern era where everyone wants fast streaming and downloading, it is difficult to provide better QoS to users in cloud computing paradigm because of its high latency and network use [19]. A number of frameworks are proposed for mitigating resource limitations & ensuring QoS for edge devices. Moreover, data is outsourced to cloud servers which is provided on demand. Therefore, the increased number of requests for accessing server's nodes and provisioning of services result longer response time. Moreover, state-of-the-art frameworks [20, 21] lack to address service provisioning through:

- Location Awareness
- User Management
- Requested data type
- Network condition.

As a result, the existing frameworks employ heavy weight procedures to access fog servers, which reduces QoS. Hence, a lightweight framework is required for leveraging remote fog servers and ensuring QoS for mitigating the resource constraints of IoT devices.

1.5 Scope of the study

The basic goal of lightweight Location Aware Fog Framework (LAFF) is to reduce latency, network use and service time for the end users to provide QoS. By achieving these targets, numerous problems are addressed, the most important of which are the resource provisioning through location management by analyzing requested data type. This study offers the LAFF to provide uninterrupted services to users. As discussed earlier, user can get live streaming without any delay with minimum network use.

1.6 Research Objectives

In time critical systems, numerous IoT devices, applications and Fog nodes continuously interact with each other. These interactions are mostly driven by the service exceptions of the users. In most cases user's location is needed to provide services near to the edge. The objective of this thesis is to improve QoS by providing an algorithm to access the exact location of the users and reroute the user's request in case fog node is hard to reach. To reduce the burden of Fog head node, user's registration is done with location management. A lightweight framework is designed to attain the following objectives:

- *To review state-of-the-art frameworks for accessing fog server's nodes for IoT devices.*
- *To identify and analyze the heavy weight aspects of the state-of-the-art frameworks for accessing fog server's nodes for IoT devices.*

- *To propose a framework which employs lightweight procedures and improve QoS for accessing fog server's nodes for IoT devices.*
- *To develop a location aware algorithm that enables latency reduction, service time reduction and minimal usage of network resources.*
- *To reduce resources utilization (RAM and CPU) to make the framework lightweight.*
- *To validate the significance of the proposed framework by comparing with state-of-the-art frameworks.*

1.7 Methodology & Approach

The procedure of this research work is divided into three steps.

The initial step is gathering data about existing approaches in the fields of CC, FC, and the IoT in combination with QoS provisioning, location management, offloading, load balancing, and user management [22, 23]. After this exercise of gathering detailed information, extraction and keen analysis is performed to develop a theoretical background to design the framework. The data extraction is finished by searching for predefined points referenced in 1.2. Further, an investigation of practical and non-practical necessities is performed.

The proposed framework is developed by using CloudSim [24] and iFogSim [25] simulator on eclipse. iFogSim is an extension of CloudSim. CloudSim and iFogSim are responsible for the simulation and events handling at Cloud and Fog devices respectively. The outcome of the system is reviewed repetitively. After the implementation is done, an extensive evaluation of the framework must be completed to demonstrate the advantages of the applied framework.

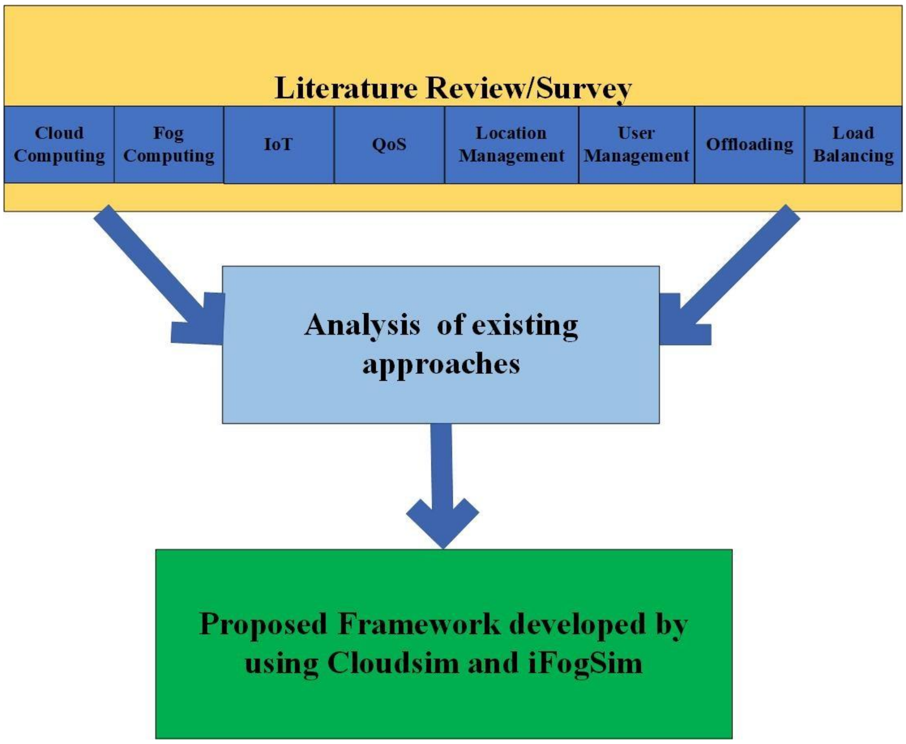

Figure 1.1: *Block diagram of Methodology and Approaches*

1.8 Thesis Contribution

This thesis advances the state-of-the-art by making the following contributions:

1. A comprehensive literature review on QoS, Location Awareness, Load balancing and resource management approaches in Fog computing environment is presented.
2. A location-aware algorithm is designed over distributed Fog nodes.
3. A location-aware framework is developed that minimizes the service time, network use and latency to provide QoS.
4. An analytical model is presented that enables the system to verify the framework.

1.9 Thesis Layout

This thesis includes six chapters, which are organized as follows.

Chapter 2 comprises of two sections. First section includes the technical background of the research. An overview of cloud computing is presented and its limitations are also identified. Fog computing is defined in next step and it is also explained that how it addresses the limitations of cloud computing. The characteristics, features and architecture of the fog computing is described in detail. The second section presents the literature review of the related work. The challenges in FC frameworks in term of load balancing and resource management are also identified.

Chapter 3 proposes a lightweight fog framework, LAFF. It explains the architecture of the proposed framework and distinct modules; user management, location management, load balancing, service management, cloud agent offloading management. Research design and algorithms for LAFF are stated in this chapter.

Chapter 4 is Evaluation. It reports different implemented configurations to evaluate the proposed framework. This section also explains the tools used for testing the proposed framework, data set characteristics of data centers and devices.

Chapter 5 includes Results and Discussion. This chapter elaborates the efficacy of the proposed framework by analyzing the experimental results. Results of this work are presented and discussed in detail.

Chapter 6 concludes the whole thesis. The chapter explains the findings of the research work, significance of the proposed solution and suggestions for future research work. **Table 1.1** depicts the organization of the thesis.

Table 1.1: *Thesis organization*

Thesis Organization **Title: A lightweight Location Aware Fog Framework (LAFF) for QoS in Internet of Things paradigm**				
Background Literature Review	Published in IEEE Access	Chapter 2	Background Publication 1. Shaheen Q, et al. Towards energy saving in computational clouds: taxonomy, review, and open challenges. IEEE Access. 2018;6:29407-29418. *(IF 4.098)* Related Work Quality of Service, Location Management, Load balancing	• Cloud Computing • Fog Computing • Quality of Services • Quality of Service provisioning. • Location Management. • Load balancing.
Lightweight Location Aware Fog Framework		Chapter 3	Lightweight LAFF is a location aware framework, ensures rich user experience in terms of QoS.	• Required Features of LAFF. • Physical topology • Architecture • Characteristics of LAFF • Physical & logical Compmonents
aEvaluation		Chapter 4	Lightweight LAFF is a fault tolerant framework to provide satisfactory QoS.	• The experimental setup • Characteristics of used hardware devices • Topology
Results and Discussion		Chapter 5	Architectural comparison proves that lightweight LAFF reduces latency, network use and service time to improve QoS.	• Architectural comparison • Simulation results • Resource utilization comparison
Conclusion and Future Directions	Published in Mobile Information Systems	Chapter 6	LAFF is the solution for latency sensitive applications as it reduces latency, network use and service time more competently as compared to other frameworks. Publication 2. Qaisar Shaheen, Muhammad Shiraz, Muhammad Usman Hashmi, Danish Mahmood, Zhu zhiyu, Rizwan Akhtar, "A Lightweight Location-Aware Fog Framework (LAFF) for QoS in Internet of Things Paradigm", Mobile Information Systems, vol. 2020, Article ID 8871976, 15 pages, 2020. https://doi.org/10.1155/2020/8871976.*(IF 1.508)*	• Quality of Service provisioning. • Location Management. • Load balancing.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1 Overview

This chapter includes the technical background of the study. First, cloud computing is discussed in detail and its limitations are highlighted. Fog computing is defined in next step and it is also explained how it addresses the limitations of cloud computing. The characteristics, features and architecture of the fog computing is also explained. The second section of this chapter presents the literature review of the related work. This section classifies current fog frameworks on the basis of load balancing and resource provisioning. The challenges in FC frameworks in terms of load balancing and resource management are identified. The most significant work of related area is exhibited and examined in this part. After the related work is presented, a table is included which outlined the related work in a brief manner. The significant aspects of the diverse research papers are compared at the end of the chapter.

2.2 Cloud Computing

CC is an Internet based framework that gives shared resources and data to PCs and various devices. It is a model to enable general access to a shared pool of configurable resources (e.g., PC 3 frameworks, servers, and applications). Distributed computing gives limited resources to customers and various capacities to store and process their data in outside server that may be arranged far from the client extending from over a city to over the world [26]. **Figure 2.1** shows the Cloud Computing Paradigm.

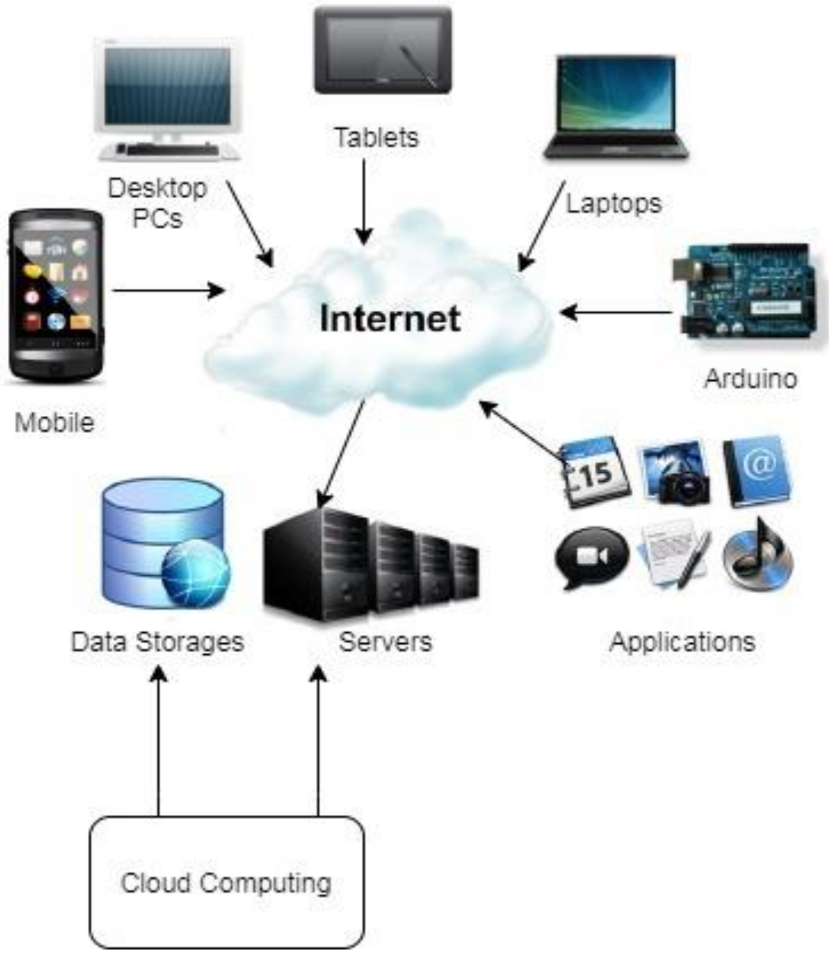

Figure 2.1: *Block diagram of Methodology and Approaches*

CC services give more benefits as compared to a conventional computing paradigm. These benefits include reliability, manageability and most importantly low cost. It is easier for users to access their data from anywhere by using CC irrespective to the location and devices as data is located in cloud data centers [24, 25]. The main purpose that all cloud service provider seeks to offer the finest cloud computing services against time to make it better by every passing day. CC includes vast number of powerful processing nodes linked with one another and to the remainder of the internet with high bandwidth networks. However, cloud data centers store cloud resources and usually located far from the end users, they are serving. The latency between an end user and the nearest accessible cloud server is typically ranged from 20-40 ms over wired systems, and up to 150 ms over 4G portable systems [29]. In spite of the fact that this is entirely adequate for an extensive range of applications, various latency sensitive applications such as augmented reality-based games expect end-to-end latency including processing delay and network under 10-20 ms [27, 28]. These restraints make it difficult to host such latency sensitive applications in the cloud. In another use case, the developing number of IoT gadgets create an enormous volume of sensory information consistently [32]. Sending all the gathered information to the center, cloud utilizes long distance Wide Area Networks (WAN) [33]. An undeniable answer to address these issues is to put cloud server nodes extremely near to the users, inside the network hops. In FC, computational hubs/nodes are extensively distributed in large number of geographical areas so computation capacity is consistently accessible in immediate proximity of any end user.

A hybrid cloud gives access to both people in general and private. A community cloud setup is deployed by a community to achieve their objectives. Cloud provides services in three fundamental categories: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Cloud Services Model are shown below in **Figure 2.2.**

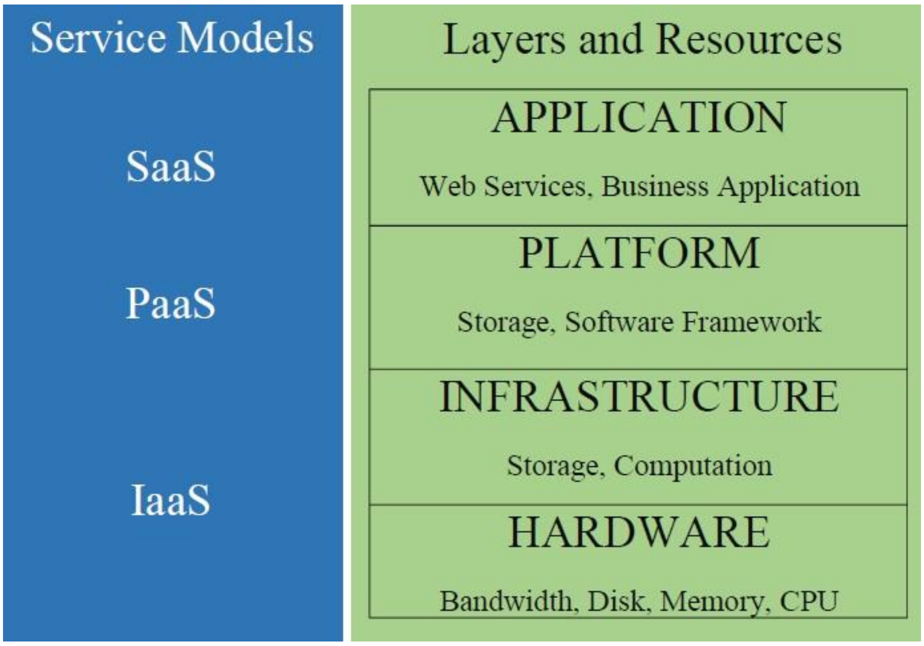

Figure 2.2: *Cloud services models with respect to layers and resources*

IaaS is a model which co-partners two parties: customer and cloud provider. It plans at virtual delivery for cloud resources at the client's doorstep. Customers can control the resources as per their need. The cloud provider extends benefits for customers such as storage space, processing, and networks. PaaS enables clients to use operating system as a service [28]. The same provision operating system is rented to numerous clients. Through SaaS, client can use software on rent/service.

2.2.1 Cloud Computing Deployment Model

The above-mentioned service models are hosted in several different deployment models listed below [34].

2.2.1.1 Private Cloud

The private cloud [34] is a technique in which an association/organization uses a cloud domain devoted to that association/organization. The organization controls the ownership, management and activity of cloud resources independently, by another outsider association, or by a blend of these two. The advantage of this model is that the information is sent to the private cloud for that specific association/organization, which means that the sensitive information is not granted to some other association identified within the same environment. The failure of a particular private cloud to secure data would be the single reason for the failure of cloud security. **Figure 2.3** elaborates the concept of private cloud.

2.2.1.2 Community Cloud

The community cloud [34] works on a similar thought as the private cloud, however, the cloud environment is dedicated to the utilization inside a particular community. The ownership, operation and management are the responsibilities of the community or organization forming the community cloud. **Figure 2.3** elaborates the concept of community cloud.

2.2.1.3 Public Cloud

In public cloud [34], the cloud environment is not devoted to a specific association or organization. It tends to be used by the general populace. The proprietorship, set-up and the management, can be dealt with either by the administration, government, some other affiliation, or a blend of them. The benefit of the public cloud is the open availability to everyone without any limitations to specific association/organization. The drawback of this openness is information and administration security. **Figure 2.3** elaborates the concept of public cloud.

2.2.1.4 Hybrid Cloud

The hybrid cloud [34] is a blend of private and public cloud. Some cloud resources are operated in private cloud such as management and ownership, while others are used through public cloud. For instance, secret information is stored in private cloud and non-critical information is kept in the open cloud. **Figure 2.3** elaborates the concept of hybrid cloud.

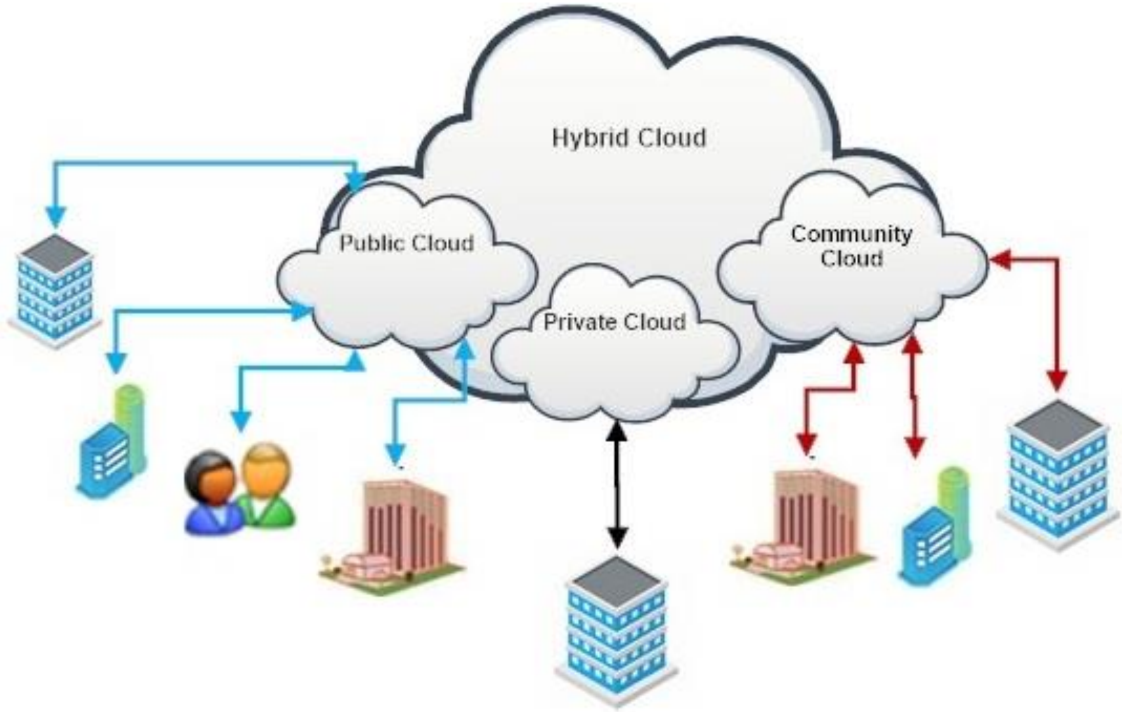

Figure 2.3: *Cloud computing deployment models*

2.2.2 Mobile Cloud Computing (MCC)

MCC has risen up out of the well-explored technologies. Basic idea behind MCC is CC environment merged with smart mobile devices. Processing power of the latest mobile devices is increasing, but the battery life and capacity of data storage is still not as per the requirement. Heavy tasks such as natural [32, 33] language processing and image processing need to be processed in cloud. Mobile devices processor waste power by neglecting their huge amount of processing power consumption especially at night. The theme of MCC is to offload heavy tasks, save storage space and reduce energy consumption by forwarding the data to the cloud, powerful IoT devices [34, 35] and mobile cloud. Application scenarios of MCC incorporate mobile healthcare (maintenance) and mobile commerce. In mobile healthcare services, mobile devices gather data and play a vital role to process lifesaving information that needs to be handled quickly. In mobile commerce, the new communication patterns and outcomes are the most significant prospects [35]. MCC paradigm is elaborated in **Figure 2.4**.

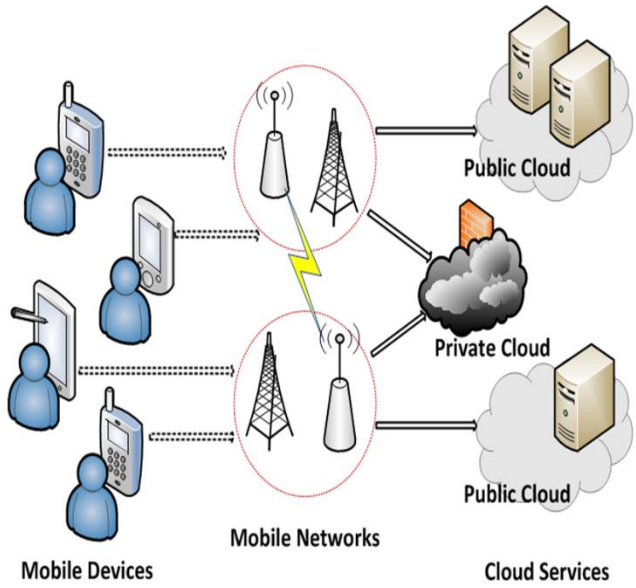

Figure 2.4: *Mobile Cloud Computing paradigm*

2.2.3 Limitations of Cloud Computing

The significant features of CC such as storage capacity, flexibility, and scalability have made it a very popular computing paradigm. However, CC does not come without its limitations. CC possess powerful data centers connected with high-speed network. These datacenters are deployed far from the users increasing the service delivery time. Following are the issues that limit the performance of some applications deployed in the CC environment:

a) The latency sensitive applications such as augmented reality and video game streaming need a maximum end-to-end latency of 20 ms (including network and processing delays) [39]. Conversely, in wired networks the latency between end users and their closest data centers ranges from 20 to 40 ms and 40 to150 ms in 4G mobile networks [40, 41]. Such delays minimize the better and efficient service provisioning in cloud environment [42].
b) Every day a large amount of data is generated from enormous number of IoT devices [43]. This data is processed and analyzed in cloud data centers. The transmission of this vast data in cloud data centers over long distance consumes large amount of unnecessary network, energy and resources [44] causing service delay.

2.3 Fog Computing

Because of the increased data generating IoT gadgets such as cell phones, sensors and wearable, future application are expected to send a huge amount of data over the network. The data processing and transmission from the sensors/gadgets to reasonable DCs gets challenging. A new paradigm is required to transfer and process the upcoming abundant data gathered from geographically dispersed sensors/gadgets. FC is the best evolving paradigm to tackle stated issues. FC provides the centralized data centers by storing, filtering, preprocessing and caching data at the edge of the network.

FC provides low latency between the end users and their applications by reducing the utilization of long-distance network. FC can be used for some other purposes too including: content caching [45], computational offloading [46], privacy and security [47], service monitoring [48] and service management [49]. FC structure, on a very basic level is not quite the same as conventional cloud design: to keep up closeness with countless clients, fog resources must be scattered over a huge geological region, for example, a city, an area or even a whole country. Different emerging applications are presented every year through a hype cycle to analyze the significant patterns of these emerging and declining trends throughout the year.

The Gartner Hype Cycle [50] is a predictive diagram that demonstrates present trending exploration themes and their genuine stage in the research. The yearly hype cycle report exhibits a major trend examination [51] indicating the significant patterns of the year. In this examination, a decline can be seen in cloud and mobile information, however innovations, for example, the mobile Data Centers (DCs), digital working conditions and IoT are arranged around the pinnacle of the extended desires. Subsequently, FC and mobile data technologies have recently been well-examined and have gotten significant place in the market. The declaration coming about in view of this super trend examination is that these advancements are right now at the pinnacle of desires, where a rational assessment of the complete extent of the theme is challenging. This peak is then trailed by a presentation of related difficulties and recent issues to be taken care of until first appropriate arrangements are adequately looked into [52]. The emerging trends are shown through Gartner Hype Cycle [50] in **Figure 2.5**.

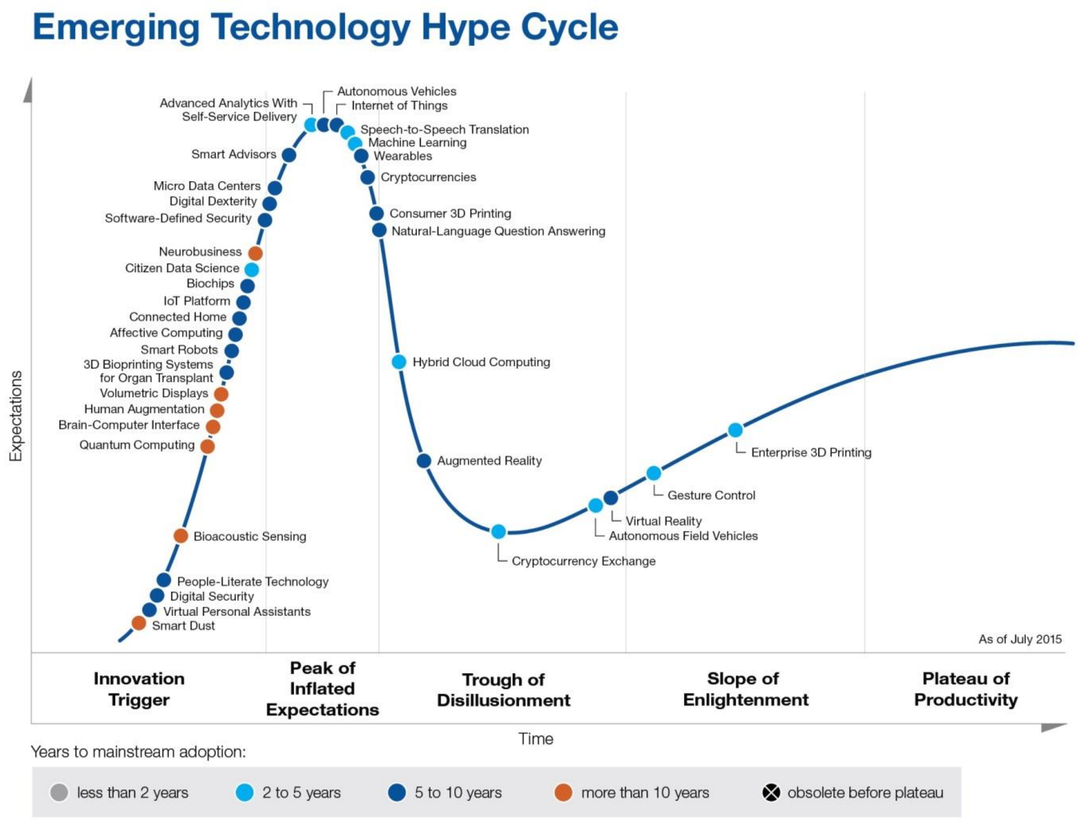

Figure 2.5: *Gartner Hype Cycle for emerging technologies*

2.3.1 Fog Computing Architecture

A distributed fog computing architecture proposed by CISCO is presented below. The FC architecture consists of three layers:

1. **User/IoT Layer**: The bottommost layer is user/IoT layer, closest to the end users and their physical environment. This layer contains IoT devices, street cameras, smart vehicles and mobile phones. The data collected from these devices is send to the upper layer, the fog layer, for further processing and storage. The devices of this layer also receive commands from the fog layer as a result to actuate their environment, depending upon the applications. These devices generally use available access network (i.e. WiFi, LoRa and cellular network) to connect to the upper layers.
2. **Fog Layer**: The middle layer between users/IoT devices and the cloud is the fog layer. A fog infrastructure aims to bring computational, storage and networking resources in the immediate proximity of its end users. This layer is composed of geographically distributed small groups of

servers known as fog nodes. These fog nodes are placed in strategic locations such as shopping malls, bus stations, streets, stadiums across a potentially large geographical area. Each fog node contains a small number of devices such as single-board computers [53], drones [54], vehicles [55] with limited computing and storage capacity. These devices can be either static or mobile [56, 57]. Fog applications that need computing, storage and network resources close to the end users are deployed in this layer.

3. **Cloud layer**: The top most layer is the cloud layer. This layer contains powerful datacenters connected to each other and to the rest of the internet with high-speed networks. It contains powerful computing and storage capacity to support applications which need extensive computational analysis and back-end storage, and deployed far from the end users. The basic FC architecture is shown below in **Figure 2.6**.

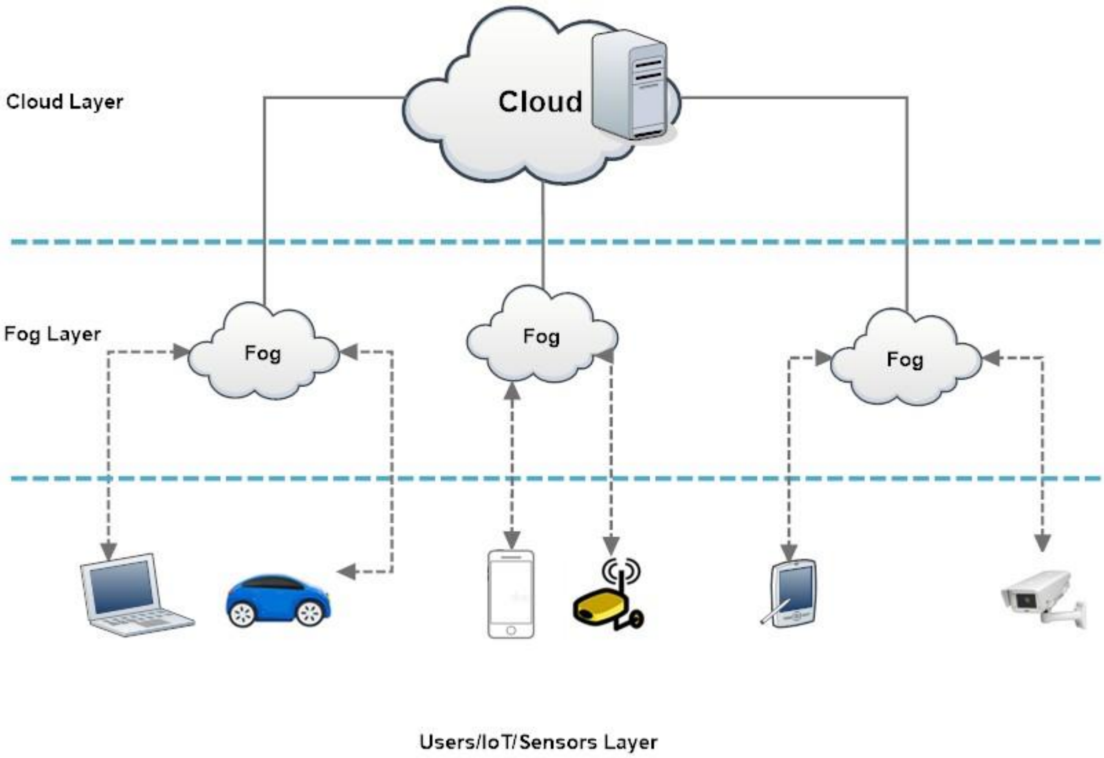

Figure 2.6:*A distributed Fog Computing architecture*

2.3.2 Characteristics of Fog Computing

The main attributes of FC paradigm, facilitating IoT to use its potential, are mentioned below:

2.3.2.1 Real-time Interactions

FC applications provide the real-time processing instead of batch processing. The real time processing means, provide fast services to user/edge devices near to the end devices.

2.3.2.2 Low Latency

In FC environment, task execution is done at fog cells (fog nodes) near to the edge of the network to reduce latency [62, 63]. The Fog play a dynamic role to gather and process the data near to the source.

2.3.2.3 Geographical Distribution

In FC, services and applications require a widely spread geographical distribution. FC plays a major role to provide the high-quality services to mobile vehicles through geographically distributed fog nodes by accessing their location at the highways/motorways and railways [62, 63].

2.3.2.4 Location Awareness

Due to location awareness of fog nodes services, fog devices store location coordinates. These services enable the features of filtering, preprocess and cache requested content to reduce the latency and service time of the process [62, 63].

2.3.2.5 Mobility

Along with the high distribution of IoT devices, it is important to consider the mobility of the participating end devices. Moving devices request dynamic transformation of the network topology. This can be possible via dynamic hierarchical fog architecture that allows to rebuild the topology. Going for a high mobility support, the FC architecture enables wireless communication with mobile devices [63].

2.3.2.6 Scalability

FC is elastic in nature due to its scalability. In FC devices, nodes or clusters can be added or removed as per the requirement of the system. The functionality of the FC can also be changed on the real time requirement, resources demand, load balancing requirement or network condition [62, 63].

2.3.2.7 Large-Scale Sensor Networks

Large-scale sensor networks communicating with fog nodes, is a significant scenario of the FC paradigm. These sensors can send task request to fog nodes. In case of availability of fog cell resources, the fog process data, otherwise this request is forwarded to other nodes for processing [62].

2.3.2.8 Device Heterogeneity

Different types of IoT devices [62] communicate with the network with no standardized functionality of communication between different types of IoT devices, network and deployment type. In fog architecture many types of requests of the IoT devices are handled in fog layer. Fog enables the communication between different types of IoT devices and resource virtualization.

2.3.3 Applications of Fog with the IoT

FC plays important role in various IoT applications. Below are some IoT applications that can benefit from FC.

2.3.3.1 Smart Traffic Lights

FC permits traffic signs to open streets by detecting flash lights. It measures the speed and the distance of nearby vehicles from the foot-travelers and cyclists. Sensor lights turn on when any movement is detected [64]. Smart traffic lights are considered as synchronized fog nodes to send alert messages to nearby vehicles. The roadside units, smart traffic 3G and WiFi improve connection between fog, vehicles and access points [65].

2.3.3.2 Healthcare

FC plays an imperative job in medicinal services. It gives real-time processing and responds to critical events in healthcare. [66]. Moreover, a reliable connection is required in healthcare IoTs to store, process and retrieve medical data from cloud. This reliable connection is not available in foresaid scenario. Fog also handles network connectivity issues and flow of traffic [67].

2.3.3.3 Smart Home

The IoT has numerous sensors and gadgets connected in home. However, these gadgets have different software and vendors, making it hard to get them cooperate and work together. Sometime they need more storage and higher computation. FC well tackled these issues. It incorporates every unique type of software and gadgets excellently in home applications with adaptable resources [68]. FC has numerous advantages for home security applications. It gives unified interface to coordinate with all gadgets. Likewise, it gives elastic resources to gadgets to enable real time processing, storage, and low latency [68, 69].

2.3.3.4 Connected Car

As indicated by Cisco, autonomous vehicles are considered as a new development for vehicles [64]. There are numerous advantageous features which rely upon the fog and internet connectivity that can be added to cars, for example, self-parking, hands free and automatic steering [64]. The prediction for next few years is that cars will have the ability to communicate with other cars through internet. FC will be the most effective answer to all Internet-associated vehicles, since it gives an elevated level of constant collaboration. In addition, it will permit vehicles, access points and traffic lights to collaborate with one another to convey a decent help to users [65]. With the utilization of the fog rather than the cloud, different mishaps and accidents can be avoided as latency is not an issue in FC as compared to CC [70].

2.3.3.5 IoT and Cyber–Physical Systems

The integration of the IoT with fog and cyber–physical systems (CPSs) is getting conceivable. The IoT network is able to interconnect devices from all over the world through the internet and telecommunication [6].

2.3.3.6 Wireless Sensor and Actuator Networks

Wireless sensors possess the ability to work on low power. Actuators are the modules of sensors to check the behavior/environment change. In FC actuators can be used as fog nodes to perform different actions to manage the end devices [8]. Wireless sensors consume low processing power, less energy and less bandwidth.

2.3.3.7 Augmented Reality (AR)

AR is a new system used to add virtual data to the real world [71]. AR is a significant application as computing devices become quicker, more ubiquitous and smaller. AR is a latency sensitive application, as a minor delay in the response time can affect the user requirement [72]. FC is the major player in the AR domain. It enables fog and cloud servers to provide services to real time applications.

2.3.4 Fog Node

In FC environment, fog node is the main component. Fog nodes are either virtual components or physical components (switches, gateways, servers and routers) that are connected to smart end devices and provide resources to these devices. Each fog node knows its logical location and geographical distribution in the cluster. In FC architecture, fog nodes provide services of data management between the end layer and the cloud layer. In FC paradigm, fog nodes act as server or a bridge. This scalability may be horizontal with hidden geolocation or mirroring mechanism [73].

2.3.4.1 Attributes of Fog Node

The essential characteristics of fog nodes that facilitate the deployment of FC capabilities are mentioned below:

- **Manageability**

Fog nodes are managed and composed by complex systems that can perform their operations automatically.

- **Programmability**

Fog nodes can be programmed at multiple levels by different stakeholders. These stakeholders include end users, domain experts, network operators and equipment providers.

- **Heterogeneity**

Fog nodes come in different form factors, and can be deployed in a wide variety of environments.

- **Autonomy**

Fog nodes are independent in making local decisions, at the node or cluster-of-nodes level.

- **Hierarchical clustering**

Fog nodes support hierarchical structures, with different layers providing different subsets of service functions while working together as a continuum [73].

2.4 Fog Computing VS Cloud Computing

The major differences between FC and CC are:

1. FC provides services near to the end users. In FC, services are available on the edge of the network for example base stations, switches, end user devices and access points. FC being at the edge of the network resolve the latency issue of the cloud. The characteristics list of FC is discussed in [74-76].
2. Fog paradigm is dependent on cloud paradigm and vice versa. Both paradigms depend on each other, their deployment and pros/cons are so different [9]. Fog paradigm is decentralized and cloud is centralized.
3. FC deployment is ad-hoc which requires less cost and planning since cloud deployment cost is quite higher.
4. CC latency is much higher than fog.
5. Maintenance cost of fog is less than cloud maintenance cost.
6. Each paradigm supports variety of applications; however, fog is more efficient for time sensitivity application. **Figure 2.7** shows the architectures of both paradigms.

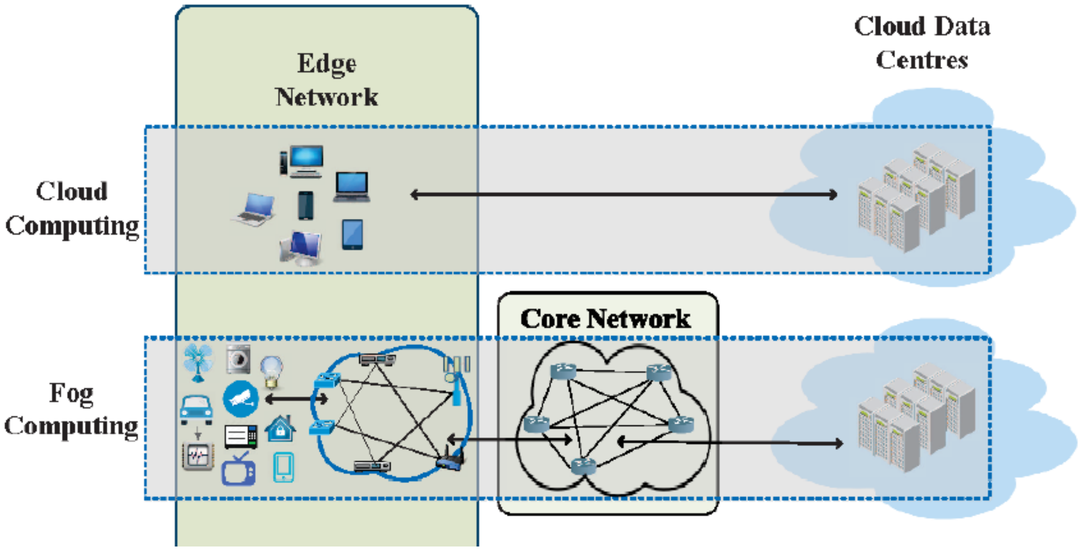

Figure 2.7: *Fog computing vs Cloud Computing architectures*

7. Fog's geographical coverage and resource management is local but in cloud it is global.

A clear bird view of comparison between cloud and fog is presented in the **Table 2.1.**

Table 2.1: *Fog Computing vs Cloud Computing*

Domain	Cloud Computing	Fog Computing
Network requirement	High-speed bandwidth, High-speed servers.	Do not need high speed, any device can act as a Fog
Applications	Require special type of application and these applications mainly suffer high latency.	It can be used in critical applications because of very low latency.
Operations/management	Require a special team to manage and operate in a fully controlled environment.	Operate according to user's requirement in their environment. It can be managed by any type of company.
Deployment	Require special planning for deployment.	Do not need special planning
Size	High cloud needs a large size of network at least thousands of servers.	Size of the network depends on user need and even every fog node can be a single server.
Network model	Centralized	Distributed and scattered in the geographical area.
Location/space	Small	Scattered in geo-graphical area.
Scalability	Scalable at center	Scalable of both center and fog

Table 2.2 shows the features of CC and FC.

Table 2.2: *Feature's comparison matrix of Cloud and Fog Computing*

Features	Cloud Computing	Fog Computing
Latency	High	Low
Connectivity	Dedicated line	Wireless
Service location	In the internet	Edge of the network
Hops between server and client	Multiple	One
attacks on en-route data	Higher probability	Very low probability
Location awareness	No	Yes
Mobility	Limited	Fully supported
Interaction for real-time	Supported	Very Supported
Distance between server and client	Numerous Hops	One hop
Service	Core architecture provide services	Edge of the network provide the services
Security	Undefined	Can be defined

2.5 Quality of Service (QoS)

QoS improves real-time reactions in time-sensitive tasks. To provide efficient services any issues regarding QoS should be handled. The FC paradigm was implemented to enhance QoS. QoS provisioning is the ultimate goal in general information systems, cloud services and in networking. QoS is defined in several ways depending on the perspective. In networking, QoS refers to: "any technology that manages data traffic to reduce packet loss, latency and jitter on the network" [9]. In general information systems: "QoS is the capacity to prioritize distinct applications, customers or information flows or to ensure a certain level of information stream efficiency". In CC, QoS is defined as: "non-functional properties of cloud services, which describe how well a service is performed, such as compliance, availability, reliability and responsiveness, price, security, latency [77]. The major parameters that define QoS include throughput, network use, transit delay, latency, availability, service time, priority, energy consumption and jitter. Considering the different definitions mentioned above according to multiple perspective, it is concluded that satisfactory QoS provisioning is the basic requirement to promote improved services [78].

2.5.1 QoS in Fog Computing

The main focus of FC is providing satisfactory QoS [79]. It is difficult to implement cloud computing QoS in FC paradigm due to some features of FC such as: distribution, heterogeneity and mobility. The main reason behind designing FC is to serve time critical applications with enhanced QoS. FC environment considers various QoS factors to design a successful system. Literature provides us with the following QoS factors: throughput, latency, service time, resource utilization, cost, execution time, energy consumption, network use, availability, scalability, and security [79-84]:

• **Throughput**: The maximum requested service rate that can be processed in the system.

• **Latency**: The term latency refers to any of several kinds of delays typically incurred in the processing of network data.

• **Service time**: The duration between the user's request and receiving the response.

• **Resource utilization**: The maximal utilization of available resources of a system.

• **Cost**: The amount of cost which a service applicant pays for computation, communications, or data storage in a specified period of time.

• **Execution time**: The duration in which a program is performed entirely.

• **Energy consumption**: The amount of energy consumed by a resource to perform a requested service.

• **Network use**: The amount of network bandwidth consumed to complete the requested task.

• **Availability**: The capability of a system to ensure that the requested resources with the expected performance are available.

• **Scalability**: The proficiency of a system to ensure the performance maintenance when system receives increased number of service requests.

• **Security**: This feature protects and secure the available data related to FC/CC environment.

In this research work, the proposed framework considers three factors of QoS for efficient service provisioning to the end users. These factors involve: latency, network use and service time.

2.5.1.1 Latency

The term latency refers to any of several kinds of delays typically incurred in the processing of network data. A low-latency network connection experiences small delay times, while a high-latency connection experiences long delay. Besides propagation delays, latency may also involve transmission delays (properties of the physical medium) and processing delays (such as passing through proxy servers or making network hops on the internet) [85].

In CC as the data centers are deployed far from the users, it may take large amount of time to execute any request. In time sensitive application, CC fails to make rapid responses. On the other hand, in FC environment the resources are deployed near to the edge of the network facilitating users with efficient services.

2.5.1.2 Network use

The second factor considered in this research for QoS provisioning is network use. Network use is referred as the amount of network consumed during processing a request. The larger distance between users and end devices increases the number of routers/hops which results in higher network use. Hence, real time provisioning of services is obstructed and QoS is decreased while leveraging remote fog nodes for the out sourced applications. Through the proposed framework the issue of network use is tried to resolve to provide satisfactory services.

2.5.1.3 Service Time

The third factor of QoS considered in this work is service time. Service time is the duration between user's request and the response time. By deploying fog head node in the fog layer in the proposed LAFF architecture, the framework become able to reduce the service time as all the fog nodes are controlled by fog head node. Most importantly the proposed LAFF algorithm registers users on fog head and the algorithm are fully aware of the user's exact location. Hence, these significant features of LAFF algorithm helps to reduce latency, network use and service time to provide efficient and satisfactory QoS to the end uses.

2.6 Literature Review

In a simplified structure, FC is characterized by a geographically distributed computing design, prepared with heterogeneous devices connected at the edge of the network. Authors in [86, 87] highlight the advantages gained from FC. An algorithm is developed and implemented in [88] which is based on local computing. Through this algorithm, workload of cloud and fog processing is reduced. However, the proposed algorithm only works with star topology. The authors talk about the terminology and current meanings of FC in [68], giving an increasingly conceivable definition of this idea. In a simplified structure, FC is characterized as a geographically distributed computing design, prepared with heterogeneous connected devices at the edge of the network. The advantages gained from FC are highlighted in [86, 87, 89, 90]. A lightweight fog-based framework is designed in [91] which highlights high productivity and constant strange data separating calculations. The Fog approach is suggested as an appropriate structure for IoT applications in [92] and proposes+ a Distributed Data Flow Programming Model as an explanation behind Fog based IoT applications. They analyze the middle requirements that Fog-based IoT applications need to meet and recognize different issues with existing approaches to manage Fog based application development. The limitation of this work is that they only focused on the fog stratum and addressed offloading and load redistribution. Two load balancing algorithms (task distributing and task grasping) are developed in [93-99] for large scale FC. Through this structure, load balancing overhead is reduced when the scale of fog is increased to get benefits of centralized and decentralized computing. IoT are scattered on geographical locations. The geographical positions of the sensor nodes may lead unbalanced loading of the fog. Load balancing is also a serious concern of fog because load disturbs services efficiency and latency. To resolve this issue, the proposed framework enables the load balancing feature among the fog nodes through a fog head concept, which increase the energy efficiency of the fog and reduce the latency.

The work done in [94] propose another Fog-2-Fog (F2F) coordinated effort model that presents offloading approach among fog nodes, as per their load and handling capacities, through a new load balancing known as (FRMS) Fog Resource Management Scheme. However, throughput and delay has not been evaluated. The idea of resource allocation in a fog environment is proposed in [95]. Authors present three tier architecture including, the user, cloud and fog layer that distribute the load between the cloud and fog nodes. Researchers also check whether required processing is available on the fog node, to allocate some tasks for execution or to postpone these tasks for time being. This work handles the challenges of heterogeneity but neither association nor scalability challenges are addressed. An integration of fog to cloud approach for job placement and movement is suggested in [96]. Although this work is promising but authors ignore the need for required memory for storing states when considering cloud-edge infrastructure even being a constraint of the devices. Authors in [97] solve the problem of load balancing in fog network through distributed technique by assigning IoT devices to appropriate fog nodes to reduce latency of all data flow. In this technique fog nodes broadcast periodically the computing and traffic load. The drawback of this research is that it introduces excessive exchange of control messages and computational load in the network A framework is designed in [98] for adding FC in medical field, with addition to scheme for resource management, jointly considering fog association, placement of VM and task distribution.

Authors purpose in [99] an algorithm for workload placement in tier edge cloud network to improve the response time of all tasks. The algorithm allots computing resources between different tiers of fog nodes for completing assigned task. The work done in [100] presents the idea of distributing workload of a fog server receiving a high traffic coming from IoT. Load balancing is implemented in fog layer through virtualization technique. Proposed IoT based Fog-To-Cloud and Data-in-Motion with load balancing (IoT-F2CDM-LB) architecture, for traffic distribution between virtual servers. However, authors divide the QoS level and use protocol as per QoS level requirement and this process increase the burden on system. To propose an ideal workload distribution framework in [101], a queuing theory is used to decrease delay and energy consumption. The proposed solution of this work divides a problem into three sub problems hence increasing the burden on the system. Authors propose a Real Time Efficient Scheduling (RTES) algorithm for load balancing in FC environment in [102]. The RTES's objective is to balance the load through available bandwidth, to execute tasks before deadline and reply the user with minimum time using by the fog layer dividing the data into multiple layers. These multiple copies of the data are a burden on network/memory.

A QoS based optimized load balancing algorithm for task scheduling is proposed in [103]. The tasks priority is evaluated on the basis of completion time and then the tasks are scheduled onto a resource. These resources complete tasks according to their priority. A novel optimized load balancing approach is introduced in [104]. This approach uses Particle Swarm Optimization method to transfer tasks from an over loaded VM to less loaded VM to balance the system load. However, the priority algorithm consumes more energy of the system. The work done in [105] focuses on the shortcomings of the existing cloud-based load balancing methods. These methods lack in terms of load prediction and system hierarchy hence cannot be implemented to P2P and dynamic approaches of FC. The dynamic FC load balancing optimization approach distributes tasks among the VMs according to the resource requirements. The resource requirements of the tasks are done through clustering and graph partitioning.

The Software Defined Cloud/Fog Networking (SDCFN) framework is designed for the Internet of Vehicles (IoV) in [106]. The framework helps to gather information for load balancing by allowing central control for communication between vehicles. To reduce latency and QoS provisioning the study [107] implemented Particle Swarm Optimization Constrained Optimization (PSOCO) for load balancing. This work provides less scalability. CooLoad approach presented in [108] is a cooperative load balancing approach for fog data centers reduce service delay time. A buffer is allotted to each fog data center to receive client's requests. A certain threshold is set for each buffer when the received requests reach above this threshold, the requests are transferred to the adjacent fog data centers for load balancing. This system is based on the assumption that for better load balancing the data centers are connected to a high-speed bandwidth. A dynamic load balancing algorithm for edge data centers is developed in [109] by using authentication method. By adopting authentication method, the load balancing algorithm finds authenticated data centers. Through Breadth First Search (BFS) technique, tasks are assigned to less loaded data centers. Data centers and their maximum capacity is estimated by the current load on the data centers. Finding the less loaded fog node is also a burden of the system. IoT resource provisioning issue is discussed in [72] and a solution is proposed to overcome this problem. The model aims to boost Fog resources and the minimization of system delay. The work in [72] is extended in [95] where QoS measurements and the deadlines for the provisioning of each kind of resources is considered. Other researches [110, 111] proposed that a task will execute either on fog layer or cloud layer. To reduce the over usage of fog resources, the history of previous arrival of requests is considered [110]. A FC framework collaborated with M2M communication protocol is proposed in [112]. The framework ensures QoS provisioning by transferring data with low latency and network use. In [113] authors proposed an algorithm with theoretical foundations for cost aware dynamic service provisioning and storage on clouds. However, this work does not consider the QoS and latency issues. Due to the limited resources, it is very difficult for cloud providers to provide all the requested resources from the cloud provider's perspectives cloud resources must be allocated in a fair and efficient manner. To guarantee QoS, resource management is used. In this study resource management is not the main theme but used to facilitate registered/unregistered users.

Fog model presented in [114] enables the resource and cloud to be used as edge resources. Fog Model brings the computing resources closer [67, 115] to the sensors and users. In [116] authors purpose an architecture based on Kubernetes for resource management to reduce latency. In [72] Fog-based framework is developed. The methodology used in this work concentrated on actualizing a Software Defined Resource Management layer at the Fog layer to serve IoT demands. Moreover, a resource provisioning module is incorporated which is in charge of serving choices dependent on measurements collected by a checking module. In [117], IoT resource provisioning issue has been discussed and proposed a solution to overcome this problem. The model measured the boost of Fog resources and the minimization of general system delay.

A resource scheduling method dependent on demand forecasts is presented in [118]. The theme of this work is to assign resources dependent on clients' interest changes by utilizing cost capacities, various kinds of services and valuing models for new and existing clients. The model accomplishes a reasonable exhibition by reallocating resources dependent on client conduct and prediction of utilization. In [119] a streamlining definition for the service arrangement of IoT applications in Fog environment is proposed and actualized as a model called Fog-Torch. Their work concentrated on hardware and software requests as well as on QoS requirement, such as bandwidth and network latency. Authors propose a latency aware application module for FC that fulfill the requirements of service delivery latency in [120]. It also considers the amount of data signals that are processed in per unit time for different applications.

The aim of this work is to ensure QoS provisioning to meet service delivery deadline by optimizing resources.

An algorithm for task management in fog infrastructure is designed in [121]. Their aim is to focus on task scheduling at the Fog layer while minimizing response time dependent on resource requested by these assignments. However, explicit QoS prerequisites are not considered in their methodology. In [122], service provisioning approach for integrated fog cloud designs has been planned. Their model highlights the minimization of network latency while ensuring legitimate service activity. A structure for WSANs applications is suggested in [123]. According to the proposed structure, gateways work as fog nodes. IoT layer consist of user's/IoT devices. In proposed structure, fog layer comprises of two layers. First layer is called master layer and the second layer is known as slave layer. Slave layer has micro servers and traditional gateways which work as fog nodes. Data flow and resource management are the responsibilities of the modules of this layer. The master layer has smarter and power full gateways acting as fog nodes. This layer incorporates with the modules to control functionalities. For the cloud layer, the authors did not elaborate the modules it includes.

The work done in [124] discusses the issue of compute resource sharing among the fog nodes to execute computational requests, while they especially focus around fog-enabled little cells in cellular systems. The authors target the shaping clusters of small cells, where each cluster represents a collection of little cells that offer resources for offloading mobile devices from their remaining workload. The goal of all this mechanism is to reduce cost of the power consumption. With that in mind, they plan their problems as an optimization issue. The aim of this work is to reduce latency for each user through clusters shaping, allocation communication and computational resources. By validating their work with other author's methodologies, the author demonstrate that their work can fulfil a higher level of client requests. Authors in [125] handles a similar issue focusing on mobile fog framework. They target CPU enhancement, data transfer capacity, and capacity sharing to process requests. Because of the heterogeneity of these resources, they map them into time resources to evaluate them in a similar unit. They independently study the two issues: Maximizing the total and expanding the result of utility capacities. They tackle their issues by utilizing convex optimization. The assessment of their methodology over three hubs with genuine estimations demonstrates that it permits lessening administration dormancy and prompts high-vitality proficiency to provide heterogeneity and the QoS.

Authors in [126] propose an architecture compromises of four layers. These layers include cloud and the IoT/user layers. Authors proposed forth layer for the IoT/user in the architecture. The fourth Layer has many sensor devices, the aim of these sensor devices is to forward the sensed raw data to the fog layer. Authors proposed second and third layer as fog layers in which many high performance and low power edge nodes are the part of the fog layer. Sensor nodes group act as an edge node. The components of this layer are responsible to perform analysis on the data within a time frame. A large number of intermediate computing nodes represent as a second layer. Every node of the second layer is connected to a group of edge nodes in third layer. The components included in this layer make quick response to control the infrastructure if any dangerous event is deducted. Second and third layer analyze data and send report to the cloud layer. First layer represents the cloud layer. Prediction and natural disaster detection are very high computing task which are performed at first layer. Performance results and real time interaction compared with cloud along with the proposed prototype. Power consumption and transmission bandwidth is reduced 0.02% during data transfer on the cloud.

A system named “DRAM” is proposed in [127] for resource allocation in FC to avoid both too low and too high loads. To achieve load balance, a fog resource allocation method is adopted to allocate resources dynamically. A fog-based resource framework is designed in [128]. This approach gives rights to users to select the resources independently and also consider cost and time to complete the tasks. In [129], a framework named FOGPLAN for QoS-aware Dynamic Fog Service Provisioning (QDFSP) is introduced. In order to meet low latency and QoS requirements of applications, QDFSP dynamically deploys application services on fog nodes, or the release of application services that have previously been deployed on fog nodes. However, different characteristics of wireless and wired fog nodes are not considered. Also, the framework is neither location aware nor fulfil the real-time requirements of IoT tasks. Authors in [130] designed a fog-based lightweight system to develop cloud gaming with high QoS. This system uses a three-layer model, including cloud, fog, and devices (e.g., desktops/smartphone players). A set of super nodes is considered, which is near to end users and is connected to the cloud. The QoS requirements are achieved through reducing latency and bandwidth consumption.

A service management technique is introduced in [131] as iHome for smart homes in the cloud. The work proposed a Service Oriented Architecture (SOA) to monitor home applications with real time responses. The performance of services in terms of CPU and RAM in iHome is evaluated. The results show that the real time responses can be returned under heavy burden of loading. The proposed system is tested under a limited number of physical appliances in a modular approach. However, many other important influential factors like cost, and energy consumption are not addressed. Also, the system does not consider the user management and network conditions. The proposed framework FATEH in [132] uses a three-layered architecture to improve QoS parameters. The first layer contains IoT devices and an agent node to collect data, the gathered data is then submitted to the next layer. The third layer consists of fog manager to efficiently process the request on smart fog nodes. The processing and storage of sensitive data is done at the third layer. The data coming from the fog manager is also processed at third layer. The drawback of this system is that, it does not take into account the network conditions and user management.

An algorithm for task management in fog infrastructure is proposed in [133] aiming to focus on task scheduling at the fog layer while minimizing response time dependent on resources requested by these tasks. In any case, explicit QoS prerequisites are considered in their methodology. In [134], authors target the shaping clusters of small cells, where each cluster represents a collection of little cells that offer resources for offloading mobile devices from their remaining workload. The aim of this work is to reduce latency for each user through clusters shaping, bandwidth allocation and computational resources.

Location based services (LBS) [135] become increasingly popular in recent years due to recent advancements in mobile computing. LBS refers to service provisioning through location-based information of users such as the geographic position. An analytical framework to balance latency and reliability by using optimal location of the virtual controller is proposed in [136]. In [137] authors propose a fog based ERDMS, a delay aware accident management system. The idea behind this system is to reduce the response and rescue time. To detect an accident, built-in sensors of the smartphones are used. The use of built-in sensors also reduces the overall cost of the system. The information about the location of the accident is collected through GPS and sent to nearest fog nodes to find a nearby hospital to provide immediate assistance to the victims. In [138] authors propose a solution to provide the facility to choose and instantiate the services. The proposal does not consider the node location and localized demands. In [91], authors propose a new layer, the Fog layer (computing) of resources which is closer to the edge of the network to provide location awareness.

In [139], web site performance optimization is automated by fogging at the edge servers. This idea explains the significance of edge location by giving dynamic and customizable optimization dependent on local network and the conditions of user's devices. WiCloud [140] is developed as mobile edge computing platform with OpenStack to improve location-awareness and to manage inter mobile-edge communication and data acquisition for an innovative service. Providing an acceptable level of QoS is an important issue in FC [141]. To design an efficient Fog-based system, various QoS factors are considered. Extracting from literature, eleven factors of QoS are defined i.e., latency, security, service time, availability, cost, energy consumption, resource utilization, reliability, execution time, deadline and scalability [142]. Moreover, latency is investigated as one of the important factors of QoS. A framework is required to ensure QoS provisioning without burdening a single resource and provide services near to the edge focusing above mentioned performance metrics. This framework needs to be more useful to reduce latency, service time and network use through user and location management by considering the network conditions. A lightweight LAFF is devised through this study to solve the foresaid issue.

LAFF has taken into account various IoT data requirements (Multi Media Data, Textual Data). Major emphases of the proposed framework is given to location awareness and knows the exact location of the users/ actuators. LAFF registers users on fog heads and employs K* heuristic algorithm [143, 144] to find the shortest path between users and fog nodes. Moreover, the algorithm also takes decision about fog head selection considering the requested data type.

The proposed work is compared with IFAM (Intelligent FC Analytical Model) [20] and TPFC (Task Placement on FC) [21]. In [20] an analytical model and reinforcement learning algorithm in an FC environment is introduced. This model aims to reduce latency among healthcare IoT, cloud servers and end-users. This study proposes a novel multitier fog processing system that provides IoT services. However, in this work the author did not consider user's location, and network condition. The other drawback of this research is that user's request for normal data is transferred to cloud to respond. The LAFF is better in terms that it considers user's location and network conditions. The framework also transfers both type of data, MMD and TD to fog to fulfill user's request. In [21] a context aware information-based approach ideally uses virtual resources accessible on the system edges to improve the presentation of IoT benefits in terms of response time, cost and energy decrease. The approach utilizes context-aware information including network conditions, location of IoT devices and service type to provide resources to IoT applications. However, the increase in the number of fog nodes and services cause exponential increase in time for problem solving. Relevant significant research work comparison is given below in **Table 2.3**.

Table 2.3: *Comparison of the related work.*

Citation	**Year of publication**	**Problem Addressed**	**Techniques**	**Strength**	**Weakness**	**Tools**	**Domain**
A. Yousefpour et al [128]	2019	QoS requirements for delay sensitive applications.	To meet low latency and QoS requirements of applications, QDFSP dynamically deploys application services on fog nodes.	Deploy and release application services dynamically.	Different characteristics of wireless and wired fog nodes are not considered and neither the location aware.	Dynamic Fog Service Provisioning (QDFSP) algorithm	IoT in FC
M.Q Tran [20]	2019	what is a suitable fog computing scheme where effective service provision models can	A context-aware information based system of services has been devised.	A task placement FC approach with multiple intelligent tiers and a context-aware task	The increase in the number of fog nodes and services cause exponential increase	real-world applications, namely, the intelligent transportation system (ITS)	IoT in FC

		be deployed		provision mechanism is used.	in time for problem solving.		
Shukla S et al [21]	2019	Healthcare IoT devices generate huge volumes of data. This large volume of data results in network congestion and high latency.	An analytical model and a hybrid fuzzy-based reinforcement learning algorithm in an FC environment	Proposes a fuzzy inference system with reinforcement learning and neural network evolution strategies for data packet allocation.	The author did not consider user's location, and network condition. And user's request for normal data is transferred to cloud to respond.	iFogSim (Net-Beans) and Spyder (Python)	IoT in FC
Q. Fan et al [97]	2018	Poor communication ability of WSN to cloud is a challenge.	Approximation and detailed routing Algorithm is used for sensors considering hops and energy consumption	Maximize throughput and minimize the latency with Energy Saving	In Fog layer sink act as a fog node which used more power.	DCF	IoT in FC
Han Z et al [146]	2018	Data transfer from IoT to cloud	Design a new protocol Peer assistant UDT based Data Transfer Protocol (PaUDT)	Improve data transmission and congestion control.	Work only p2p network	PaUDT protocol with P2P network.	IoT in FC
X. Xu et al [154]	2018	How bottlenecks, resource efficiency, low load, and overload	Proposed a dynamic resource allocation method, named DRAM.	Proposed method aims to achieve high load balancing for all the types of	Load balancing for each type of computing node is achieved through	Dynamic resource algorithm	IoT in FC

		can be avoided.		computing nodes in the fog and the cloud platforms.	different algorithm which is a burden on the System		
Ramírez W et al [144]	2017	Dynamic Service Execution (DSE) problem in Fog to cloud scenarios	Proposing two basic resource allocation strategies, First-Fit and Random-Fit	improve all key metrics Service Response Time, Power Consumption, Network Bandwidth and Service Disruption Probability		First-Fit and Random-Fit	IoT in FC
Yousef pour et al [145]	2017	Service delay in IoT and Cloud	Proposed Framework place between cloud and IoT and design an analytical model for minimizing the service delay	Analytical model can support other policies of FC.		Distributed and centralized mode of communication	FC
Munir A et al [86]	2017	Energy efficiency, latency, performance, response time and mobility are the problems in Fog	Develop a novel integrated fog cloud IoT (IFCIoT) architectural paradigm	Improve energy efficiency, reduce latency, increase performance and response time with mobility management		integrated fog cloud IoT (IFCIoT)	IoT in FC
Adhatarao, S et al [70]	2017	IP do not fulfil the need to integrate	Information-Centric Networking (ICN)	Extend the Availability and	Use Cases describe the functional	Named Data Networking	WSN in Fog computing

		Sensor Networks with the Internet.	based FOGG computing Gateway	control of the sensor through intelligent data processing.	ity of FOGG.	(NDN) and Information-Centric Networking (ICN)	
Bhargava K et al [148]	2017	Addresses the problem of localization of Ambient Assisted Living (AAL).	Present a fog enabled WSN system with Edge mining technique.	Due to edge mining technique sensors send the data according to a predefined format to the cloud.	Consider the constant speed of the user.	Iterative Edge Mining: IEM and Genetic Algorithm.	WSN in FC.
Zhang G st al [150]	2017	How to computation-intensive tasks offload effectively from resource-constrained devices.	Proposed a data acquisition mechanism for clustering WSNs.	Even in case of data abnormality proposed approach enables reliable and efficient data acquisition.	Data actuations decrease the performance	Unusual data filtering Algorithm and Suspicious data detection Algorithm.	WSN in FC
Skarlat, O et al [132]	2017	How resource allocation and latency reduction can be managed in FOG	Proposed a FC framework using Multi user Multiple Input Multiple Output (MU-MIMO) technique	Given efficient plan of chunk size, order of delivery, nodes, and channels under the minimized latency.	Partitioning problem needs a little more work to be inserted into the Genetic algorithm (GA) framework	Genetic Algorithm (GA) approach with order transmission and partitioning.	IoT in FC
Lin Y et al [129]	2017	As the graphics rendering is offloaded to the	Proposes a lightweight system, which incorporates super	Propose the reputation based super node	Only best perform in video streaming	PeerSim and PlanetLab	Gaming in Fog/Cloud

		cloud, the data transmission between the end-users and the cloud significantly increases the	nodes that are responsible for rendering game and videos streaming	selection strategy.			
Yi, S et al [68]	2016	Design goals and platforms are the challenges of fog computing	Present the design and implement a fog platform	Performance in term of response improved	Present the prototype platform	Open Stack modules	IoT in FC
Naranjo PG et al [147]	2016	In WSN, Cluster head selection is the main problem due to energy saving.	Proposed a FOG supported sensor network using a new Stable Election protocol.	Proposed algorithm to save the energy and maximize the network life.	Use high power sensors as a gateway.	New Stable Election Protocol(N-Sep)	WSN in Fog computing
Aazam M et al [151]	2016	How data can be uploaded with different frequencies on the cloud without extra burden on core network and cloud.	Proposed fog based efficient resource management framework.	Resources manage through probability of resource utilization and user characteristic.	For resource management, characteristics of user should be known.		IoT in FC
N.K. Giang et al [149]	2015	Unnecessary communication is a burden on the core network and the data center of the cloud.	Proposed a smart gateway for Data preprocessing and trimming according to the format.	Based on the application feedback, Gateway must decide the time and type of data to be sent.	Only Suitable for mobile objects and large-scale IoT/WSN.		WSN/IoT in FC

Daniluk K et al [152]	2015	Provide better user services, reduce latency and enable real time big data analytics.	Proposed a FC layer for WSN in order to manipulate more efficient way with different energy states.	Support real time analytic process filters the data and send to the cloud.	Implement on very small scale for experiment.	Wireless sensor nodes working on shallow sleep, deep sleep, awake algorithms.	WSN in FC
Daniluk K et al [153]	2015	How application deadline can be met and energy saving during communication of cloud to things.	Develop a mathematical model of a three-tier cloud of things to access the applicability of the fog.	Cloud operations perform in the fog for saving energy and provide the data in real time.	Cooperation of different entities of different tiers is not discussed.		IoT in FC

2.7 Summary

In this section the analysis of the literature review is presented. Firstly, the related literature is reviewed regarding IoT, Load balancing, resource allocation, latency, network use and QoS. IoT is still in its evolution phase, and correspondingly several architectures are addressed in the literature. By reviewing the available related work, it is found that, to improve the QoS for users in terms of reduced latency, network use and service time, there is a need to develop a lightweight fog-based framework. In second part of literature, load balancing of fog is explored and concluded that load balancing optimizes the use of resources available, maximize throughput, minimize response time, and avoid overload of any single resource. Instead of using a single component, load balancing takes advantage of multiple resources. From the third part of the literature, it is concluded that QoS can be improved through the efficient management of the resources. The gaps identified through the literature are: resources management does not consider location of the user and requested data type, the second gap is, if any service providing node become hard to reach due to any abnormality, the service delay would increase.

It is the need of hour to design and develop lightweight framework in fog environment to provide QoS through location management. The literature review, here, provides us the ground for planning further research to explore QoS provisioning methodologies.

CHAPTER 3

LIGHTWEIGHT LOCATION AWARE FOG FRAMEWORK (LAFF)

3.1 Overview

In this chapter the lightweight LAFF is explained that empowers the capabilities of FC. Required features, physical topology of the simulation, architecture of the framework, physical and logical components are described in detail. The architecture of proposed Lightweight LAFF is elaborated. Characteristics, components and features of the framework are also discussed. The working of the proposed framework is described stepwise. The analytical model of the lightweight LAFF is discussed with mathematical equations. The proposed algorithm and its flow diagram are also presented.

3.2 Location Aware Fog Framework (LAFF)

The main focus of this research is to develop a lightweight fog-based framework which aims to provide better QoS to users in terms of reduced latency, network use and service time. The proposed framework's goals can be achieved through the user and location management. LAFF provides reliability, availability and extensibility. The traditional fog architecture is used in this framework by modifying the architecture through adding some modules according to the requirement of proposed LAFF. A fog head is added to the fog layer along with a dynamic algorithm for user registration and location management.

3.2.1 Characteristics of Lightweight LAFF

The lightweight LAFF possess the following characteristics:

- The framework employs the concept of fog head node that keeps track of other fog nodes in terms of user registration.

- The fog head node responds to the users by considering the requested data type MMD or TD. Fog head node assigns a dedicated fog node to the user to accomplish the request
- The framework is location aware with addition of load balancing, cloud agent, service management, offloading management and K* algorithm modules.
- Fog head helps to reduce service time to fulfil user's request by communicating to the fog nodes.
- The proposed lightweight LAFF persistently satisfies QoS by providing an accurate location aware algorithm.
- A comparative analysis is also presented to analyse network usage, service time, latency, RAM and CPU utilization to prove that LAFF is a lightweight framework.

3.3 Architecture of LAFF

The traditional three-layered fog architecture is the base to develop the lightweight LAFF. The top layer of lightweight LAFF is CC layer which coordinates with lower layers for data collection and storage for future use. Cloud layer can be used for data processing and storage for a large amount of the data for longer duration. In terms of failure of the fog head, cloud provide services to end devices or users.

The intermediate layer of the framework is fog layer. FC is ultimately the addition or extension of CC in terms of architecture. It covers the certain issues like high latency, network usage, resource management and centralization issues of cloud. A fog head is added in the fog layer to make the system lightweight. When a central fog node (fog head node) receives a request from the user, the fog head takes decision on the basis of data type and services module of the fog as defined in LAFF algorithm. In the lightweight LAFF framework fog nodes communicate with each other through central fog node (Fog head). Fog heads are spread over the globe and managed by the proxy server (Location management). The LAFF architecture is presented in **Figure 3.1**.

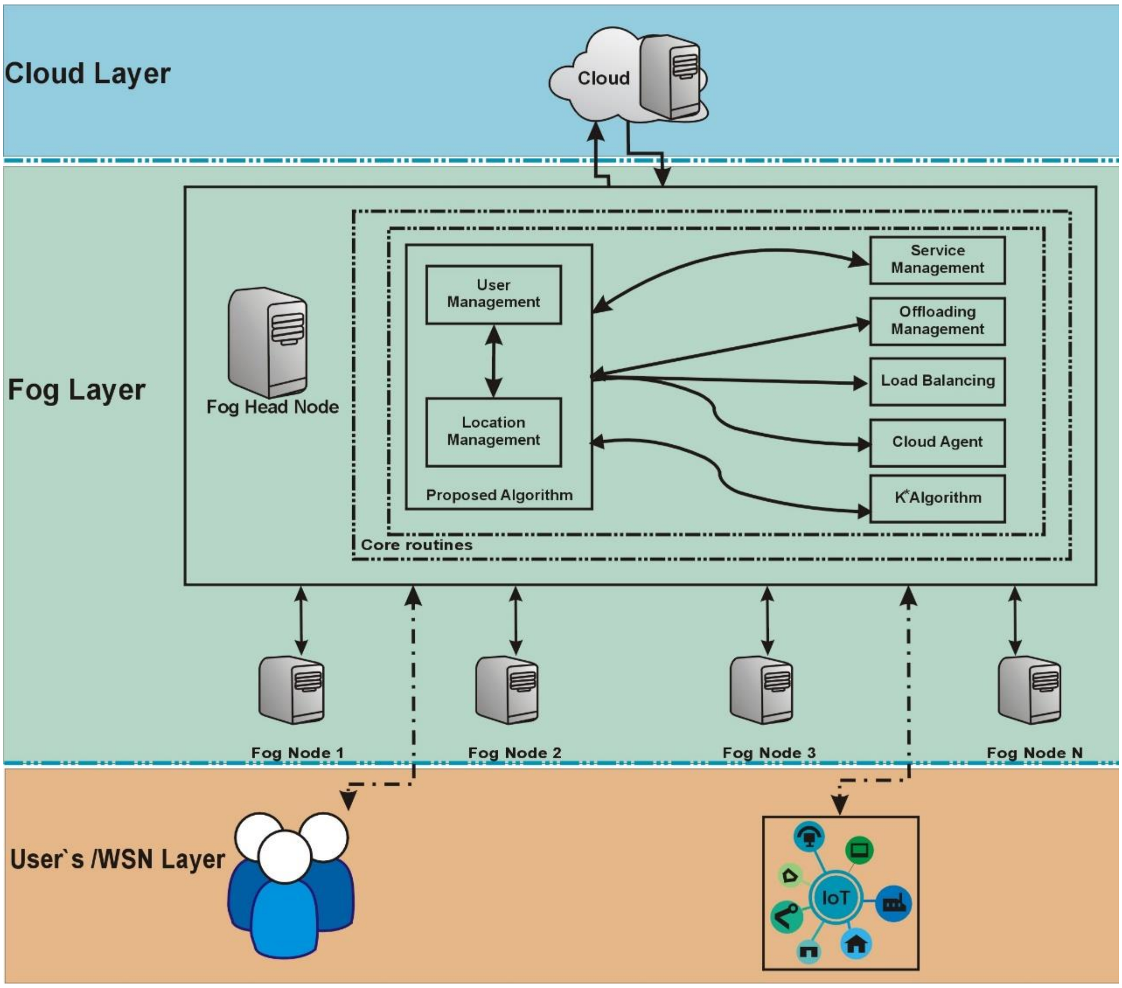

Figure 3.1: *Architecture of LAFF*

3.3.1 Components of LAFF Architecture

The following section explains the components of the architecture of lightweight LAFF framework:

3.3.1.1 Cloud Layer

The top layer of lightweight LAFF is cloud layer which coordinates with lower layers for data collection and storage for future use. Cloud layer is used to process and store large amount of the data for longer duration. If the fog head fails to provide services to the users, then cloud facilitates the users. Cloud layer component is:

Cloud: Cloud is placed at higher layer of the lightweight LAFF. Cloud facilitates the fog layer in terms of storing data for later use and higher processing when needed. Cloud layer plays a supervisory role to handle communication, processing and storing large amount of the data for longer duration. Cloud servers are independent units act as the centralized hosts and possess all necessary software they need to run. Cloud storage has many distributed resources acting as one unit. This distribution of data makes the cloud very fault tolerant. In this work cloud is connected to the fog head to communicate with all fog nodes through cloud agent.

3.3.1.2 Fog Layer

Fog layer is the middle layer of the lightweight LAFF which aims to provide the processing facility of the data near to the edge. The following sections explain the modules of the fog layer:

Fog head: Fog heads are fixed and predetermined physically with respect to geographical region and have lager hardware resources. Fog head is deployed between the fog nodes and the cloud and is responsible to communicate with cloud and all fog nodes. Fog head works according to the devised algorithm to access user's location and to identify requested data type. Users are registered at fog head. Fog head knows the exact location of all fog nodes. Tasks are assigned to the nodes considering requested data type. Fog head is also responsible to manage and maintain the information on hardware level. Fog head nodes have the following helping modules and the proposed algorithm calls these modules as per their requirement.

User Management Module: Registration or management of users and storing their details for future use is the responsibility of the user management module. The registered users are stored in Hashmap against specific identifiers for fast-track communication. The advantage of using Hashmap is that it is not synchronized, hence, saves additional usage of network and service time. The user management module communicates with the location management module to update/get user's location to provide services.

Location Management Module: Location management module manages the location of the users. The location is configured with coordinates. The x and y are range coordinate variables that are used for finding the shortest path in K* search [143]. The coordinates from 10 to 50 identify the location of existing users while other coordinates (coord1, coord2) contain new users and n represents the coordinate value range. The mathematical representation of Geo function is described below in equation (3.1):

$$\forall\, x \cup y \exists\ Geo(coord1, coord2)\ \Delta\ coord\theta < n\ \cap\ coord\theta > n \qquad (3.1)$$

$$\boldsymbol{where}\ 10\ \leq n \leq 50$$

On each request of the user, the location management module is accessed to match/update the location management table.

Service Management Module: Service Management Module (SMM) provides services to the fog layer. SMM registers services and coordinates with fog nodes to provide services to users. It manages the fog nodes services delivery assurance. SMM monitors all the resources of the nodes and fog head.

Offloading Management Module: Offloading management module offloads the task from a fog node and assign to other nodes to provide dedicated services to assure QoS. Through this module, the framework enables to offload the task from a fog node and dedicate to the user.

Cloud Agent Module: Cloud agent module facilitates the fog head and cloud to communicate with each other for storing and updating data on the cloud. Cloud agent module works as a broker between the fog layer and the cloud layer.

Fog Nodes: Fog nodes work as a server of the geographic area in which the fog node is deployed. Fog nodes process data near to the edge to reduce the burden of the cloud. Through the fog nodes, the lightweight LAFF provides better QoS by reducing latency and service time to accomplish the request.

K* Algorithm Module: Through the proposed algorithm user's location is accessed and nearest fog node is assigned to the user to fulfill the request. If this nearest fog node is hard to reach due to any abnormality, the LAFF algorithm uses a heuristic search algorithm k* [120], which is used to find shortest path between users and fog node. The vertices in this case, are added between registered and un-registered users. The advantage of using K* algorithm is that it only uses executed portion of the graph. It reduces the network usage by only working on a required portion instead of communicating to whole weighted graph. The complexity of K* algorithm is O (nlogn + ru + uu) where n is the number of vertices. In [143] the author used the same k* algorithm to find shortest path between source and destination.

3.3.1.3 User/IoT Layer

The bottom layer of the lightweight LAFF is user layer in which all sensor nodes and users exist. Users interact with the upper layers to get response for their request. Sensor/IoT nodes also communicate with the upper layers for reducing the latency and increasing the life of the sensor's nodes. All the data is generated at this layer from an enormous number of IoT devices and transferred to the upper layers for further processing and storage. The simulation has fractional selectivity model adds mobility of the IoT nodes and make themselves dynamic on IoT layer.

3.3.2 Physical Topology

The physical topology shows the pattern of nodes and devices in the network. Physical entities are created, and their competence, capability, and configurations are specified. These entities include sensors, actuators and gateways. The links between these entities and their configurations are also established. Physical network topology is important to understand the pattern of the network, how various network devices are organized, and how they communicate with each other. These configurations and capacity determine the load a network can tolerate and the amount of data it can transfer. In **Figure 3.2** the physical topology of the setup is described.

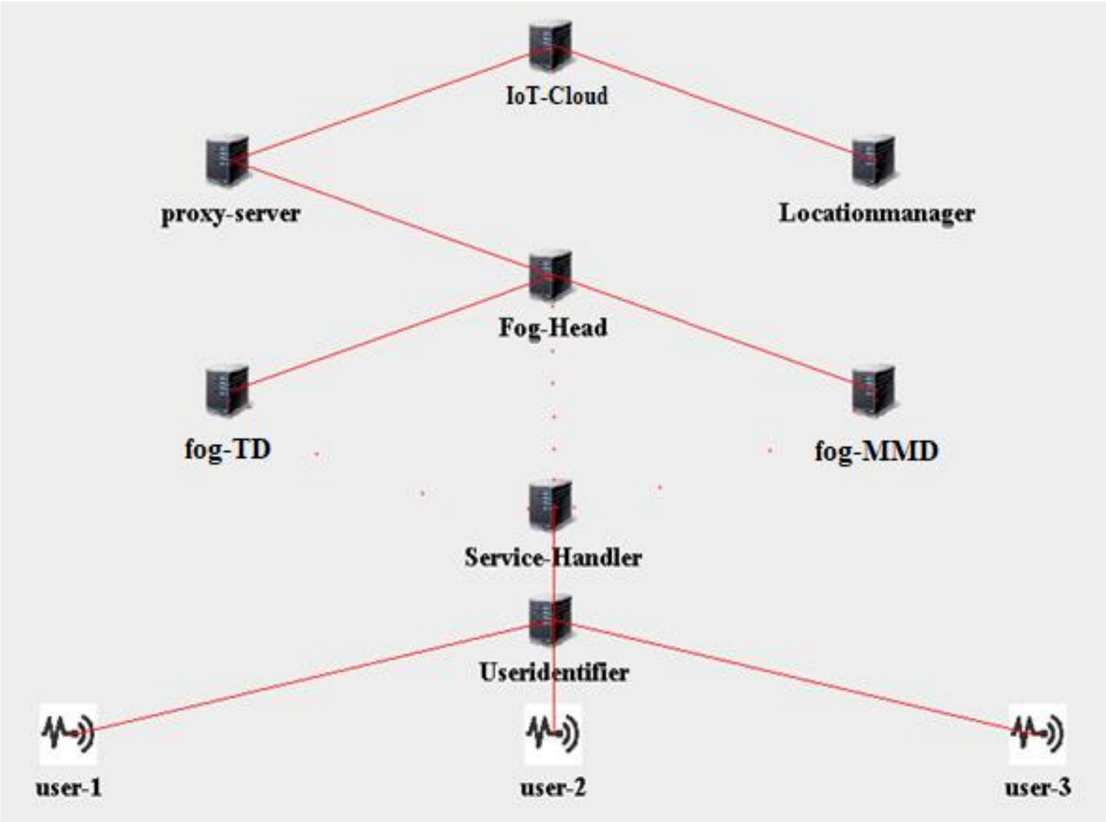

Figure 3.2: *Physical topology of simulation setup*

Components of the proposed system are as follow:

3.3.3 Physical Components

Physical components of the lightweight LAFF include Cloud /Fog devices (Fog nodes). The Fog devices coordinate in hierarchical order. The bottom level fog nodes are directly linked with associated actuators or sensors. Fog devices offer networks, computational resources and memory in CC, and play the role of datacenters. The sensors in iFogSim generates tuples that are referred as tasks in CC.

3.3.3.1 Cloud Server

Cloud servers are the centralized hosts while fog devices are the geographically distributed hosts. Cloud servers have all the required software to run and can act as independent units. Cloud server plays a supervisory role to handle communication and data storage. Cloud storage have many scattered resources functioning as one unit. This distribution of data makes the cloud very fault tolerant. In this work cloud connects to fog head to communicate with all fog nodes.

3.3.3.2 Fog Devices/Nodes

Fog nodes are mini cloud, situated at the edge of the network. These fog nodes are implemented through a variety of edge devices, interconnected by a variety, mostly wireless communication technologies. Thus, a fog node is an infrastructure implementing the said mini-cloud. As latency is minimized with location management so that the hosts must be distributed geographically over the globe near the sensor devices or users.

3.3.3.3 Sensors

Sensors are charged coupled devices and paired metal-oxide semiconductor imagers, used for light data transmission. The sensors used for heavy (video) data transfer, include passive infrared, ultrasonic and microwave sensor.

3.3.3.4 Actuators

An actuator is a device that transforms an electrical signal to a physical output. Actuators are used as displays the output. These can be the screens or areas where sensors are attached.

3.3.3.5 Gateways

Gateways are those devices in the network that connect sensors with nearest hosts/Fog nodes. These gateways are sits between IoT devices, sensors, Fog Nodes and Cloud. Gateways performs several critical functions from translating protocols to encrypting, processing, managing and filtering data.

3.3.3.6 Physical Links

The physical links contain the data speed for up and down transfer between IoT devices, sensors, fog nodes and cloud. The links are the connection between the communications devices to transfer the data.

3.3.4 Logical Components

The logical components of the system are responsible for the state of the art simulation. The described gears are module driven components. There are two logical components of iFogSim; Application modules (AppModules) and Application edges (AppEdges). List of logical components is given below:

3.3.4.1 Fog Controller

Fog controller class is a storage device that automatically transfer data from one premises storage to fog storage. Fog controller class control the all activities during the data transfer to fog storage.

3.3.4.2 Cloud Controller

A cloud controller class refers to a storage appliance that automatically moves data from one premises storage to cloud storage. Cloud controller class control the all activities during the data transfer to cloud storage.

3.3.4.3 Module Placements

Module placement class is a strategy or a program consists of interconnected modules. These modules run on fog data centers close to the devices to reduce the response time. It also helps to prevent excessive and, in some scenarios, unnecessary data transfers to the cloud.

3.3.4.4 Module Allocation Policy

Module Allocation policy class aims to ensure applications QoS to satisfy service delivery deadline and improve resource usage in fog environment. Through this class it is ensured for the deadline to meet for provisioning of QoS to the users.

3.3.4.5 Topology Creator

Topology creator class is the graphical/coding feature of iFogSim for creating the topology simulation. Through this class all topology in the simulations is created and managed in iFogSim.

3.3.4.6 App Module

App Module classes are a collection of inter-dependent AppModules and support the idea of distributed application. These types of classes are coordinate each other to run the functionality of the classes.

3.3.4.7 IoT-Cloud

IoT-Cloud class plays a supervisory role to manage the Fog and lower layer. IoT Cloud provides the facility of a data center to fog head/Fog nodes. The fog node/head communicate the IoT-Cloud layer for data storage permanently.

3.3.4.8 Fog Broker

Fog broker is a unit that manages the performance, delivery of fog services, use and relationship between fog providers and fog consumers. Fog broker class work as a bridge between the fog providers and fog consumers to ensure the performance and delivery of all type of services.

3.3.4.9 Datacenter

Fog provides the facility to store and process the data between the Cloud and the data producer. This facility is provided through the datacenter class. This class exist in the middle layer of the framework.

3.3.4.10 Network Topology

Network topology class provide the facility to manage the topology in the simulation environment by providing the connection and all other configurations. These classes can be found in Project files.

3.4 Working of LAFF

In the proposed LAFF, location awareness under fog computing umbrella, is introduced to reduce latency, service time and network usage along with minimal resources utilization. LAFF employs a location-aware algorithm that has ability to trace user's exact location through Fog head. Fog head is the controller of data centers of all fog nodes. The idea of Fog head is used in Fog computing technology [156]. The Fog head node is not only limited to search for current nodes but also for new nodes ($\mathbf{F_{head} \rightarrow F_{MMD} + F_{TD} + F_{others}}$). $\mathbf{F_{head}}$ represents fog head node, $\mathbf{F_{MMD}}$ refers to Fog Multi- Media-Data node, $\mathbf{F_{TD}}$ is fog Textual-Data node and $\mathbf{F_{others}}$ are *nth* new fog nodes. The search radius of Fog head ($\mathbf{F_{head}}$) is extended to n^{th} new nodes as the framework is developed by keeping the idea of scalability as well. After accessing the user's exact location fog head dedicates a nearest fog node in response to the user's request considering requested data type. If any nearest fog node is hard to reach then the k* algorithm is used to find the shortest path from user/actuator to fog node by estimating the coordinates [143, 144]. This dedicated node serves the user without any interruption. This framework also registers users (user management) and determines the requested data type. TD requests include text-based information or images etc (fog-TD servers handle these data types). MMD requests include videos and movies etc (fog-MMD servers handle these data types). The lightweight LAFF reduces latency L_{ld}, service time f and network use $\mathbf{ú_{nw}}$.

3.5 Features of LAFF

To minimize the service delay, fog head communicates to fog nodes and queries are processed on a short distance in this way service latency is minimized. If queries are not communicated through fog nodes and transferred to the upper layer like fog head and cloud, then the service delays are at a larger value. The latency L_{ld} is calculated by dividing available time $T_{available}$ with total time T_{total}

under product of 100.The equation (3.2) represents the formula for calculating latency.

$$L_{ld} = T_{available} / T_{total} * 100 \text{ (milliseconds)} \quad (3.2)$$

IoT service delay-minimizing policy: Policy adopted in this regard is to implement a minimum delay tolerance system. The values are considered to be very low as compared to other systems' latency. Latency, network use and service time is reduced by using equation (3.1).

If a fog node is hard to reach, the LAFF uses k* algorithm to find shortest path between users and the IoT devices. The path is selected from a pool of fog nodes (F_1—F_n) to have idle space for processing in order to provide better QoS. The list of fog nodes F= {F_1, F_2, F_3... F_n} and users U= {U_1, U_2, U_3... U_n} having tasks T for updating the Cloud C is represented by the equation (3.3).

$$Un \prod (Fn, T) \rightarrow C \quad (3.3)$$

Equation (3) represents the n array product from completion of task to update the Cloud.

The remaining components of the proposed paradigm are mathematically defined in analytical model. In the Analytical Model, the mapping between the components of different layers is discussed. In the simulation, the analytical model is implemented in iFogSim.

3.6 Analytical Model

A set (S) T_{iot} for all sensors S {S_1, S_2, S_3...S_n} and actuators A {A_1, A_2, A_3... A_n} under a tuples load α with transmission time L' is defined. Events E {E_1, E_2, E_3… E_n} happen at sensors {S} where n is the nth mapped sensor to an event E. Equation (3.4) and equation (3.5) represents the events that happened at fog °F and cloud ℃ through sensors.

$$°F: \boldsymbol{T_{iot}\{S_n, E_n\} * \alpha \rightarrow A_n} \quad (3.4)$$

$$℃ \nVdash \boldsymbol{G}: \boldsymbol{T_{iot}\{S_n, E_n\} * \alpha \rightarrow A_n} \quad (3.5)$$

The increase in the number of hops increase the latency, network usage and service time. The mapping of a sensor to a fog node is described in equation (3.6) that expresses the relationship between transmissions. Here $\boldsymbol{i}$ is IoT device number, $\boldsymbol{j}$ represents the column of devices where IoT device are mapped and **M** is mapping. **L** is load (MMD or TD load), **S** is sensors and F is fog node.

$$\boldsymbol{M}(\boldsymbol{i},\boldsymbol{1}): \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \quad \boldsymbol{j} = \boldsymbol{1} \text{Ľ} * \boldsymbol{Si} \rightarrow {}^{\circ}\text{F} \tag{3.6}$$

The latency L_{td} is computed using equation (3.2).

The service time *f* is expressed in terms of time taken by a service provider SP {SP_1, SP_2, SP_3 ... SP_n} by providing a service Ť to a user(s) ú {U_1, U_2, U_3... U_n}. The mapping relation is explained in equation (3.7).

$$\text{f}: \boldsymbol{SP} \rightarrow \text{ú} \prod \text{Ľ} * \text{Ť} \tag{3.7}$$

To calculate the service time ***f*** in simulation environment, equation (3.8) is used.

$$\text{f} = C_{ins}(T_{ms}) - T_k(T_t)\ (ms) \tag{3.8}$$

Where $\mathbf{C_{ins}(T_{ms})}$ represents the time in milliseconds fetched by calendar instance and $\mathbf{T_k(\ S_t)}$ is the simulation time stored by Timekeeper class. The simulation time is the amount of time spent in processing the K* search, allocating nodes, processing requests of users and updating cloud related to processing. In order to calculate network usage $\mathbf{ú_{nw}}$ the Tuples $\mathbf{T_{ud}}$ captured by network usage monitor $\mathbf{M_{nu}}$ are added to the total bandwidth used $\mathbf{B_u}$ in transmission and then divided by maximum simulation time $\mathbf{ST_{max}}$. Equation (3.9) is used for calculation:

$$ú_{nw} = M_{nu}\ (T_{ud}) + (B_u / ST_{max})\ \ (Kbps) \tag{3.9}$$

3.6.1 Proposed Algorithm

The algorithm and flow diagram are described below:

Algorithm 1: LAFF

Inputs: Tasks T, Start services S, User u, Geo (coord1, coord2) Gets integer based coordinates.
Output: Assign Nearest Fog-MMD or Fog-TD to the user.

```
start;
submit tasks;
place operators;
start services;
while allusers do
    getlocation;
    if coord>10 And coord<50 then
        existing reg user;
    else
        reg as new user;
    end
    if reguser then
        if multimedia data then
            if clocation==plocation then
                if hard to find then
                    start K* search;
                    calculate tasks on nodes (Fn * Tn);
                    offload data from nearest fognode (F*T-1/T);
                    allocate fog-MMD;
                    F(u, T);
                else
                    find nearest fog node;
                    Search(F1–>Fn);
                end
            else
                register location;
                Geo(coord1,coord2);
            end
        else
            transfer to fog-TD;
            Fl(uu,T);
        end
    else
        unreg user;
    end
    find idle fog node;
    uu(F1–>Fn);
    send to cloud;
    uu(C,T);
    repeat;
end
```

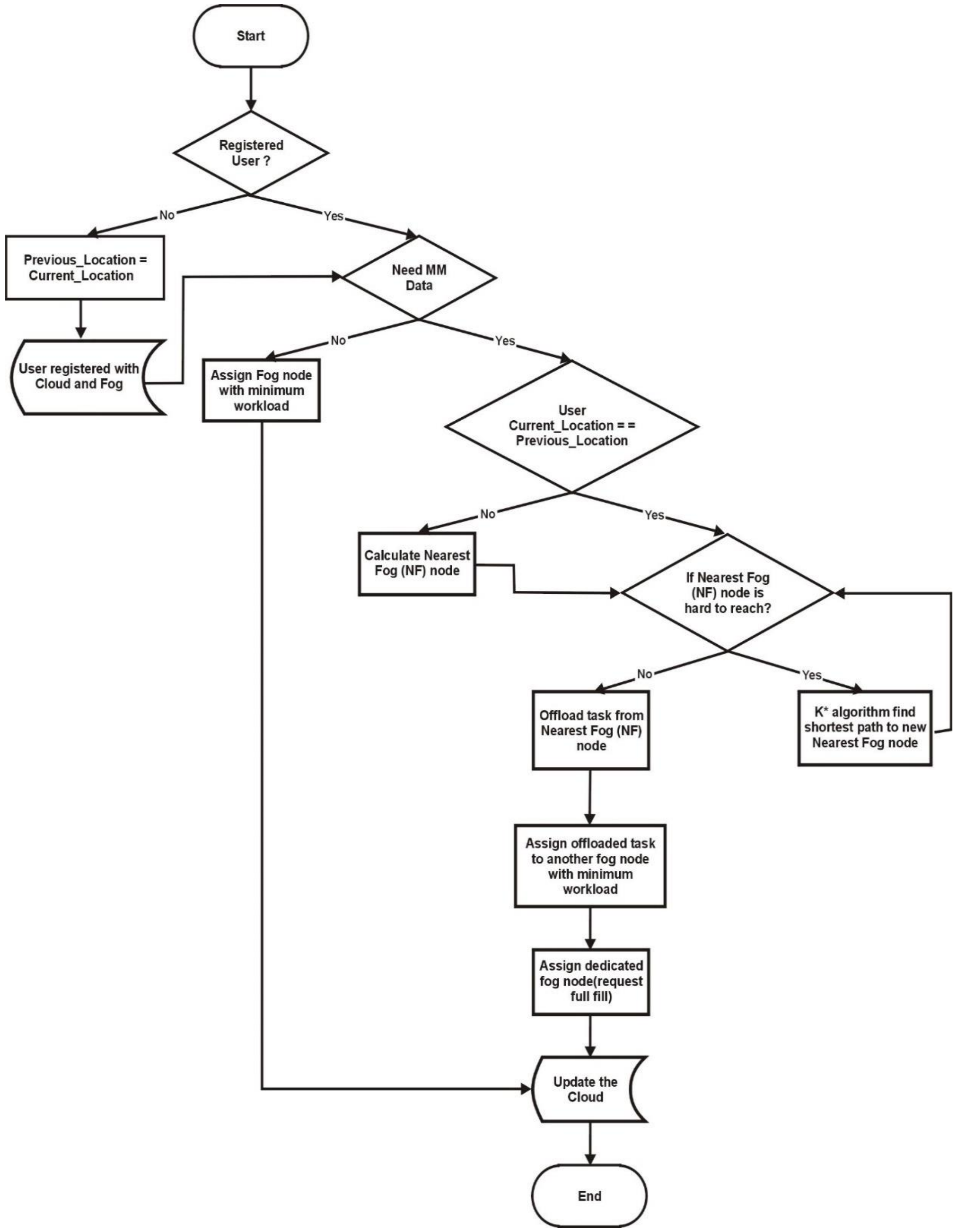

Figure 3.3: *Flow diagram of proposed algorithm*

3.7 Use Case Scenario

Important use case scenario of the lightweight LAFF is presented in this section.

3.7.1 Use Case

To prove the significance of the proposed algorithm a use case is described**.**

3.7.1.1 Actors

Jeena, Theif, Users (Police Vans)

3.7.1.2 Pre-Conditions

Registered user with known location and requested MMD.

3.7.1.3 Post-Conditions

A user is able to request Fog framework to access CCTV cameras to get live streaming.

3.7.1.4 Scenario

Jeena is walking through a street, a thief snatched her bag and ran away. Jeena called the police and complained about the thief. The police man asked Jeena's location where she is now and to which direction the thief has gone. Jeena provides police officer his desired information. The police officer started tracking the thief through CCTV cameras to get live-streaming of the thief and also informed the police vans of the area where the thief is traced. The police vans caught the thief through accessing the exact location of the thief.

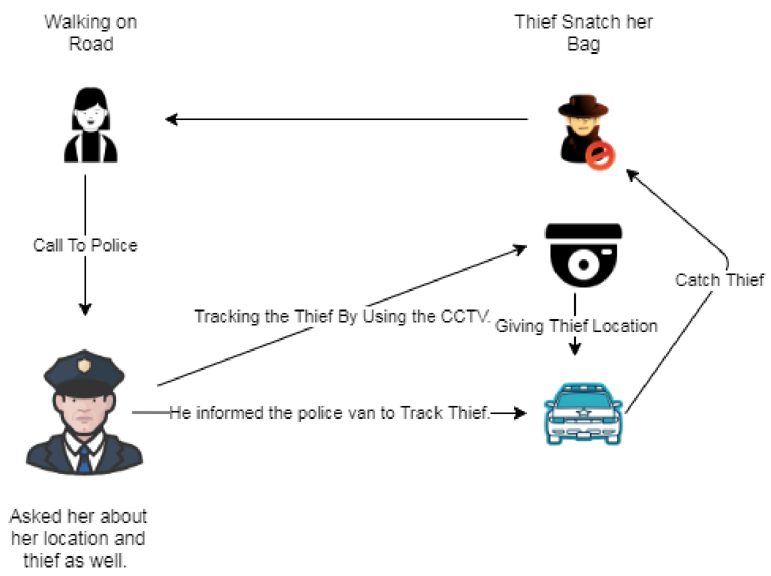

Figure 3.4: *Flow of the scenario activities*

However, live-streaming is a heavy task to run and requires lot of computational power which require a framework with low latency, service time and network use to assure QoS. In this case a nearest fog node will be assigned to the police vans so that they can trace the thief without any data loss and interruption.

3.8 Summary

A lightweight fog framework is proposed to reduce latency, network use and service time to provide QoS to the end user. The research gap identified by reviewing the literature that resource provisioning does not consider location management of users, is addressed by designing lightweight LAFF. QoS is provided to the users by location management and analyzing the requested data type. K* algorithm helps finding the nearest fog node in case any other fog node is hard to reach.

CHAPTER 4

EVALUATION

4.1 Overview

The following chapter outlines the experimental setup and describes the approach applied in this research. Further, a section is devoted to present the characteristics of used hardware, and software libraries applied to conduct the simulation.

4.2 Experimental Setup

This lightweight LAFF is developed by using CloudSim [24] and iFogSim simulator [25] on eclipse. CloudSim is responsible for the simulation and events handling at Cloud. iFogSim is responsible for events handling at Fog devices. The addition of a layer of fog devices ensure the data security [157] and low latency. It also minimizes the delays as hosts become near to edge of devices [116]. Multiple libraries are used for different purposes. Libraries like Javascript object notation (Json) data saver, jfreechart and common-math are used. Priority Scheduling is used for task and resource allocation. Huge amount of data is generated from the IoT devices. Excel and CSV sheets are created to store this massive data. Below are the important parameters and steps required to set in order to execute simulation

4.2.1 Step 1

The calendar is initialized to keep the record to conclude at the end when the simulation is started. In the end, simulation variable is initialized by tracing flag to “false” so that the details log which are not relevant to the simulation are not shown.

4.2.2 Step 2

Fog broker initialization is based on data center broker considering the requirements of the clients related to QoS. The data center broker class coordinates between users and cloud service. A fog broker helps users to create tuples on the fog. Tuples are extended from cloudlets class to model tasks in CloudSim and iFogSim.

4.2.3 Step 3

The cloud and fog data centers have their own characteristics. In real case, the characteristics of fog device are less powerful and have less storage than cloud data center. The capacity function of Cloud $C(l+d)^n$ and Fog $F(l)^n$ for load l and new expected data *d* is represented in equation (4.1) and equation (4.2).

$$C(l+d)^n = \sum_{k=0}^{n} \binom{n}{k} u^k d^{n-k} \tag{4.1}$$

$$F(l)^n = \sum_{k=0}^{n} \binom{n}{k} u^k l^{n-k} \tag{4.2}$$

Where n is number of total requests and k represents the capacity of responses that are sent against the requests n. The response k is always sent against request n. The $C(l+d)^n$ function equation (4.1) shows that the Cloud has more storage than Fog devices (equation 4.2).

Below are the tables for devices and their characteristics:

4.2.4 Cloud Data Centers

Cloud Data Centers (CDCs) are the centralized hosts and play a supervisory role to handle communication and data storage. Cloud storage has many distributed resources acting as one unit. Characteristics of Cloud Data Center are described in **Table 4.1**.

Table 4.1: *Characteristics of Cloud Data Center*

Name of Device	Cloud-DC
Million instructions per second	44480
Ram	20GB
Uploading Bandwidth	100 Mbits
Downloading Bandwidth	10 Gbits
Level	0 (Top)

4.2.5 Fog Data Centers

Fog data centers store data for further processing and communication with users. Fog data centers work in distributed manner. Characteristics of Fog Data Center are given below in **Table 4.2**.

Table 4.2: *Characteristics of Fog Data Center*

Name of Device	Fog-DC
Million instructions per second	20000
Ram	10GB
Uploading Bandwidth	10 Gbits
Downloading Bandwidth	10 Gbits
Level	1 (Cloud Child)

4.2.6 Location Manager Data Centers

Location management data centers store information regarding user's location. The location manager data center stores the previous location and also checks current locations of the users. Characteristics of location manager Data Center are presented below in **Table 4.3**.

Table 4.3: *Characteristics of location manager Data Center*

Name of Device	Location-manager
Million instructions per second	2000
Ram	1GB
Uploading Bandwidth	10 Gbits
Downloading Bandwidth	10 Gbits
Level	1

4.2.7 Fog Head

Fog head knows the location of all fog nodes and communicates between the cloud and all nodes. Fog head is also responsible for managing and maintaining the information on the hardware level. Fog head characteristics are given below in **Table 4.4**.

Table 4.4: *Characteristics of fog head*

Name of Device	Fog-Head
Million instructions per second	20000
Ram	8GB
Uploading Bandwidth	10 Gbits
Downloading Bandwidth	10 Gbits
Level	2

Fog-TD refers to the textual data: audio, pictures and texts. Characteristics of fog-TD are as described in **Table 4.5**.

Table 4.5: *Characteristics of Fog-TD*

Name of Device	Fog-TD
Million instructions per second	20000
Ram	2 GB
Uploading Bandwidth	10 Gbits
Downloading Bandwidth	10 Gbits
Level	3 (Fog head child)

Fog-MMD is refers to multi-media data which includes videos and live streaming. Characteristics of fog-MMD are presented in **Table 4.6.**

Table 4.6: *Characteristics of Fog-MMD*

Name of Device	Fog-MMD
Million instructions per second	20000
Ram	4GB
Uploading Bandwidth	10 Gbits
Downloading Bandwidth	10 Gbits
Level	3 (Fog head child)

4.2.8 Gateway Devices

These gateway devices are part of the fog layer and responsible for communication between proxy server and cloud devices. Here are the characteristics of gateway devices in **Table 4.7.**

Table 4.7: *Characteristics of gateway devices*

Name of Device	Gateway Devices
Million instructions per second	1000 Mips
Ram	1GB
Uploading Bandwidth	10 Gbits
Downloading Bandwidth	10 Gbits
Level	4

4.2.9 Sensor Devices

Sensor devices are created for the present scenario which produce the data with following characteristics. Sensor devices characteristics are presented in **Table 4.8**.

Table 4.8: *Characteristics of sensor devices*

Name of Device	Sensor Devices
Million instructions per second	1000 Mips
Ram	1GB
Uploading Bandwidth	10 Gbits
Downloading Bandwidth	256 Mbits
Level	1

4.2.10 Sensors and Actuators

As the actual device model is based on sensor devices, generating a huge amount of data that needs to be processed, each device involves a sensor and an actuator attached to it. The purpose of the sensor is to 'sense' the data which is identified by the selector module of the server.

4.2.10.1 Module to module interaction

Tuples are sent from one module to the other in order to interact with each other. The tuples which are sent up to the fog or cloud for processing, are identified as TuplesUp and tuples that are sent downward from one module to the other are TupleDown. Also, tuples are mapped to modules using the tuple mapping techniques defined in iFogSim. The network usage is calculated on the basis of tuples flow. The network usage μ^n is defined in terms of μ^{fog} (fog network length) and μ^{cloud} (cloud network length) by dividing a tuple size T^L with simulation total time ꞩ as presented in equation (4.3).

$$\mu^n \rightarrow \mu^{fog} + \mu^{cloud} \prod T^L / ꞩ \quad (4.3)$$

Windows builder plugin is used for creating attractive graphical user interface. The Gui is compromised on the standards of Human Computer Interaction (HCI). The app screenshot can be seen below in **Figure 4.1**.

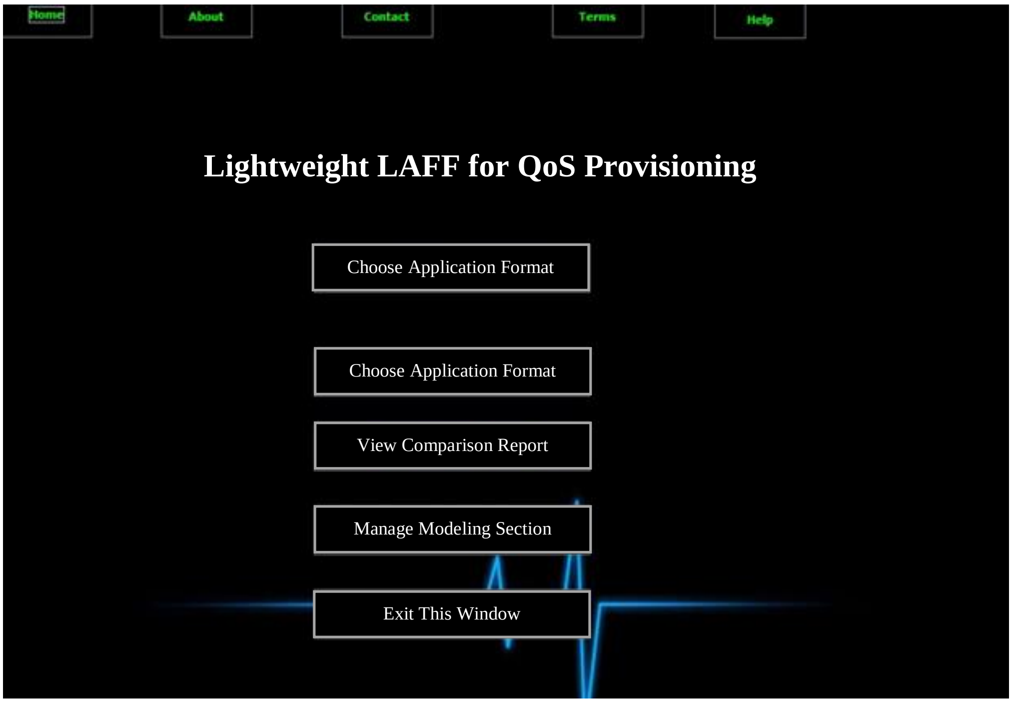

Figure 4.5: *Front end of the system*

Topology

The topology of the lightweight LAFF is shown in **Figure 4.2**:

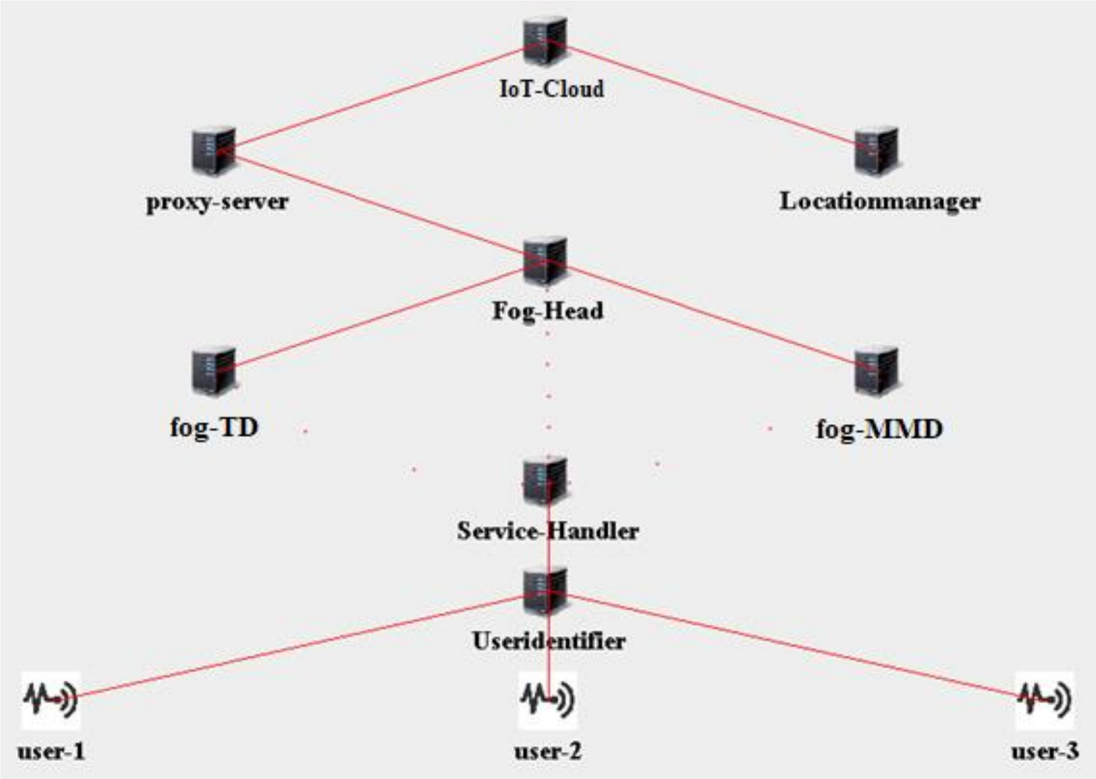

Figure 4.6: *Topology of the lightweight LAFF*

4.2.11.1 Explanation

Initially the normal flow of the system is as:

User->UserIdentifier->ServiceHandler-> FogHead ->proxyServer->Fog(MMD or TD) ->Cloud-server

The fog head handles user's requests. Through location management module, user's location is traced and a fog head is deployed there to respond. If the location manager is not idle, then the proxy server can be formed. Fog head asks user identifier to identify the type of requested data. Requests may be for MMD or TD. After the Fog head determines the type of the requested data, it allocates the required fog nodes to the users. The specific fog node facilitates the user accordingly. The fog-MMD node is loaded with very powerful processing capability whereas a low spec is configured on fog-TD node. **Table 4.5** and **Table 4.6** represents the specifications of both fog-TD and fog-MMD. The proposed algorithm makes this work unique and distinguishing.

The flow after initial one is given below.

User->UserIdentifier->ServiceHandler->FogHead->Fog(MMD or TD) ->User

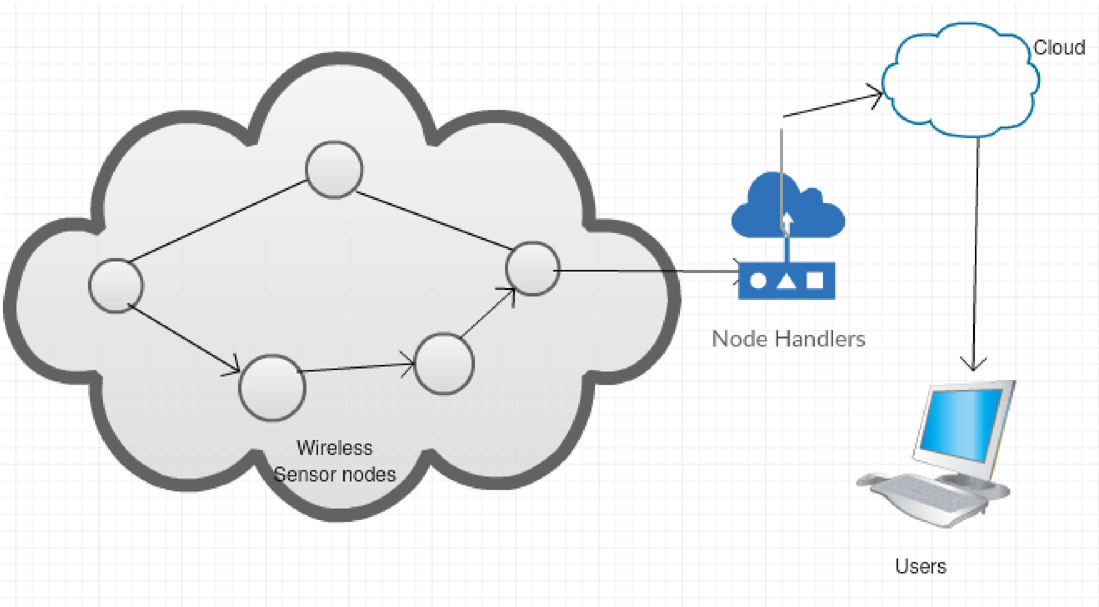

Figure 4.7: *Flow of the LAFF in case when fog head fails to fulfil user's request.*

If Fog head fails to identify the relative fog service provider, the request is then transferred to the Cloud-server to facilitate the user as represented in **Figure 4.3**. The lightweight LAFF is a fault-tolerant framework due to the cloud's availability in case fog head fails to fulfill the request.

4.3 Data Configuration

A data set with tuples size 3000 bytes, bandwidth 1000 KB and network length 500 gigabytes is implemented in below mentioned configurations. The tuple in iFogSim, represents the term data row where there is sequence of bytes in such data rows.

Simulation runs on iFogSim for different configurations. The configurations are presented in **Table 4.9**.

Table 4.9: *Data configuration*

Configuration	No of fog nodes	No of Users
1	2	10
2	3	16
3	4	20
4	5	24
5	6	28
6	7	36
7	8	48
8	9	52
9	12	60
10	15	90

The results of the above-mentioned configurations are shown in the next chapter.

4.4 Summary

The experimental setups for the lightweight LAFF are presented in this chapter. It is concluded that lightweight LAFF is fault tolerant framework because in case of failure to provide services by fog head, cloud is there to facilitate users.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Overview

Proficiency of the proposed framework is surveyed. Different parameters, such as latency, network usage and service time in fog architecture are considered and compared. The lightweight LAFF under the state-of-the-art simulation is performed and results are also compared with other fog-based frameworks: TPFC and IFAM. The results are mentioned in this chapter.

5.2 Architecture Comparison

The lightweight LAFF is compared with two other fog-based frameworks: IFAM (Intelligent FC Analytical Model) and TPFC (Task Placement on Fog Computing) [20, 21]. The primary motivation behind this evaluation is to confirm the adequacy of the LAFF in terms of reducing latency, service time and network use to facilitate users by providing better QoS. A use case is also presented at the end of the chapter. Both frameworks IFAM and TPFC are described below:

5.2.1 Intelligent FC Analytical Model (IFAM)

In IFAM [20] an analytical model and reinforcement learning algorithm in an FC environment is introduced. This model aims to reduce latency among healthcare IoT, cloud servers and end-users. This research proposes a novel multitier fog processing system that provides IoT services. However, in this work the author did not consider user's location, and network conditions. The other drawback of this research is that user's request for normal data is transferred to cloud to respond [175]. The LAFF is better in terms that it considers user's location and network conditions. The framework (LAFF) also transfers both type of data, MMD and TD to fog to fulfill user's request.

5.2.2 Task Placement on Fog Computing (TPFC)

In TPFC [21] a context aware information-based approach is described which ideally uses virtual resources accessible on the system edges to improve the presentation of IoT benefits in terms of response time, cost and energy decrease. The approach utilizes context-aware information including network conditions, location of IoT devices and service type to provide resources to IoT applications. However, the increase in the number of fog nodes and services cause exponential increase in time for problem solving [175]. Architectural comparison among the most relevant frameworks is described in **Table 5.1**.

Table 5.1:*Architectural comparison among the most relevant Frameworks*

LAFF [175]	IFAM [20]	TPFC [21]	FogPlan [128]
LAFF taken into account various IoT data requirements (Multimedia Data, Textual Data)	Transfer light data type to cloud [175].	TPFC taken into account various IoT data requirements.	FogPlan does not consider multiple IoT data requirements.[176]
LAFF is location Aware Framework, knows the exact location of the users	Unable to access user's exact location [175].	TPFC is a location aware framework.	Does not know the exact location of users [176]
The algorithm use K* heuristic algorithm to find the shortest path between user and fog node, if the fog node is hard to reach	Does not have such backup.	Does not have such backup [175].	Does not have such backup.
The algorithm takes decision considering the requested data type. Does not release low demand services.	The IFAM algorithm takes decision on the basis of latency.	The proposed framework takes decision on the basis of services type.	The algorithm prioritizes the deployment of high demand services on fog nodes and releases low demand services.[177]
LAFF considers three parameters to provide better QoS; network use, latency and service time	Service time is not addressed in this work.	Service time is not addressed in this work.	FOGPLAN consider one parameter delay to provide QoS [175]
LAFF Improves computational resources (RAM and CPU) to prove that LAFF is a lightweight framework.	Considers only RAM Consumption.	Resource utilization is not considered.	FOGPLAN consider cost factor to prove that the framework is lightweight [175].
LAFF registers users on fog head.	Does not register user.	Does not register users.	Does not register users [175].
LAFF Consider network condition	Network condition is not considered in this work [175].	The framework considers network condition.	Does not Consider network condition.

The results of the comparison are presented in section 6.4.

5.3 Procedure to get results from App

Step1: User run the java Application (jar or from Eclipse)

Step2: User view Simulation of Service Option **Figure 5.1.**

Step3: User click on the options given below

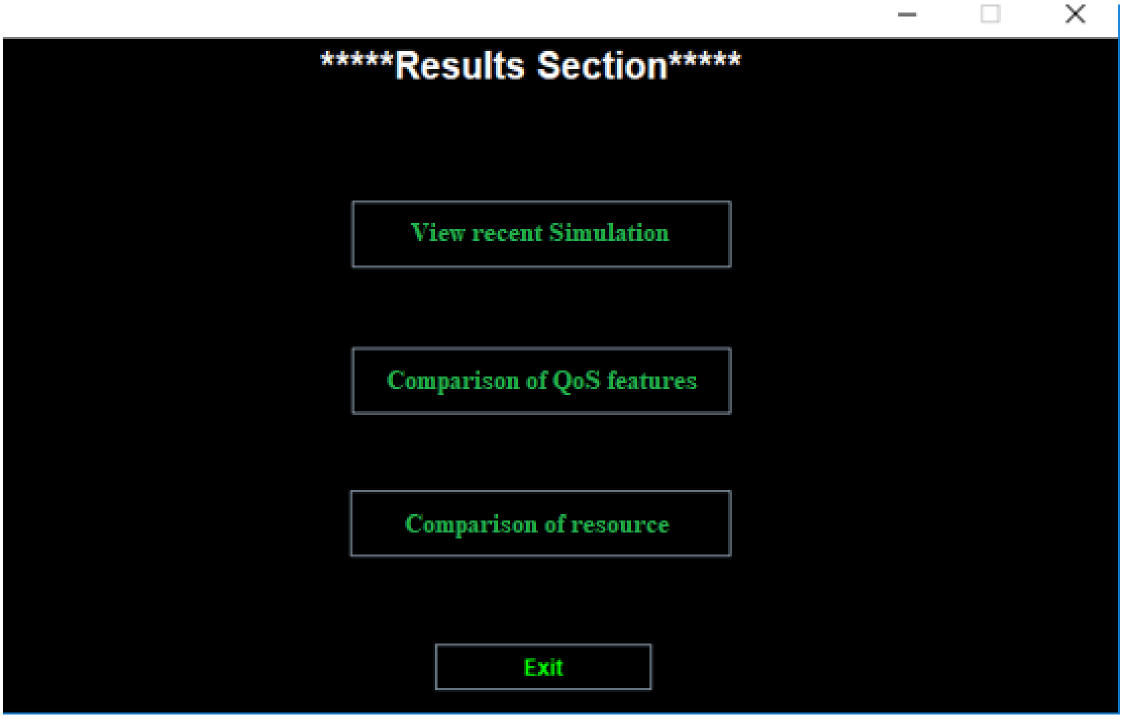

Figure 5.1: *GUI of the results section*

5.4 Simulation Results

The results gained through the simulation for LAFF are described below. The results are compared with other fog-based frameworks to prove the proficiency of the lightweight LAFF. Various parameters such as latency, network usage and service time are recorded and compared. Ten configurations are employed with varying number of devices and nodes so that a consistent pattern is extracted.

5.4.1 Latency

Security applications are very time-sensitive. Results cannot be delayed. For instance, if we come to know that a terrorist is going to blast a bomb somewhere, finding terrorist's locations on time is a crucial and time-sensitive task. Delay cannot be afforded as it can lead to negative consequences. This delay is calculated by implementing a control loop. Latency is calculated by using a module-to-module latency and then average of them is taken, latency is much higher when IFAM and TPFC modules are executed as depicted in **Figure 5.2**. This comparison is done in established scenario for the LAFF. The results show that the lightweight LAFF reduced average latency by 11.01 percent when compared to both frameworks. The agenda not only stops at reducing the latency but also reduces the network usage and service time in order to provide better QoS and consistent data.

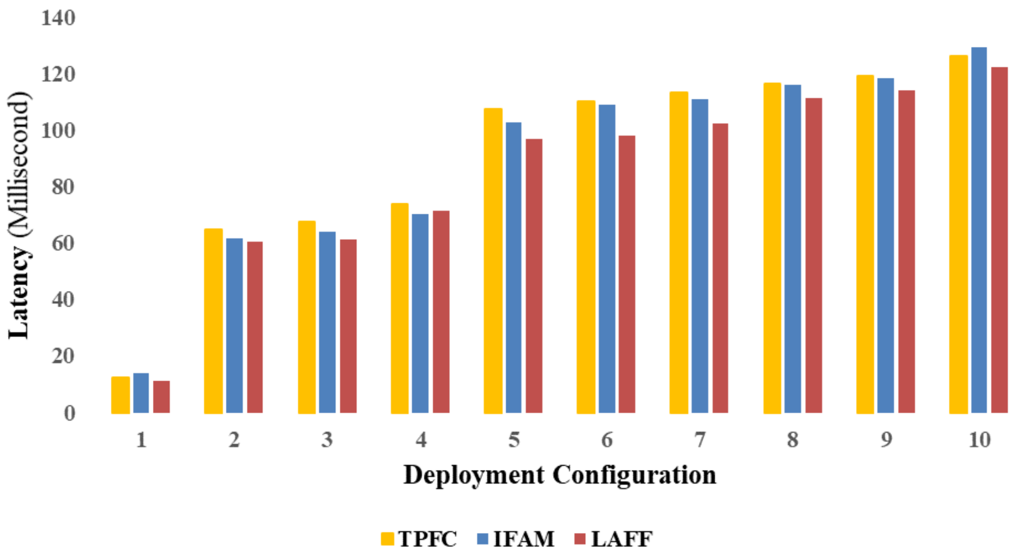

Figure 5.2: *Latency Comparison*

5.4.2 Network Usage

This parameter is characterized by the utilization of system resources in terms of data sent and received from the network interfaces. The network usage should be kept at minimum for better performance. LAFF reduced network traffic and consumption in terms of resource utilization. The results depict that the network utilization of the LAFF is reduced by average 7.51 percent as compared to IFAM and TPFC as shown in **Figure 5.3**. The comparison of the LAFF with TPFC and IFAM showed that the LAFF provides better QoS.

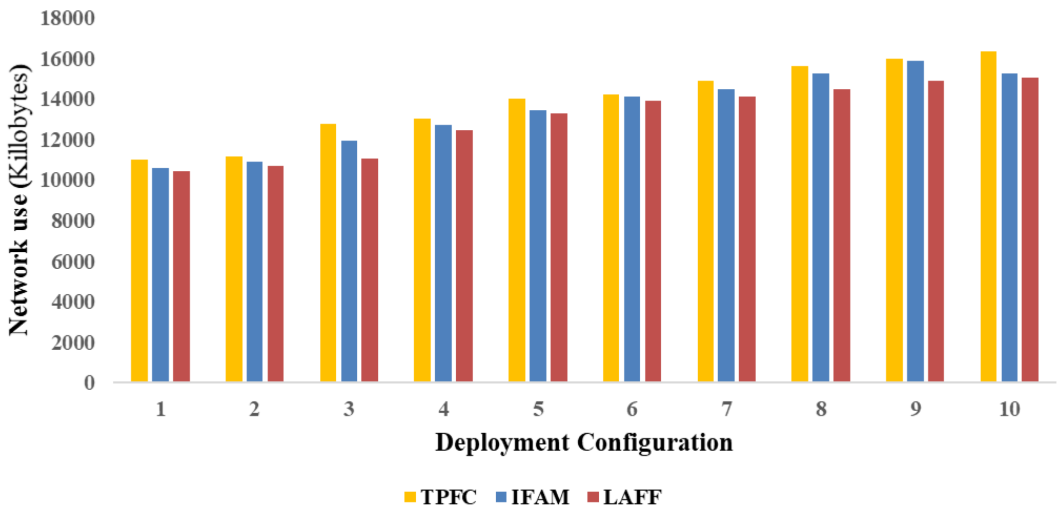

Figure 5.3: *Network Usage Comparison*

5.4.3 Service Time

Service time is the most important parameter in terms of QoS. Service time is the amount of time spent to provide services to a user by a service provider. The service providers are the small hosts integrated with fog nodes and cloud in order to use storage and transmissions. The service time comparison is shown in **Figure 5.4**. It shows that the average amount of time is 14.8 percent lesser than the TPFC and IFAM.

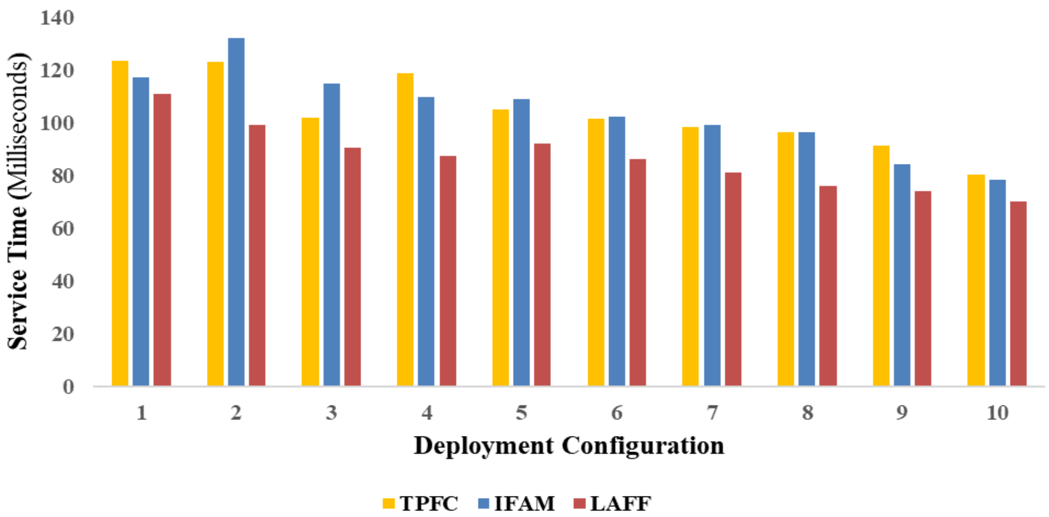

Figure 5.4: *Service Time Comparison*

5.5 LAFF as a Lightweight Framework

LAFF is a lightweight framework as it consumes less computational resources. RAM consumption and CPU utilization of a framework can increase the burden on resources. Since most of the Fog nodes are not abundant in resources, execution of heavyweight software systems can cause significant computing overhead on them. Therefore, it is required to deploy lightweight frameworks in Fog computing environments. The framework that consumes less RAM and CPU, is considered lighter than the other frameworks [158]. Ten configurations are employed with varying numbers of devices and nodes so that consistent patterns could be extracted.

5.5.1 Ram Consumption

RAM is one of the most important components of the fog node. If the framework consumes more ram, the ram system will crash and become unresponsive. To prove that the proposed framework is lightweight, RAM consumption of the framework is compared with TPFC and IFAM. The **Figure 5.5** shows the RAM consumption for the data transmission and processing in fog nodes. The results showed that the proposed framework's RAM consumption is average 8.41 percent less than the both compared frameworks.

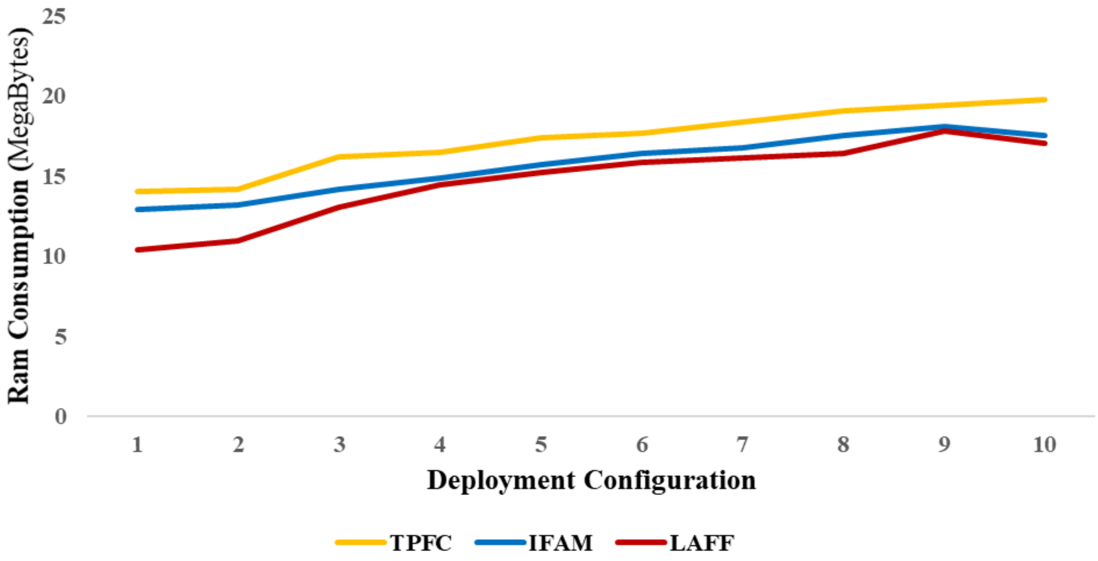

Figure 5.5: *Ram Consumption*

5.5.2 CPU Utilization

CPU utilization is the amount of work handled by a CPU of the fog node. The time taken between the start and the completion of a given task executed on a fog node is referred as CPU utilization and measured in milliseconds. The time taken for separating and combining tasks before and after their scheduling is not included in the study. A task is composed of a set of instructions. It is assumed that each instruction requires one clock cycle to be executed. In the proposed framework offloading module helps to minimize the CPU utilization, therefore, increasing the fog node performance. The results showed in **Figure 5.6** that the proposed framework's CPU utilization is average 16.23 percent less than the both compared frameworks.

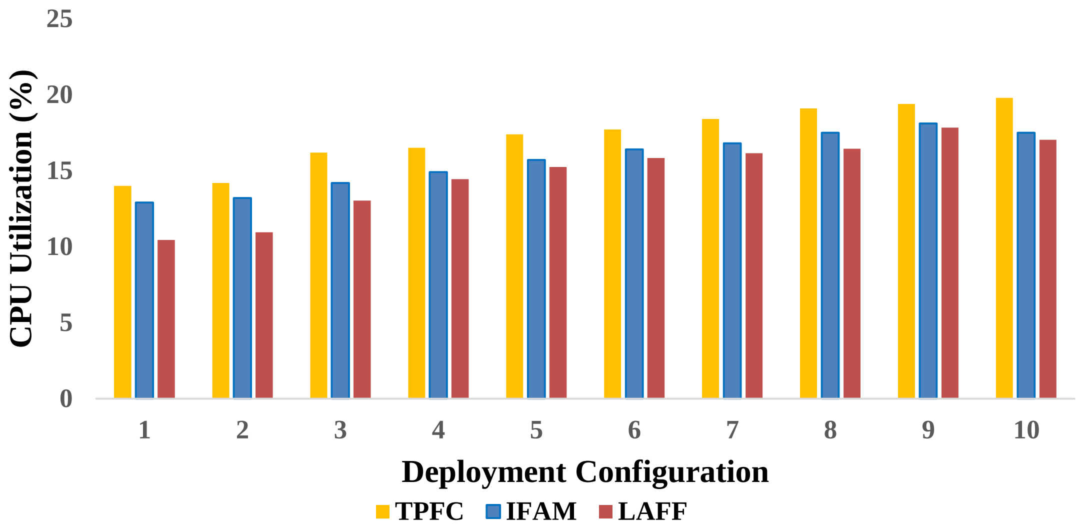

Figure 5.6: *CPU Utilization*

5.6 Comparison with related works

Comparative analysis of the proposed work and related works is done in which all parameters are consider. Below is the **Table 5.2** providing the best picture of the comparison of parameters of related work and proposed framework.

In the lightweight LAFF, QoS is achieved by reducing latency, network use and service time. As data processing and filtration is done near to the edge due to the fog layer that's why it uses less bandwidth to execute requests. To manage the load of the fog nodes, fog head distribute the load on other fog nodes for better utilization of the resources. The lightweight LAFF provide mobility, as users are mobile. This framework is heterogeneous because it allows different types of sensors and user devices to connect with each other. Fog head and fog nodes can be added or removed in the proposed framework that's how it provides scalability. It also provides better management due to easy access to fog layer as other modules can be incorporated in this system. Reduced latency, network use, service time is achieved through this framework as discussed in this chapter above. **Table 5.2** presents the clear view that the lightweight LAFF have significant features from other frameworks.

Table 5.2: *The parameter comparison*

Citation	**Year of Publication**	**QoS**	**Latency**	**Network Usage**	**Service Time**	**Network Condition**	**Load Balancing**	**Mobility**	**Heterogeneity**	**Scalability**	**Management**	**RAM Consumption**	**CPU Utilization**
G. Myrizakis et al [130]	2020	✓					✓					✓	✓
J. Santos et al [174]	2019										✓		
A. Yousef et al [128]	2019	✓	✓	✓			✓	✓	✓	✓	✓		
Prabhdeep S et al[131]	2019	✓	✓								✓		
Minh-Q et al [20]	2019	✓	✓	✓			✓	✓	✓	✓	✓		✓
Shukla S et al [21]	2019	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	
S. Ningning et al [105]	2018				✓		✓				✓		
Q. Fan et al [97]	2018						✓				✓		
Istabraq M et al [121]	2018	✓					✓	✓					
Changlong Li et al [94]	2018	✓					✓				✓		
Viorel Mihai et al [88]	2018		✓							✓			
A. Kapsalis et al [158]	2017	✓							✓				
R. Moreno et al [159]	2017								✓	✓			
S. Li et al [167]	2017	✓							✓				
M. Malensek et al [91]	2017								✓				
L. Gu et al [169]	2017	✓							✓				

Q. He et al [170]	2017	✓							✓				
M. Tang et al [171]	2017	✓							✓				
L. Gu et al [98]	2017	✓						✓		✓			
Skarlat, O et al [132]	2017	✓	✓								✓		
Ramírez W et al [144]	2017	✓									✓		
Y. pour et al [145]	2017	✓											
Lin Y et al[129]	2017	✓	✓	✓			✓		✓	✓	✓		
G. Myrizakis et al [130]	2016	✓						✓		✓			
X. Hou et al [160]	2016							✓					
Y. Yan et al [161]	2016									✓			
R. Brzoza et al [126]	2016									✓			
K. Inthara et al [163]	2016	✓											
D. Ye et al [166]	2016	✓						✓					
C. Fricker et al [92]	2016	✓							✓	✓			
F. Jalali et al [168]	2016	✓							✓				
S. Sarkar et al [172]	2016	✓								✓			
S. Agarwal et al [173]	2016								✓				
R. Deng et al [101]	2016	✓								✓			
V. Manisha et al [102]	2016		✓		✓		✓						
M. Aazam et al [162]	2015							✓					
R. Deng et al [164]	2015	✓							✓				
M. Aazam et al [165]	2015	✓							✓				
J. Oueis et al [123]	2015	✓							✓				
Our Work		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Figure 5.7 shows contributions of the existing literature redarding related parameters.

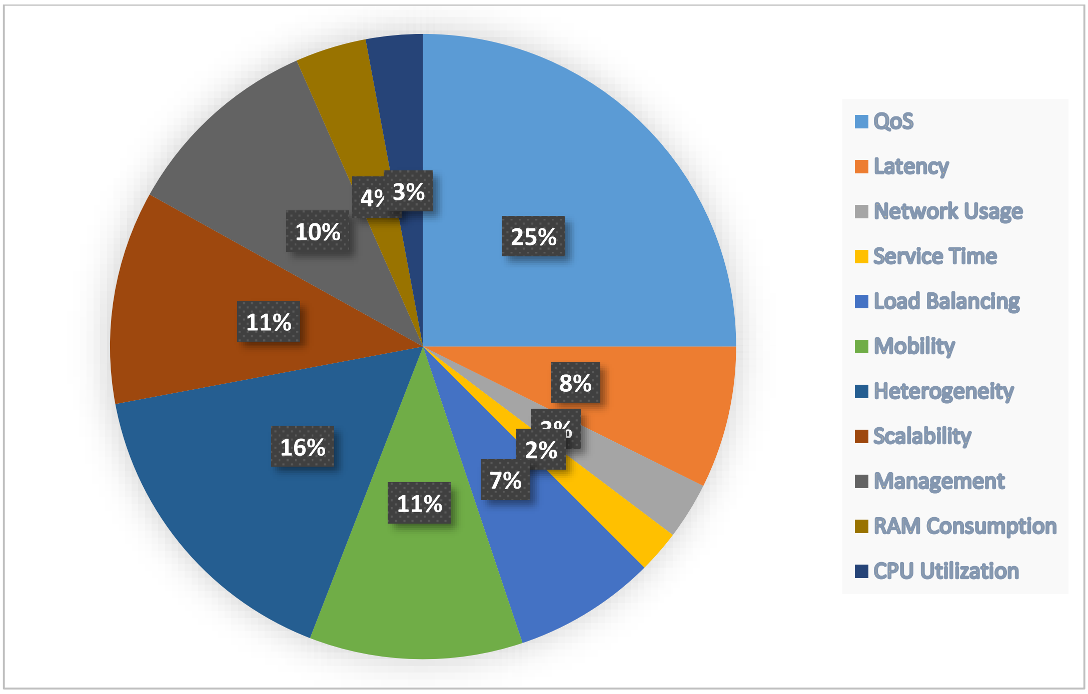

Figure 5.7: *Existing literature contribution of related parameters*

5.7 Summary

Lightweight LAFF employs lightweight procedures to provide better services to users. It is examined that the lightweight framework lessens latency by 11.01 percent, network usage by 7.5 percent and service time 14.8 percent as compared to TPFC and IFAM. The framework provides better QoS, by reducing latency, network use, and service time. This framework is scalable, heterogeneous, and provide mobility. The framework is lightweight as it consumes resources RAM 8.41 percent and CPU 16.23 percent less than the compared frameworks.

CHAPTER 6

CONCLUSION AND FUTURE DIRECTION

The research aims to open up the door towards new ways of computing. This research has gone through different experiments to conclude the results. Thousands of lines of codes are written to perform this simulation. The results are drawn from simulation of the framework.

This research aims to provide better quality of service in terms of reduced latency, network use and service time. The objectives proposed in section 1.6 are achieved as follows: Latest literature of the same domain is reviewed and research gaps are identified. These research gaps help to established new research objectives. The State of the arts frameworks did not consider distance between IoT nodes and fog server, resulting in increased latency. This large distance also increases the number of hops and nodes between the end devices and the fog servers which increase the network use. Due to this increase in latency and network use rise the service time. Hence, the need to design a lightweight framework emerged to provide QoS to users. Therefore, by utilizing the traditional three tier fog architecture, a lightweight framework is designed to enhance QoS. In this lightweight framework, the fog layer is updated with user management module, location management module, offloading module, load balancing module. The K* algorithm module is added to find the shortest path between the source and destination if any fog node is hard to reach. Users demand uninterrupted services in time sensitive applications, during live streaming and web gaming. To provide better QoS to these users, an algorithm is developed which enables the proposed framework to access user's exact location. Users are registered on fog head node. This user registration reduces service time. This framework minimizes latency, network use and service time, ultimately improving the QoS. The proposed framework is claimed as "lightweight", to prove the argument, the RAM and CPU utilization is reduced and compared with two other fog-based frameworks. The comparison shows that the proposed framework utilizes minimum resources. The proposed framework is implemented in simulation environment by using iFogSim.

The access to data and content is more smother and faster when accelerated with location awareness. Responsiveness and consistency are increased if latency is minimized and bottleneck issues are catered. Therefore, LAFF is designed as a location aware algorithm which ensures rich user experiences and provides better QoS by reducing network utilization, service time and latency. Fog layer performs fast, reliable and efficient data communication [166] and transfer. It helps in localized analysis for different types of decision just like in present scenario locally requested data is analyzed [6]. Some users need dedicated service and some need general service.

By providing consistent dedicated service to users, people can be attracted to purchase services. This can only be done when a lightweight framework exists in fog environment. The previously proposed frameworks are performing one specific task or two such as load balancing, resource management etc. However, the framework proposed through this research is performing multiple tasks at a time such as resource management, load balancing, cloud management and also reduce latency, network use and service time. The previously proposed frameworks are implemented in a system as a module because they are performing one or two specific tasks. But this lightweight LAFF will be implemented as a whole system because it is performing all required tasks by itself that the previously implemented frameworks are doing such as resource management, load balancing, cloudlet management, resource provisioning etc. The concept of Fog head is used in this research. Fog head manages fog nodes, as the load balancing of fog node is done in fog head not in the cloud. Therefore, fog head helps in reducing time by communicating to the cloud. Fog head also register users by accessing their location. Fog head updates user's nearest fog node to fulfil user's request more competently.

It is examined that the LAFF lessens, average latency, network use and service time by 11.01 percent, 7.51 percent and 14.8 percent respectively as compared to IFAM and TPFC. Similarly, resource utilization in terms of RAM and CPU is reduced by average 8.41 percent and 16.23 percent as compared to TPFC and IFAM making LAFF comparatively a lightweight framework. Location aware feature is significant in defense and intelligence areas. Hence, the proposed LAFF improves QoS while accessing remote computational servers for the outsourced applications in fog computing. The limitation of the proposed framework is that fog nodes are assumed as static (not moveable).

It is suggested that the future work will include the utilization of the proposed lightweight framework with various experimentation of machine learning techniques to determine the optimal distribution and configuration of fog nodes while taking into consideration the requested data type by using time module and data type of previously sent requests. Moreover, in case of system failure machine learning technique will provide backup. In addition, optimization mechanisms will be added to determine the optimal distribution and configuration of fog nodes while taking into consideration the computational resources.

REFRENCES

[1] Mike J. Walker, Betsy Burton, and Michele Cantara. Gartner Hype Cycle 2018. https://www.gartner.com/doc/3383817, 2018.

[2] McKinsey & Company, “The Internet of Things: How to capture the value of IoT.”, https://www.mckinsey.com/featured-insights/internet-of-things/ our- insights/the- internet- of- things- how- to- capture- the- value- of- IoT. May, 2018.

[3] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility”, *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599 – 616, 2009.

[4] Marques G, Pitarma R, M. Garcia N, Pombo N. Internet of Things Architectures, Technologies, Applications, Challenges, and Future Directions for Enhanced Living Environments and Healthcare Systems: A Review. *Electronics.* 2019; 8(10):1081. https://doi.org/10.3390/electronics8101081.

[5] [4] M. Amiri, H. Al Osman, S. Shirmohammadi, and M. Abdallah, "Toward Delay-Efficient Game-Aware Data Centers for Cloud Gaming," *ACM Trans. Multimed. Comput. Commun*. Appl., vol. 12, no. 5s, pp. 1--19, Sep. 2016.

[6] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog Computing and Its Role in the Internet of Things”, in First Edition of the *MCC Workshop on Mobile Cloud Computing*, ser. MCC, 12. ACM, 2012, pp. 13–16.

[7 S. Park, O. Simeone and S. Shamai Shitz, “Joint optimization of cloud and edge processing for fog radio access networks”, *IEEE Transactions on Wireless Communications*, vol. 15, no. 11, pp. 7621–7632, Nov 2016.

[8] H. Madsen, B. Burtschy, G. Albeanu, and F. Popentiu-Vladicescu, “Reliability in the utility computing era: Towards reliable Fog computing”, in 2013 *20th International Conference on Systems, Signals and Image Processing* (IWSSIP), July 2013, pp. 43–46.

[9] Cisco Inc., “Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are”, https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/ computing-overview.pdf, 2015.

[10] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog Computing and Its Role in the Internet of Things”, in *First Edition of the MCC Workshop on Mobile Cloud Computing*, ser. MCC, 12. ACM, 2012, pp. 13–16.

[11] National Intelligence Council. Six Technologies with Potential Impacts on US Interests out to 2025. Disruptive Civil Technologies 2008, April 2008.

[12] 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: A complete survey”, *Journal of Systems Architecture*, vol. 98, pp. 289 – 330, 2019.

[13] J. Yao and N. Ansari, “QoS-Aware Fog Resource Provisioning and Mobile Device Power Control in IoT Networks”, *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 167–175, March 2019.

[14] C. Puliafito, E. Mingozzi, F. Longo, A. Puliafito, and O. Rana, “Fog computing for the internet of things: A Survey”, *ACM Transactions on Internet Technology* (TOIT), vol. 19, no. 2, p. 18, 2019.

[15] A. H. Sodhro, Z. Luo, A. K. Sangaiah, and S. W. Baik, “Mobile edge computing based QoS optimization in medical healthcare applications”, *International Journal of Information Management*, vol. 45, pp. 308–318, 2019.

[16] S. Yan, M. Peng, M. A. Abana and W. Wang, "An evolutionary game for user access mode selection in fog radio access networks", *IEEE Access*, vol. 5, pp. 2200-2210, 2017.

[17] S. F. Hassan and R. Fareed, "Video streaming processing using fog computing", 2018 *International Conference on Advanced Science and Engineering* (ICOASE), Duhok, 2018, pp. 140-144, doi: 10.1109/ICOASE.2018.8548869.

[18] S. Mi Jun and N. Hoon Park, "Controlling non IP bluetooth devices in UPnP home network", The 6th *International Conference on Advanced Communication Technology*, Phoenix Park, Korea, 2004, pp. 714-718.

[19] A. Yousefpour, G. Ishigaki, R. Gour and J. P. Jue, "On Reducing IoT Service Delay via Fog Offloading", in *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 998-1010, doi: 10.1109/JIOT.2017.2788802, April 2018.

[20] M.Q. Tran and D. T. Nguyen, V. A. Le, D.H. Nguyen. and T.V. Pham, “Task Placement on Fog Computing Made Efficient for IoT Application Provision”, *Wireless Communications and Mobile Computing*, Article ID 6215454, 17 pages, 2019. https://doi.org/10.1155/2019/6215454. vol. 2019.

[21] Shukla S, Hassan MF, Khan MK, Jung LT and Awang A, “An Analytical Model to Minimize the Latency in Healthcare Internet-of-Things in Fog Computing Environment”, *PLOS ONE* 14(11): e0224934, 2019.

[22] W. Kim and S. Chung, "User-Participatory Fog Computing Architecture and Its Management Schemes for Improving Feasibility", in *IEEE Access*, vol. 6, pp. 20262-20278, 2018. doi: 10.1109/ACCESS.2018.2815629.

[23] Wazid, M. Das, A. Kumar & Kumar, Neeraj & Vasilakos, Athanasios., "Design of Secure Key Management and User Authentication Scheme for Fog Computing Services", *Future Generation Computer Systems*. 91. 10.1016/j.future.2018.09.017.

[24] Calheiros, Rodrigo N., et al. "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms", *Software: Practice and experience*, 41.1 (2011): 23-50.

[25] Gupta, H., V. Dastjerdi, A., Ghosh, S. K., & Buyya, R. "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in the Internet of Things, Edge and Fog computing environments", *Software: Practice and Experience*, 47(9), 1275-1296.

[26] S. Kumar, R. H. Goudar,"Cloud Computing – Research Issues, Challenges, Architecture, Platforms and Applications: A Survey" , *International Journal of Future Computer and Communication*, Vol. 1, No. 4, DOI 10.7763, December 2012.

[27] Att. URL https://www.business.att.com/solutions/Portfolio/cloud.

[28] Amazon web services. URL https://en.wikipedia.org/wiki/Amazon_Web_Services.

[29] CLAudit Project, Planetary-scale Cloud Latency Auditing Platform, http://bit. do/bS4js, 2016.

[30] M. Abrash, Latency – The Sine Qua Non of AR and VR, http : / / blogs .valvesoftware.com/abrash/latency-the-sine-qua-non-of-ar-and-vr/, 2012.

[31] OculusRift Blog, Delivers Some Home Truths On Latency | Oculus Rift Blog, https://oculusrift-blog.com/john-carmacks-message-of-latency/682/.

[32] S. Yangui et al., "A platform as-a-service for hybrid cloud/fog environments," in Proc. *IEEE Int. Symp*. Local Metropolitan Area Netw. (LANMAN), Rome, Italy, 2016, pp. 1–7.

[33] H. Atlam et al., "Fog Computing and the Internet of Things: a Review" , in: *Big Data and Cognitive Computing* 2.2 (2018).

[34] https://www.infosys.tuwien.ac.at/docs/03_Skarlat_FogFrame.pdf.

[35] N. Fernando, Seng W. Loke, and W. Rahayu, "Mobile Cloud Computing. *Future Generation Computer Systems*", 29(1):84–106, January 2013.

[36] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya., "Heterogeneity in Mobile Cloud Computing: Taxonomy and Open Challenges", *Communications Surveys and Tutorials*, 16(1):369–392, January 2014.

[37] Luis M. Vaquero and L. R. Merino, “Finding Your Way in the Fog: Towards a Comprehensive Definition of Fog computing”, *ACM SIGCOMM Computer Communication Review*, 44(5):27–32, October 2014.

[38] T. H. Dinh, C. Lee, D. Niyato and P. Wang., “A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches”, *Wireless Communications and Mobile Computing*, 13(18):1587–1611, February 2013.

[39] C. Project and Planetary-scale, “Cloud Latency Auditing Platform”, http://bit. do/bS4js, 2016.

[40] M. Abrash, “Latency – The Sine Qua Non of AR and VR, http : / / blogs . valvesoftware.com/abrash/latency- the- sine- qua- non- of- ar- and- vr/, 2012.

[41] O. Blog, “Delivers Some Home Truths On Latency | Oculus Rift Blog, https://oculusrift-blog.com/john-carmacks-message-of-latency/682/.

[42] X. Chen, “Decentralized Computation Offloading Game for Mobile Cloud Computing”, in: *IEEE Transactions on Parallel and Distributed Systems* 26.4 (2014).

[44] H. Atlam et al., “Fog Computing and the Internet of Things: a Review”, in: *Big Data and Cognitive Computing* 2.2 (2018).

[45] G. Ma et al., “Understanding Performance of Edge content Caching for Mobile Video Streaming”, in: Selected Areas in *Communications* 35.5 (2017).

[46] X. Meng et al., “Delay-Constrained Hybrid Computation Offloading with Cloud and Fog Computing”, in: IEEE Access 5 (2017).

[47] A. Rauf et al., “Security and Privacy for IoT and Fog Computing Paradigm” , in: *Proc. Learning and Technology Conference*, 2018.

[48] Open Fog Consortium Laboratories, Patient Monitoring, https : / / www .openfogconsortium.org/wp-content/uploads/Patient-Mo, 2019.

[49] B. Maharjan et al., “A Human-Centric Cloud and Fog-Assisted IoT Platform for Disaster Management”, in: Proc. *IEEE MobileCloud*, 2019.

[50] Mike J. Walker, B. Burton and M. Cantara, Gartner Hype Cycle 2018. https://www.gartner.com/doc/3383817, 2018.

[51] B. Burton and D. Willis, Gartner Mega Trends 2016. https://www. gartner.com/doc/3407820, 2016. A Cloud Computingessed: 2016-09-20.

[52] J. v. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, “Internet of Things (IoT): A Vision, Architectural Elements and Future Directions”, *Future Generation Computer Systems*, 29(7):1645 – 1660, September 2013.201-

[53] Wikipedia, Single Board Computers, https : / / en . wikipedia . org / wiki / Single-board_computer, 2019.

[54] N. Mohamed et al., “UAVFog: A UAV-based Fog Computing for Internet of Things” , in: Proc. *IEEE SmartWorld*, 2017.

[55] C. Zhu et al., “Vehicular Fog Computing for Video Crowdsourcing: Applications, Feasibility, and Challenges”, in*: IEEE Communications Magazine*, vol. 56, 10, 2018.

[56] E. Marín-Tordera et al., “Do We All Really Know What a Fog Node is? Current Trends Towards an Open Definition”, in: *Computer Communications* 109 (2017).

[57] K. Toczé and S. Nadjm-Tehrani, “ORCH: Distributed Orchestration Framework using Mobile Edge Devices”, in: *Proc. IEEE ICFEC,* 2019.

[58] A. Ahmed, “Efficient Cloud Application Deployment in Distributed Fog Infrastructures”, *Operating Systems* [cs.OS]. Université de Rennes 1, France, 2020. English.

[59] L. Vaquero, “Finding YourWay in the Fog: Towards a Comprehensive Definition of Fog Computing”, in: *ACM Computer Communication Review* 44.5 (2014).

[60] OpenFog Consortium, OpenFog Reference Architecture for Fog Computing, https://www.openfogconsortium.org/ra/, 2017.

[61] Standard Association, IEEE 1934-2018 -IEEE Standard for Adoption of Openfog Reference Architecture for Fog Computing, https://standards.ieee.org/ standard/1934-2018.

[62] M. B. Yassein, W. Mardini, and A. Khalil, “Smart homes automation using Z-wave protocol”, *International Conference on Engineering* & MIS (ICEMIS), 2016.

[63] Open Fog Consortium, Openfogconsortium.org, .https://www.openfogconsortium.org. A Cloud Computingessed: 12- Sep-2018.

[64] Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are. White Paper. 2016. Available online: http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing overview.pdf (accessed on 8 April 2018).

[65] Peter, N, “FOG Computing and Its Real Time Applications”, *Int. J. Emerg. Technol. Adv. Eng*. 2015, 5, 266–269.

[66] Nikoloudakis, Y.; Markakis, E.; Mastorakis, G.; Pallis, E.; Skianis, C, "An NF V-powered emergency system for smart enhanced living environments", In Proceedings of the *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks* (NFV-SDN), Berlin, Germany, 6–8 November 2017; pp. 258–263.

[67] A. V. Dastjerdi and R. Buyya, "Fog Computing: Helping the Internet of Things Realize Its Potential", in *Computer*, vol. 49, no. 8, pp. 112-116, doi: 10.1109/MC.2016.245.

[68] Yi, S., Hao, Z., Qin, Z., Li, Q., "Fog computing: Platform and applications", In Proceedings of the *3rd Workshop on Hot Topics in Web Systems and Technologies*, HotWeb 2015, Washington, DC, USA, 24–25 October 2016; pp. 73–78.

[69] Atlam, H.F.; Attiya, G.; El-Fishawy, N, "Integration of Color and Texture Features in CBIR System", *Int. J. Comput*. Appl. 2017, 164, 23–28.

[70] Adhatarao, S.S.; Arumaithurai, M.; Fu, X., "FOGG: A Fog Computing Based Gateway to Integrate Sensor Networks to Internet", In *Proceedings of the 29th International Teletraffic Congress*, Genoa, Italy, 5–7 September 2017; Volume 2, pp. 42–47.

[71] Kim, S.J.J., "A user study trends in augmented reality and virtual reality research: A qualitative study with the past three years of the ISMAR and IEEE VR conference papers", In *Proceedings of the 2012 International Symposium on Ubiquitous Virtual Reality*, ISUVR 2012, Daejeon, Korea, 22–25 August 2012; pp. 1–5.

[72] Dastjerdi, A.V.; Gupta, H.; Calheiros, R.N.; Ghosh, S.K.; Buyya, R., "Fog Computing: Principles, architectures, and applications", In *Internet of Things: Principles and Paradigms*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2016; pp. 61–75.

[73] https://doi.org/10.6028/NIST.SP.500-325.

[74] S. Kunal, A. Saha, R. Amin, "An overview of cloud-fog computing: Architectures, applications with security challenges," *Security and Privacy/Wiley*, 16 May 2019, https://doi.org/10.1002/spy2.72.

[75] Mahmud R., Kotagiri R., Buyya R. (2018) Fog Computing: A Taxonomy, Survey and Future Directions. In: Di Martino B., Li KC., Yang L., Esposito A. (eds) *Internet of Everything. Internet of Things (Technology, Communications and Computing). Springer*, Singapore. https://doi.org/10.1007/978-981-10-5861-5_5, 2017.

[76] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges," in *IEEE Communications Surveys & Tutorials*, vol.

20, no. 1, pp. 416-464, Firstquarter 2018, doi: 10.1109/COMST.2017.2771153.

[77] Vambe, William & Chang, Chii & Sibanda, Khulumani, “A Review of Quality of Service in Fog Computing for the Internet of Things”, *International Journal of Fog Computing*. 3. 22-40. 2020.

[78] Zheng, X., Da Xu, L., & Chai, “SQoS Recommendation in Cloud Services”, *IEEE Access*: Practical Innovations, Open Solutions, 5, 5171–5177. (2017).

[79] Brogi A and Forti S., “QoS-aware deployment of IoT applications through the fog”, *IEEE Internet Things* J. 2017;4(5):1185-1192.

[80] Jiang Y, Huang Z and Tsang DHK., “Challenges and solutions in fog computing orchestration”,. *IEEE Network*. 2017;1-8.

[81] Singh S and Chana I., “QoS-aware autonomic resource management in cloud computing: a systematic review”, *ACM Comput Surv*. 2015; 48(3):1-46.

[82] Jatoth C, Gangadharan GR and Buyya R., “Computational intelligence based QoS-aware web service composition: a systematic literature review”, *IEEE Transactions on Services Computing*. 2017;10(3):475-492.

[83] Neghabi AA, Navimipour NJ, Hosseinzadeh M and Rezaee A., “Load balancing mechanisms in the software defined networks: a systematic and comprehensive review of the literature”,. *IEEE Access*. 2018;6:14159-14178.

[84] Guevara J. C., Bittencourt L. F., and N. da Fonseca L. S., "Class of service in fog computing", in 2017 *IEEE 9th Latin-American Conference on Communications* (LATINCOM), 2017, pp. 1-6.

[85] https://www.lifewire.com/latency-on-computer-networks-818119

[86] Munir A, Kansakar P, Khan SU, “IFOG COMPUTING IoT: Integrated Fog Cloud IoT: A novel architectural paradigm for the future Internet of Things”, *IEEE Consumer Electronics Magazine*, 2017 Jul, 6(3):74-82.

[87] G. R. kumar, N. Saikiran and A. Sathish, “FOG: A novel approach for adapting IoT/IoE in cloud environment”, *International Journal of Engineering Trends and Technology*, (IJETT), vol. 42, pp. 189-192, Dec. 2016.

[88] V. Mihai, C. Dragana, G. Stamatescu, D. Popescu and L. Ichim, “Wireless Sensor Network Architecture based on Fog computing”, 2018 *5th International Conference on Control, Decision and Information Technologies*, (CoDIT’18), April 10-13, 2018.

[89] S. Yi, C. Li, and Qun Li. 2015, “A Survey of Fog Computing: Concepts, Applications and Issues”, In *Proceedings of the 2015 Workshop on Mobile*

Big Data, (Mobidata '15). ACM, New York, NY, USA, 37-42. DOI: https://doi.org/10.1145/2757384.2757397.

[90] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwalder, and B. Koldehofe, “Opportunistic Spatiotemporal Event Processing for Mobile Situation Awareness”, in *Proceedings of the ACM international conference on Distributed event-based systems*, pp. 192 -206, 2013.

[91] M. Malensek, S. L. Pallickara, and S. Pallickara, “HERMES: Federating fog and cloud domains to support query evaluations in continuous sensing environments*”, IEEE Cloud Comput*, vol. 4, no. 2, pp. 54–62, Mar./Apr. 2017.

[92] C. Fricker, F. Guillemin, P. Robert and G. Thompson, “Analysis of an offloading scheme for data centers in the framework of fog computing”, *ACM Transition Model Perform Eval Computing System*, vol. 1, no. 4, pp. 1–18, Sep. 2016.

[93] C. Li, H. Zhuang, Q. Wang and X. Zhou, “SSLB: Self-Similarity- Based Load Balancing for Large-Scale Fog computing”, *Arabian Journal for Science and Engineering*, 7487–7498. 10.1007/s13369-018-3169-3. February 2018.

[94] M. Al-khafajiy, T. Baker and H. Al-Libawy, “Improving Fog computing performance via Fog-2-Fog collaboration”, *Future Generation Computer Systems*, https://doi.org/10.1016/j.future. May 2019.

[95] B. Ottenwälder, B. Koldehofe, K. Rothermel and U. Ramachandran, “Operator migration for mobility driven distributed complex event processing”, in: Proceedings of the *7th ACM International Conference on Distributed Event-based Systems*, ACM, 2013, pp. 183–194.

[96] Q. Fan and N. Ansari, "Towards Workload Balancing in Fog Computing Empowered IoT", in *IEEE Transactions on Network Science and Engineering*. doi: 10.1109/TNSE.2852762, 2018.

[97] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang , “Cost efficient resource management in Fog computing supported medical cyber physical system”, *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 1, pp. 108–119, Jan 2017.

[98] Q. Fan, N. Ansari, and X. Sun, “Energy driven avatar migration in green cloudlet networks”, *IEEE Communications Letters*, vol. 21, no. 7, pp. 1601–1604, 2017.

[99] Istabraq M. A. Joboury, and Emad H. A. Hemiary, “IoT-F2CDM-LB: IoT Based Fog- to-Cloud and Data-in-Motion Architectures with Load Balancing”, *EAI Endorsed Transactions on Internet of Things*, Volume. 4, Issue 13, doi: 10.4108/eai.6-4-2018.155332, 23 August 2018.

[100] R. Deng, R. Lu, C. Lai, T. H. Luan and H. Liang, "Optimal Workload Allocation in Fog-Cloud Computing Toward Balanced Delay and Power Consumption" , in *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1171-1181,
doi: 10.1109/JIOT.2016.2565516, Dec. 2016..

[101] M. Verma, N. Bhardwaj and A. K. Yadav, "Real Time Efficient Scheduling Algorithm for Load Balancing in Fog computing Environment", *I.J. Information Technology and Computer Science,* 4, 1-10. DOI: 10.5815/ijitcs.2016.04.01.

[102] X. Wu, M. Deng, R. Zhang, B. Zeng and S. Zhou, "A Task Scheduling Algorithm Based on QOS-driven in Cloud Computing", *Proc. Comput. Sci*, 1162–1169, 2013, http://dx.doi.org/10.1016/j.procs.2013.05.148.

[103] F. Ramezani, J. Lu, F.K. Hussain, "Task-based system load balancing in cloud computing using particle swarm optimization", *Int. J. Parallel Program*, 42 (5), 739–754, 2014, http://dx.doi.org/10.1007/s10766-013-0275-4.

[104] S. Ningning, G. Chao, A. Xingshuo and Z. Qiang, "Fog computing dynamic load balancing mechanism based on graph repartitioning*", China Communications,* Vol. 13, No. 3, (2016) 156–164, http://dx.doi.org/10.1109/CC.2016.7445510.

[105] X. He, Z. Ren, C. Shi, and J. Fang, "A Novel Load Balancing Strategy of Software-defined Cloud/Fog Networking in the Internet of Vehicles," *China Communications*, vol. 13, no. 2, pp. 140–149, 2016.

[106] K. E. Parsopoulos, M. N. Vrahatis, "Particle Swarm Optimization Method for Constrained Optimization Problems", *Intelligent Technologies–Theory and Application: New Trends in Intelligent Technologies*, vol. 76, no. 1, pp. 214–220, 2002.

[107] R. Beraldi, A. Mtibaa, and H. Alnuweiri, "Cooperative Load Balancing Scheme for Edge Computing Resources", in *2nd International Conference on Fog and Mobile Edge Computing, IEEE,* 2017, pp. 94–100.

[108] D. Puthal, M. S. Obaidat, P. Nanda, M. Prasad, S. P. Mohanty, and A. Y. Zomaya, "Secure and Sustainable Load Balancing of Edge Data Centers in Fog Computing", *IEEE Communications Magazine,* vol. 56, no. 5, pp. 60–65, 2018.

[109] Da Silva, R.A.C.; da Fonseca, N.L.S. , "Resource Allocation Mechanism for a Fog-Cloud Infrastructure", In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6. [CrossRef]

[110] Mahmud, R.; Ramamohanarao, K.; Buyya, R., “Latency-aware Application Module Management for Fog Computing Environments”, *ACM Trans. Internet Technol*. 2019, 19, 9. [CrossRef]

[111] T. H. Ashrafi, M. A. Hossain, S. E. Arefin, K. D. J. Das, and A. Chakrabarty, “Service Based FOG Computing Model for IoT”, In: 2017 *IEEE 3rd International Conference on Collaboration and Internet Computing* (CIC). 2017.

[112] I.-H. Hou, T. Zhao, S. Wang, and K. Chan, “Asymptotically optimal algorithm for online reconfiguration of edge-clouds” , in Proc. *17th ACM Int. Symp. Mobile Ad Hoc Netw. Comput*., 2016, pp. 291–300.

[113] C.H. Hong and B. Varghese, “Resource Management in Fog/Edge Computing: A Survey on Architectures”, *Infrastructure, and Algorithms. ACM Comput. Surv*, 52, 5, 37 pages. DOI: https://doi.org/10.1145/3326066, September 2019.

[114] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos, “Challenges and Opportunities in Edge Computing”, in *IEEE International Conference on Smart Cloud*, 2016, pp. 20–26.

[115] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges”, *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[116] O. Skarlat, S. Schulte, Borkowski, M. Leitner, “Resource provisioning for IoT services in the fog”, In Proceedings of the *IEEE 9th International Conference on Service- Oriented Computing and Applications*, (SOCA), Macau, China, 4–6 November 2016, pp. 32–9.

[117] Aazam and M. Huh E. “Dynamic Resource Provisioning through Fog micro Datacenter”, In Proceedings of the *IEEE international conference on pervasive computing and communication workshops*, St. Louis, MO, USA, 23–27, pp. 105–110, March 2015.

[118] A. Brogi, S. Forti, and A. Ibrahim, “How to best deploy your fog applications, probably”, in Proceedings of the 1st IEEE International Conference on Fog and Edge Computing, ICFEC 2017, pp. 105–114, 2017.

[119] R. Mahmud, R. Kotagiri, and R. Buyya, “Fog computing: A taxonomy, survey and future directions”, in *Internet of Everything*. Springer, 2018, pp. 103–130.

[120] D. Zeng,Gu, L. Guo, S. Cheng and Z. Yu.,“Joint optimization of task scheduling and image placement in Fog computing supported software-defined embedded system”, *IEEE Trans. Comput*, 2016, 65, 3702–3712.

[121] Souza, V. Barbosa, C. Ramírez W, Masip-Bruin, X. M. Tordera, E. Ren and G. Tashakor, G., “Handling service allocation in combined fog-cloud scenarios”, In Proceedings of the *IEEE International Conference on*

Communications (ICloud Computing), Kuala Lumpur, Malaysia, 22–27, May 2016; pp.15.

[122] W. Lee, K. Nam, H.-G. Roh, and S.-H. Kim, “A gateway based fog computing architecture for wireless sensors and actuator networks”, in Proc. *18th Int. Conf. Adv. Commun, Technol*. (ICACT), 2016, p. 1.

[123] J. Oueis, E. C. Strinati, S. Sardellitti, and S. Barbarossa, “Small cell clustering for efficient distributed fog computing: A multi-user case”, in Proc. *IEEE 82nd Veh. Technol. Conf*, (VTC Fall), Boston, MA, USA, 2015, pp. 1–5.

[124] T. Nishio, R. Shinkuma, T. Takahashi and N. B. Mandayam, “Service oriented heterogeneous resource sharing for optimizing service latency in mobile cloud,” in Proc. *1st Int Workshop Mobile Cloud Comput*, Netw., Bengaluru, India, 2013, pp. 19–26.

[125] R. Brzoza-Woch, M. Konieczny, P. Nawrocki, T. Szydlo, and K. Zielinski, “Embedded systems in the application of fog computing Levee monitoring use case”, in Proc. 11th *IEEE Symp*. Ind. Embedded Syst, (SIES), Kraków, Poland, 2016, pp. 1–6.

[126] Fog X. Xu, F. Shucun, C. Qing et al., “Dynamic resource allocation for load balancing in fog environment”, *Wireless Communications & Mobile Computing*, vol. 2018, no. 2, Article ID 6421607, 15 pages, 2018.

[127] L. Ni, J. Zhang, C. Jiang, C. Yan, and K. Yu, “Resource allocation strategy in fog computing based on priced timed petri nets”, *Ieee Internet of tings Journal*, vol. 4, no. 5, pp. 1216–1228, 2017.

[128] Lin Y, Shen H., “CloudFog: leveraging fog to extend cloud gaming for thin-client MMOG with high quality of service”, *IEEE Transactions on Parallel and Distributed Sy*. 2017;28(2):431-445.

[129] A. Yousef pour., "FOGPLAN: A Lightweight QoS-Aware Dynamic Fog Service Provisioning Framework" , in *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5080-5096, June 2019, doi: 10.1109/JIOT.2019.2896311.

[130] Myrizakis, G., Petrakis, E.G.M., “Home: smart home management as a service in the cloud and the fog”, In: *Advanced Information Networking and Applications, Springer International* Publishing, pp. 1181–1192.2020.

[131] Prabhdeep S and Rajbir K, “Design and develop Quality of Service Framework using Fog Computing for Smart City Applications”, *International Journal of Innovative Technology and Exploring Engineering* (IJITEE) ISSN: 2278-3075, Volume-9 Issue-1S, November 2019.

[132] Skarlat, O. Nardelli, M. Schulte and S. Dustdar, “Towards qos-aware fog service placement”, In “Proceedings of the 1st *International Conference on Fog and Edge Computing* (ICFEC)”, Madrid, Spain, 14–15, pp. 89–96, May 2017.

[133] Won-Suk Kim and Sang-Hwa Chung, "User incentive model and its optimization scheme in user-participatory fog computing environment", *Computer Networks*, Volume 145, Pages 76-88, ISSN 1389-1286, 2018.

[134] M. K. Tefera, X. Yang and Q. T. Sun, "A Survey of System Architectures, Privacy Preservation, and Main Research Challenges on Location-Based Services", *KSII Transactions on Internet and Information Systems*, vol. 13, no. 6, pp. 3199-3218, 2019. DOI: 10.3837/tiis.2019.06.024.

[135] H. Inaltekin, M. Gorlatova, and M. Chiang, "Virtualized control over fog: Interplay between reliability and latency", *arXiv*:1712.00100v2, Feb. 2018. [Online]. Available: http://arxiv.org/abs/1712.00100.

[136] B. K. Dar, M. A. Shah, S. U. Islam, C. Maple, S. Mussadiq and S. Khan, "Delay-Aware Accident Detection and Response System Using Fog Computing", in IEEE Access, vol. 7, pp. 70975-70985, 2019. doi: 10.1109/ACCESS.2019.2910862.

[137] A. Sambola, M. Fernández, E. Hinojosa, and Y. Puerto, J., "The multi-period incremental service facility location problem", *Comput. Oper*. Res. 2009, 36, 1356–1375. [CrossRef].

[138] Zhu, J., Chan, D.S. Prabhu, M.S. Natarajan, P. Hu, H. and Bonomi, F., "Improving web sites performance using edge servers in fog computing architecture", In Proceedings of the 2013 *IEEE 7th International Symposium on Service Oriented System Engineering (SOSE)*, San Francisco Bay, CA, USA, 25–28 March 2013; pp. 320–323.

[139] Li, H.; Shou, G. Hu and Y. Guo, Z., "Mobile edge computing: Progress and challenges", In Proceedings of the 2016 4th *IEEE International Conference on Mobile Cloud Computing, Services, and Engineering* (MobileCloud), Oxford, UK, 29 March–1 April 2016; pp. 83–84.

[140] Jiang Y, Huang Z, Tsang DHK., "Challenges and solutions in fog computing orchestration", *IEEE Network*. 2017;1-8.

[141] H. Kashani, M. Rahmani, AM and J. Navimipour, N. , "Quality of service-aware approaches in fog computing", *Int J Commun Syst*. 2020; 33:e4340. https://doi.org/10.1002/dac.434039.

[142] Mishra, M. Roy and S.K. Mukherjee, A., "An energy-aware multi-sensor geo-fog paradigm for mission critical applications", *J Ambient Intell Human Comput*, Sep-2019. 41.

[143] H. Aljazzar and S. Leue, "K∗: A heuristic search algorithm for finding the k shortest paths", *Artif. Intell*. 2011.

[144] Ramírez W, Masip-Bruin X, Marin-Tordera E, Souza VB, Jukan A, Ren GJ, de Dios OG., "Evaluating the benefits of combined and continuous Fog-to-Cloud architectures". *Computer Communications*, 2017 Nov 15;113:43-52.

[145] Y. pour A, Ishigaki G and Jue JP., "Fog computing: Towards minimizing delay in the internet of things",. InEdge Computing (EDGE), 2017 *IEEE International Conference* on 2017,Jun 25 (pp. 17-24). IEEE.

[146] Han Z, Fan W, Li J and Xu M., "A Novel UDT-Based Transfer Speed-Up Protocol for FOG COMPUTING", *Wireless Communications and Mobile Computing*. 2018;2018.

[147] Naranjo PG, Shojafar M, Abraham A, "Ba Cloud Computing are lliE. A new stable election-based routing algorithm to preserve aliveness and energy in fog-supported WSNs", In *Systems, Man, and Cybernetics* (SMC), IEEE International Conference, Oct 9-2016, (pp. 002413-002418).

[148] Bhargava K and Ivanov S.,"A Fog computing approach for localization in WSN. In Personal, Indoor, and Mobile Radio Communications (PIMRC)", *IEEE 28th Annual International Symposium* on 2017 Oct 8 (pp. 1-7). IEEE.

[149] N.K. Giang , M. Blackstock , R. Lea , V.C. Leung , "Developing IoT applications in the fog: a distributed dataflow approach", in: 2015 5th International Conference on the Internet of Things (IOT), IEEE, 2015, pp. 155–162 .

[150] Zhang G and Li R., "Fog computing architecture-based data acquisition for WSN applications", *China Communications*. 2017 Nov;14(11):69-81.

[151] Aazam M, St-Hilaire M, Lung CH, Lambadaris I., "PRE-Fog: IoT trace based probabilistic resource estimation at Fog", In *Consumer Communications & Networking* Conference (CLOUD COMPUTINGNC), 13th IEEE Annual 2016 Jan 9 (pp. 12-17).

[152] Daniluk K., "Smart Decision FOG COMPUTING Layer in Energy-Efficient Multi-hop Temperature Monitoring System using WSN", In *Fed CSIS* Position Papers 2015 (pp. 167-172).

[153] Li W, Santos I, Delicato FC, Pires PF, Pirmez L, Wei W, Song H, Zomaya A, Khan S. , "System modelling and performance evaluation of a three-tier Cloud of Things", *Future Generation Computer Systems*. 2017 May 1;70:104-25.

[154] Fog X. Xu, F. Shucun, C. Qing et al., "Dynamic resource allocation for load balancing in fog environment", *Wireless Communications & Mobile Computing*, vol. 2018, no. 2, Article ID 6421607, 15 pages, 2018.

[155] Ning, Zhaolong, J. Huang and X. Wang., "Vehicular Fog computing: Enabling real-time traffic management for smart cities", *IEEE Wireless Communications*, 26, no. 1, pp. 87-93, 2019.

[156] Yi, Shanhe, Z. Qin and Q. Li., "Security and Privacy Issues of Fog Computing: A survey", In International Conference on Wireless Algorithms, Systems, and Applications, pp. 685-695. Springer, Cham, 2015.

[157] S. Tuli, R. Mahmud, S. Tuli and R. Buyya, “FogBus: A Blockchain-based Lightweight Framework for Edge and Fog Computing*”, Journal of Systems and Software*, Volume 154, 2019, Pages 22-36, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2019.04.050.

[158] A. Kapsalis, P. Kasnesis, I. S. Venieris, D. I. Kaklamani, and C. Z. Patrikakis, “A cooperative fog approach for effective workload balancing”, *Cloud Computing.*, vol. 4, no. 2, pp. 36–45, Mar./Apr. 2017.

[159] R. M.Vozmediano, R. S. Montero, E. Huedo, and I. M. Llorente, “Cross-site virtual network in cloud and fog computing”, *IEEE Cloud Computing*, vol. 4, no. 2, pp. 46–53, Mar./Apr. 2017.

[160] X. Hou et al., “Vehicular fog computing: A viewpoint of vehicles as the infrastructures”, IEEE Trans. Veh. Technol, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.

[161] Y. Yan and W. Su, “A fog computing solution for advanced metering infrastructure”, in Proc. *IEEE/PES Transm*. Distrib. Conf. Expo. (T D), Dallas, TX, USA, 2016, pp. 1–4.

[162] M. Aazam and E.N. Huh, “E-HAMC: Leveraging fog computing for emergency alert service”, in Proc. *IEEE Int. Conf. Pervasive Comput. Commun*. Workshops (PerCom Workshops), St. Louis, MO, USA, 2015, pp. 518–523.

[163] K. Intharawijitr, K. Iida and H. Koga, “Analysis of fog model considering computing and communication latency in 5G cellular networks”, in Proc. *IEEE Int. Conf. Pervasive Comput. Commun*. Workshops (PerCom Workshops), Sydney, NSW, Australia, 2016, pp. 1–4.

[164] R. Deng, R. Lu, C. Lai, and T. H. Luan, “Towards power consumption delay tradeoff by workload allocation in cloud-fog computing” , in Proc. *IEEE Int. Conf. Commun. (ICC)*, London, U.K., 2015, pp. 3909–3914.

[165] M. Aazam and E.-N. Huh, “Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT” , in Proc. *IEEE 29th Int. Conf. Adv. Inf. Netw. Appl.* (AINA), Gwangju, South Korea, 2015, pp. 687–694.

[166] D. Ye, M. Wu, S. Tang, and R. Yu, “Scalable fog computing with service offloading in bus networks”, in Proc. *IEEE 3rd Int. Conf. Cyber Security Cloud Computing. (CSCloud)*, Beijing, China, 2016, pp. 247–251.

[167] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, “Coding for distributed fog computing”, *IEEE Commun*. Mag., vol. 55, no. 4, pp. 34–40, Apr. 2017.

[168] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, “Fog computing may help to save energy in cloud computing” , *IEEE J. Sel. Areas Communication*, vol. 34, no. 5, pp. 1728–1739, May 2016.

[169] L. Gu, D. Zeng, S. Guo, A. Barnawi and Y. Xiang, “Cost efficient resource management in fog computing supported medical cyber physical system” , *IEEE Transition Emerging Topics Comput*, vol. 5, no. 1, pp. 108–119, Jan./Mar. 2017.

[170] Q. He, C. Zhang, X. Ma and J. Liu, “Fog-based transcoding for crowd sourced video live cast”, *IEEE Communication*, Mag, vol. 55, no. 4, pp. 28–33, Apr. 2017.

[171] M. Tang, L. Gao, H. Pang, J. Huang and L. Sun, “Optimizations and economics of crowd sourced mobile streaming”, *IEEE Communication*, Mag, vol. 55, no. 4, pp. 21–27, Apr. 2017.

[172] S. Sarkar and S. Misra, “Theoretical modelling of fog computing: A green computing paradigm to support IoT applications”, *IET Network*, vol. 5, no. 2, pp. 23–29, Mar. 2016.

[173] S. Agarwal, S. Yadav, and A. K. Yadav, “An efficient architecture and algorithm for resource provisioning in fog computing”, *Int. Journal of Information Engineering Electron*. Bus., vol. 8, no. 1, pp. 48–61, 2016. ID 6421607, 15 pages, Volume 2018.

[174] J. Santos, T. Wauters, B. Volckaert and De Turck, “Resource Provisioning in Fog Computing: From Theory to Practice”, *Sensors*, 2019, 19, 2238.

[175] Q. Shaheen, M. Shiraz, M.U. Hashmi, D. Mahmood, Z. zhiyu and R. Akhtar, "A Lightweight Location-Aware Fog Framework (LAFF) for QoS in Internet of Things Paradigm", *Mobile Information Systems*, vol. 2020, ArticleID 8871976, 15 pages, 2020. https://doi.org/10.1155/2020/8871976.

[176] Zhiming He, Y. Zhang, B. Tak and L. Peng, "Green Fog Planning for Optimal Internet-of-Thing Task Scheduling", *Access IEEE*, vol. 8, pp. 1224-1234, 2020.

[177] N. Siasi, A. Jaesim and N. Ghani, "Tabu Search for Efficient Service Function Chain Provisioning in Fog Networks", *Collaboration and Internet Computing* (CIC) 2019 IEEE 5th International Conference on, pp. 145-150, 2019.

[178] H. O. Hassan, S. Azizi and M. Shojafar, "Priority, network and energy-aware placement of IoT-based application services in fog-cloud environments", in *IET Communications*, vol. 14, no. 13, pp. 2117-2129, 11 8 2020, doi: 10.1049/iet-com.2020.0007.

[179] Cisco Inc., “Internet of Things (IoT) Data Continues to Explode Exponentially. Who Is Using That Data and How”, - Cisco Blog, https: //blogs. cisco. com/datacenter/internet- of- things- iot- data- continues- to-explode exponentially-who-is-using-that-data-and-how, 2019.