PERFORMANCE ANALYSIS OF SMART GATEWAYS IN FOG NETWORK

A Thesis

presented to

the Faculty of Natural and Applied Sciences

at Notre Dame University-Louaize

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

FADI Y. KHALIL

DECEMBER 2018

Notre Dame University - Louaize
Faculty of Natural and Applied Sciences
Department of Computer Science

We hereby approve the thesis of

Fadi Y. Khalil

Candidate for the degree of Master of Science in Computer Science

[Signature]	
Dr. Hoda Maalouf	Supervisor, Chair
[Signature]	
Dr. Maya Samaha Rupert	Committee Member
[Signature]	
Dr. Nazir Hawi	Committee Member

Declaration

I, Fadi Y. Khalil

☑ authorize the Notre Dame University Louaize to supply copies of my thesis to libraries or individuals upon request.

☐ do not authorize the Notre Dame University Louaize to supply copies of my thesis to libraries or individuals on request.

Signature

Date

December 20, 2018

Acknowledgement

I would like to express my sincere gratitude to my advisor, Dr. Hoda Maalouf, for the continuous support of my Master study, for her patience, motivation, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better and friendlier advisor.

I would also like to thank my family: my wife, my kids and my parents for their moral support throughout writing this thesis and my entire life.

Table of Contents

List of Figures

List of Tables

List of Code Listing

List of Abbreviation

Abstract

With the advancement in computing and wireless technologies, the world has witnessed a growing number of connected devices to the Internet at an unpretending rate. Fog computing emerges as an alternative solution to reduce the burden of Data Centers in traditional Cloud Computing and to support geographically distributed, latency sensitive, and QoS-aware IoT appliations. Fog Computing suggests, that the intelligence should move from the data centers to the edge of network.

The main objective of this thesis is to determine the significance and feasibility of implementing a broker node between the IoT devices and the Fog networks in a healthcare application. This thesis studies different implementation scenarios and analyzes the impact of a broker on latency. Moreover, it explains when it is necessary to deploy a broker and specifies the most crucial network parameters involved in the analysis.

This thesis analyzes both IoT-Fog and IoT-Broker-Fog scenarios in the context of processing delay according to the increasing number of generated packets and network service rate. Simulations of the different scenarios were done using the MATLAB version R2017a.

Keywords: IoT, Fog Computing, Cloud Computing, Broker Node, Smart Gateway.

Chapter 1: Introduction and Problem Definition

The Information and Communication Technology (ICT) infrastructures are expanding continuously with the growth in the number of devices and applications for Internet-of-Things (IoT)[1]. The traditional centralized cloud computing paradigm is no more efficient enough for latency sensitive multimedia services and other time-sensitive services, like emergency and healthcare. In contrast to Cloud, Fog, an emerging architecture for bringing processing, storage, and control from the Cloud closer to the Things/Users, can mitigate the issues traditional cloud cannot solve being standalone. Fog can provide services with faster response. Moreover, it can preprocess and filter data according to the requirements. Therefore, fog computing may be considered the best choice to enable the IoT to provide efficient and secure services for many IoT users.

1.1 Introduction to the General Problem

With the fast increment in the number of Internet of Things (IoT) devices, a huge amount of data is daily generated and should actually be processed and filtered before being sent to the cloud; otherwise, the whole IoT scheme would be highly challenging to implement. In this regard, researchers have proposed fog computing as a middle layer between IoTs and Cloud computing to solve this issue, specifically for real-time applications with low-delay constraints, such as healthcare solutions, which demand prompt response and processing. E-health applications have taken a remarkable lead in terms of services and

features in the past few years. Today, millions of people are motivated and have confidence to lead a healthier lifestyle by using such applications. In order to maximize the services' spectrum from these applications over cloud, several challenges like data privacy and communication cost need serious attention. Fog computing can be used as an intermediary layer between the cloud and end users in order to ease these issues.

1.2 Problem Definition

In the near past, E-health and wellness applications have emerged to improve human health and well-being. Thousands of people are now using smart gadgetries and wearable health devices to live a healthier lifestyle. Consequently, a high volume of data is being generated daily and the traditional centralized cloud computing paradigm faces several challenges in terms of high latency, central storage, and network failure. According to Cisco global cloud index, a city of one million people will produce 180 million gigabytes of data per day by 2019 [2]. In addition, every single minute, YouTube users uploads around 400hrs of new videos; Instagram users like 2,430,555 posts; Siri answers 99,206 requests; Snapchat users watch 6,944,444 videos; etc. [3]. Fog computing, a "new" computing paradigm where some of the computations take place in the fog devices, can minimize the unnecessary communication from data generating nodes to cloud. Fog computing as an intermediary layer between the cloud and end user plays its pivotal role in terms of low latency and context awareness.

Many studies have been conducted on Fog Computing so far, but only few used Fog computing with healthcare system. In addition, the number of papers that followed the concept of fog computing at the smart gateway is countable. To the best of our

knowledge, none of the available studies tackles the importance and efficiency of using a broker and when to deploy it in the IoT-Broker-Fog-Cloud scenarios.

1.3 Research Objective

The main purpose of this thesis is to show the advantage of using a broker as a choke point between Internet of Things (IoT) devices and Fog Network. Therefore, the main contribution of this study is to calculate and compare the latency between IoT device-Fog Node and IoT device-Broker-Fog Node for smart wearable health technologies, which is not considered and/or found yet in any literature.

1.4 Approach and Main Result

Our main approach in this project is to analyze both IoT-Fog and IoT-Broker-Fog scenarios in the context of processing delay according to increasing number of users, say number of generated packets and the network service rate. We accomplished this by an event-driven simulation using MATLAB R2017a.

This approach led to the following results:

- Creating an event-driven simulation
- Delay has been reduced in IoT-Broker-Fog approach when a broker is deployed
- Data Traffic to Fog is decreased
- Optimum Value of Batch period T has been found
- The deployment of Broker in the IoT-Broker-Fog scenarios depends on the type of application, the traffic in the network and on the network performance.

1.5 Thesis Organization

The structure of this thesis is organized as follows:

Chapter 1 is about the introduction and the problem definition of the thesis.

Chapter 2 starts off with the explanation of the IoT technology and cloud computing and then gives general definition of the basic concepts of Fog computing and related computing paradigm.

Chapter 3 is an intensive literature review of the theoretical framework. Also explored in this chapter are architectures such as Fog computing with Smart Gateway and Fogging for delay-sensitive applications.

Chapter 4 provides an in-depth and detailed explanation of the conducted simulation and the platform used in addition to the technique used to acquire the requirements needed.

Finally, chapter 5 comprises conclusions of all the work done based on the statistics and figures mentioned in the simulation. It also gives an insight into the future work in this area.

Chapter 2: Definition of the Basic Concepts

2.1 Internet of Things

The IoT is a currently growing technology trend discussed all over the world. The term IoT "the Internet of Things" was first used by Kevin Ashton [4],[5]. The Internet of Things (IoT) as defined by IERC is a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual "things" have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network [6].

The term "things" in IoT refers to any object on earth with a network connectivity, whether it is intelligent things such as sensors, wearable, mobile phones etc, or a dumb object, like food item, household item, medicine, etc. that can be part of the internet (Figure 2-1).

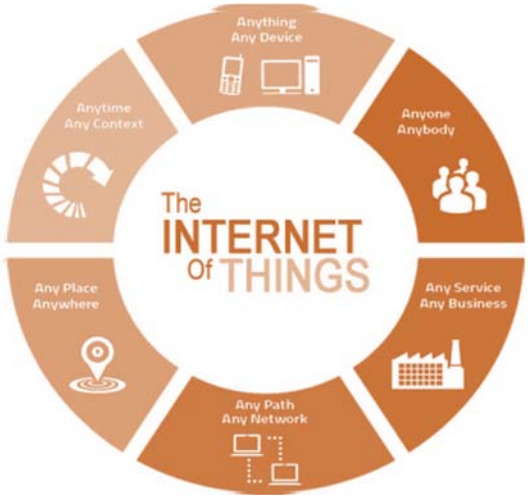


Figure 2-1: The Term "Internet of Things"[7]

The objective of IoT is to reach a stage at which many of the objects around us will have the ability to connect to the internet to communicate with each other without human intervention [8]. In order to connect the mentioned IoT devices to the Internet many technologies are applied, e.g. WiFi, Bluetooth, RFID, and Near Field Communication (NFC)[9]. For wide area links, there are existing mobile networks using GSM, GPRS, 5G for example and satellite connections.

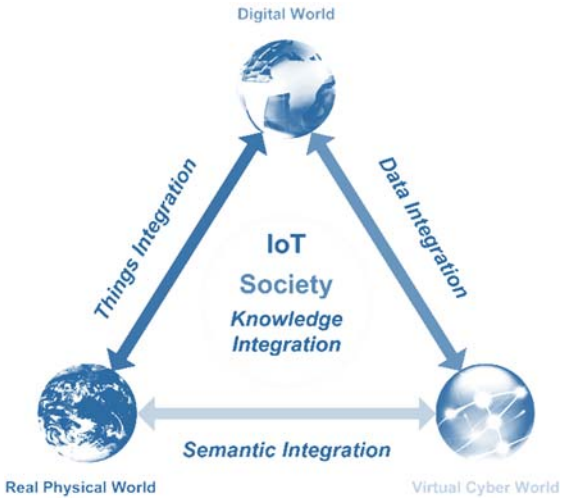


Figure 2-2: depicts IoT in a global context[6]

The Internet of things (IoT) is a very promising novel technology which has high impact on several aspects of everyday-life and user behavior and provides many benefits to our society. E-health, demotic, enhanced learning are some examples of possible application scenarios in which the IoT paradigm will play a significant role in the near future.

It is predicted that in the following decades, the way healthcare is currently provided will be transformed from hospital-centered, first to hospital-home-balanced by 2020, and then ultimately to home-centered by 2030 [10]. This transformation necessitates a lot of interest

and research attention of the IoT architectures and technologies for smart spaces and healthcare applications and for many diverse areas. Figure 2-2 depicts the IoT society.

2.2 Cloud Computing

Since the advent of the first computer in 1946, the Internet, WWW, and Cloud computing were among the few major milestones that have stood along the way of the development of the Information Technology. According to the official NIST definition, "cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [11]. This definition includes characteristics, service and deployment models (see Figure 2-3). Cloud service providers such as Microsoft, Amazon Web Servies (AWS), Google, Apple, Facebook, at&t [12] offer three Cloud Service Models as per the NIST publication 800-145 [11]: SaaS (Software as a Service), PaaS (Platform as a Service), IaaS (Infrastructure as a Service).

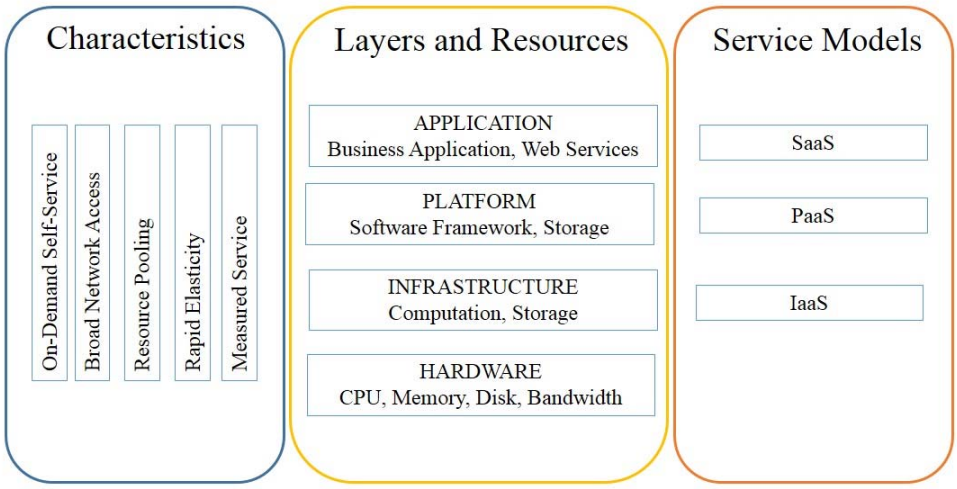


Figure 2-3: Cloud Computing Overview (Adapted from [13])

Flexibility, storing large amount of data and data recovery, ease of access, easy to share information, and "pay-as-you-go" computing model are some of the benefits that Cloud provides and consequently make it a necessity for innovative and daring organizations. As reported, around 90% of global Internet users are now relying on the services provided by cloud, either directly through consumer services or indirectly through their service provider's reliance upon different commercial clouds[11].

2.3 Fog Computing

2.3.1 Definition

Fog Computing has been defined by many organizations and researchers from different viewpoints. National Institute of Standards and Technology (NIST) defines Fog computing as a layered model for enabling ubiquitous access to a shared continuum of scalable

computing resources. The model facilitates the deployment of distributed, latency-aware applications and services, and consists of fog nodes (physical or virtual), residing between smart end-devices and centralized (cloud) services. [14].

In another way, fog computing is defined by the OpenFog Consortium as a horizontal, system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum [15].

Vaquero and Rodero-Merino [16] defined fog computing as "a scenario where a huge number of heterogeneous (wireless and sometimes autonomous) ubiquitous and decentralized devices communicate and potentially cooperate among them and with the network to perform storage and processing tasks without the intervention of third parties. These tasks can be for supporting basic network functions or new services and applications that run in a sandboxed environment. Users leasing part of their devices to host these services get incentives for doing so." This definition gives integrative view of fog computing but fails to point out the unique connection to the cloud.

Therefore, a more general definition comes in [17]. Fog computing is a geographically distributed computing architecture with a resource pool consisting of one or more ubiquitously connected heterogeneous devices (including edge devices) at the edge of network and not exclusively seamlessly backed by cloud services, to collaboratively provide elastic computation, storage and communication (and many other new services and tasks) in isolated environments to a large scale of clients in proximity.

F. Bonomi and R. Milito defined Fog Computing as follows: "Fog Computing is a highly virtualized platform that provides compute, storage, and networking services between end

devices and traditional Cloud Computing Data Centers, typically, but not exclusively located at the edge of network" [18].

Fog computing, also known as fog networking or fogging [19]. The concept of Fog computing was first introduced by Cisco in 2012 to extend the cloud computing to the edge of the network and to address the challenges of IoT services and applications in conventional Cloud paradigm [18]. The term "Fog" is used simply because "fog is a cloud close to the ground" [18].

2.3.2 Fog Computing-Based Architecture

The Fog computing acts as an intermediary between the cloud and smart end-devices (usually IoT devices) which brings the cloud computing and services closer to the end devices themselves. This accordingly leads to a three-layer ecosystem (Cloud-Fog-Thing). This architecture is a good compromise between fully centralized cloud networking and fully distributed fog networking.

Fog computing is not observed as a mandatory layer for such ecosystems nor is the centralized (cloud) service perceived as being required for a fog computing layer to support the functionality of smart end-devices. Different use case scenarios might have different architectures based on the optimal approach to supporting end-devices functionality [14]. However, the three-tier architecture [20] is one of the widely used architectures in fog computing. Figure 2-4 depicts the architecture.

- The first tier involves different things or end-devices including sensor nodes, smart devices such as tablets, smartphones, smart watch, and others. These devices are usually termed as Terminal Nodes (TNs) and equipped with GPS so that location-

based services are delivered to the TNs in real-time based on region-specific data analysis [20].

- The second tier is the fog computing layer. This layer consists of different type of fog nodes (FNs). Any device with computing, storage and network connectivity can be a fog node, such as industrial controllers, switches, routers, embedded servers and video surveillance cameras [21]. Traditional base station equipped with certain storing and computing capabilities are considered suitable for Fog Computing; for example, Road Side Unit (RSU) and small cell access point can be used as potential Fog nodes [22]. Vehicules at the edge of network with computation facilities can serve as Fog nodes [22]. Fog nodes can also be virtual components (e.g. virtualized switches, virtualized machines, etc.) [14]. This Fog layer can be formed by one or more fog domains and each fog domains encompass different FNs which can be controlled by the same or different providers. The communication inside the fog layer can be in any form of connections (e.g. WiFi, Bluetooth, or Ethernet) [16]. The communication between layer 1 and fog computing is done through Local Area Network (LAN) whereas the communication between layer 1 / layer 2 and cloud computing requires connection over the Wide Area Network (WAN).
- The third layer is the cloud computing layer where the cloud servers and data centers reside.

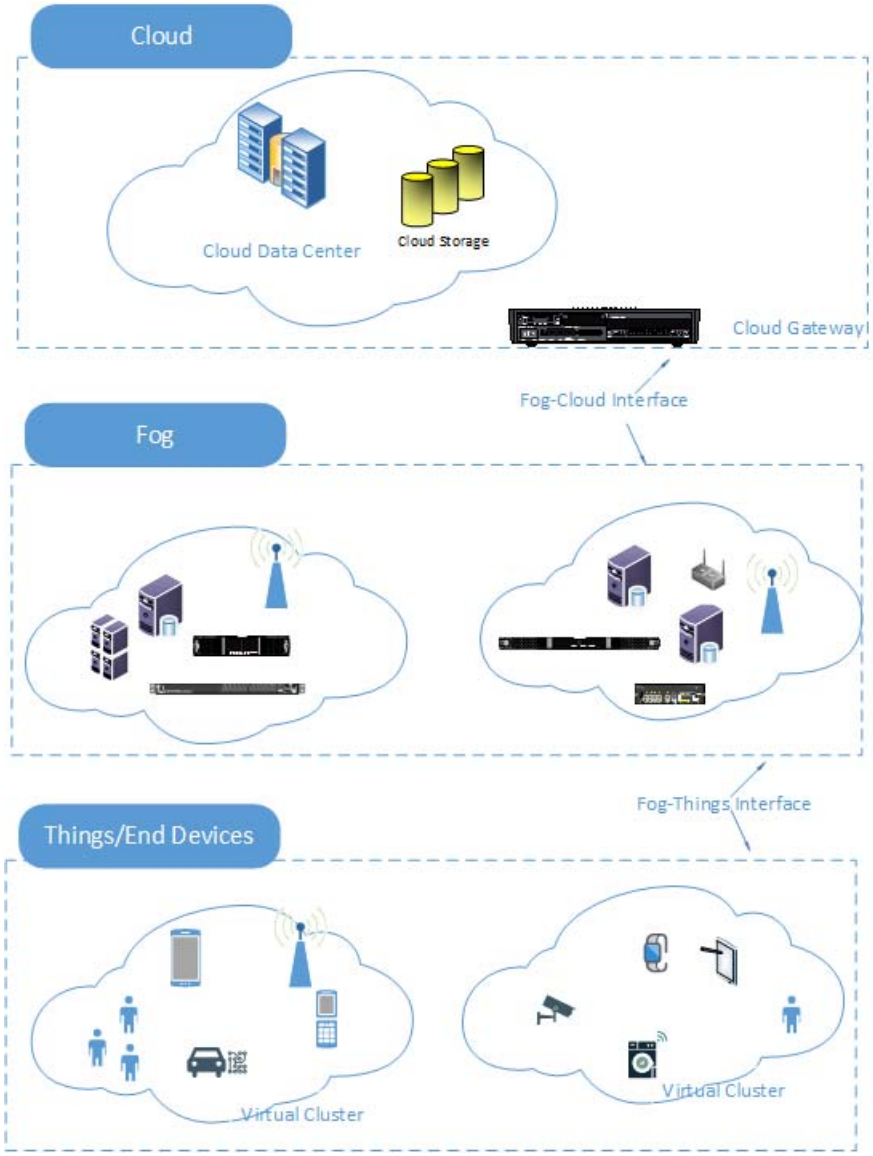


Figure 2-4: Three-tiers Fog Computing architecture (Adapted from [23])

According to Aazam, and Huh [24], the architecture of fog computing contains six layers which are physical and virtualization, monitoring, preprocessing, temporary storage, security and transport layer as shown in table 2-1.

The Physical and Virtualization layer contains physical TNs and virtual sensor nodes. Monitoring layer monitors the activities of the underlying nodes and networks. In addition, the energy consumption of fog nodes is monitored, so effective measures can be taken in

time. Preprocessing layer analyses the collected data, performs data filtering, trimming, and in the end, more meaningful and necessary data is generated.

The data is stored temporary in the Temporary Storage layer. When the data are transmitted to the cloud, they no longer need to be stored locally and may be removed from the temporary storage media. The security-related issues are handled in the security layer. Finally, the transport layer is responsible for uploading data to the Cloud.

Table 2-1: Fog Layers Architecture (adapted from [23])

Layers	Description
Transport Layer	Sends data to Cloud
Security Layer	Handles security related issue
Temporary Storage Layer	Stores the data temporarily
Processing Layer	Data filtering and trimming
Monitoring Layer	Handles service requests and envery consumption
Physical and Virtualization Layer	Contains TNs and Virtual sensor node

2.3.3 Fog Computing Essential Characteristics

Fog computing is considered to be the building blocks of the cloud. According to [14], [17], and [25], the following nine characteristics are essential in distinguishing fog computing form other computing paradigms. However, a TNs or IoT user is not required to make use of all characteristics when consuming a fog computing service [14].

1. **Location awareness and low latency**: Fog computing supports location awareness in which fog nodes can be positioned in different locations. Because FN are often closer to smart end-devices, Fog Computing provides lower latency when processing the data of these devices than from a centralized cloud service.

2. **Geographical distribution**: The services and applications provided by the fog are distributed and can be deployed anywhere in contrast to the traditional centralized cloud.
3. **Scalability**: The fog provides distributed computing and storage resources which can work with large-scale networks. Fog computing is adaptive in nature supporting elastic compute, resource pooling, data-load changes just to name a few of the supported adaptive functions.
4. **Support for mobility**: Fog applications has the ability to connect directly to mobile devices and therefore support mobility techniques, such as the locator ID separation protocol (LISP) which needs a distributed directory system.
5. **Real-time interactions**: Fog computing applications afford real-time interactions between fog nodes rather than the batch processing employed in the cloud.
6. **Heterogeneity**: Fog computing supports collection and processing of data from different Fog nodes or end devices which are designed by different manufacturers.
7. **Interoperability**: Fog components can interoperate and work with different domains and across different service providers.
8. **Support for on-line analytics and interplay with the cloud**: The fog is and intermediate layer between the cloud and end devices in order to play a significant role in the absorption and processing of the data close to the end devices.
9. **Predominance of wireless access**: Although fog computing is used in wired environments, the large scale of wireless sensors in IoT demand distributed analytics and compute. For this reason, fog computing is very well suited to wireless IoT access networks.

2.3.4 Fog Node Attributes

The fog node is the core component of the fog computing architecture. Fog nodes are either physical components or virtual components, as we have previously mentioned, that are tightly coupled with the smart end-devices or access networks, and provide computing resources to these devices. Fog nodes need to support one or more of the following attributes in order to simplify the deployment of a fog computing capability that exhibits the characteristics described in section 2.3.3 [14]:

- **Autonomy:** Fog nodes can operate independently, making local decisions, at the node or cluster-of-nodes level.
- **Heterogeneity:** Fog nodes come in different form factors, and can be deployed in a wide variety of environments.
- **Hierarchical clustering**: Fog nodes support hierarchical structures, with different layers providing different subsets of service functions while working together as a continuum.
- **Manageability**: Fog nodes are managed and orchestrated by complex systems that can perform most routine operations automatically.
- **Programmability**. Fog nodes are inherently programmable at multiple levels, by multiple stakeholders - such as network operators, domain experts, equipment providers, or end users.

2.3.5 Fog Computing Service Models

Similar to the cloud computing service models defined in NIST [11], the following types of service models can be applied:

- Software as a Service (SaaS): The capability provided to the fog service customer is to use the fog provider's applications running on a cluster of federated fog nodes managed by the provider. This type of service is similar to the cloud computing Software as a Service (SaaS) and implies that the end-device or smart thing accesses the fog node's applications through a thin client interface or a program interface. The end-user does not manage or control the underlying fog node's infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.
- Platform as a Service (PaaS): The capability provided to the fog service customer is similar to the cloud computing Platform as a Service (PaaS) and allows deployment onto the platforms of federated fog nodes forming a cluster, of customer-created or acquired applications created using programming languages, libraries, services, and tools supported by the fog service provider. The fog service customer does not manage or control the underlying fog platform(s) and infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.
- Infrastructure as a Service (IaaS): The capability provided to the fog service customer is to provision processing, storage, networks, and other fundamental computing resources leveraging the infrastructure of the fog nodes forming a federated cluster. Similar to cloud computing Infrastructure as a Service (IaaS) services, the customer is able to deploy and run arbitrary software, which can

include operating systems and applications. The consumer does not manage or control the underlying infrastructure of the fog nodes cluster but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

2.3.6 Related Computing Paradigm

Several computing paradigms related to edge and fog concept have already been introduced. Cyber foraging, Cloudlet, Mobile/Multi-Access Edge Computing (MEC) are among them.

Fog computing is often inaccurately confused with edge computing, but there are key differences between the two concepts [26], [27]. Fog works with the cloud, whereas edge is defined by the exclusion of cloud. Fog is hierarchical, where edge tends to be limited to a small number of layers. In additional to computation, fog also addresses networking, storage, control and acceleration [15]. OpenFog consortium describes fog computing as a superset of edge computing and in the IT and IoT food chain, they describe it as: edge is to fog, as apple is to fruit [27].

- The fundamental idea of **Edge Computing** is to bring the computation facilities closer to the source of the data. More precisely, Edge computing enables data processing at the edge network [28]. In other words, it is about pushing intelligence to the edge of the network. Edge computing can be traced back to the 1990s, when Akamai launched its Content Distribution Network (CDN), which introduced nodes at locations geographically closer to the end user [29]. Edge network basically consists of end devices (e.g. mobile phone, smart objects, etc.), edge devices (e.g. border routers, base stations, wireless access points etc.), edge servers, etc. and

these components can be equipped with necessary capabilities for supporting edge computation [22]. In addition, Edge computing concentrate more toward the end devices side rather than providing cloud based services [30]. Table 2-2 depicts the computing paradigms evolution.

Table 2-2: Computing Paradigm Evolution

Computing Paradigm	Year
Edge Computing	1990s
Cyber Foraging	2001-2002
Cloud Computing	2006
Cloudlet	2009
Fog Computing	2012
MEC	2014-2017

- **Cyber foraging** was first introduced by Satyanarayanan [31] in 2001, and was further refined in 2002 [32]. Cyber foraging constructed as "living off the land", may be an effective way to extend the battery life of mobile phones while preserving their computing and data manipulation capabilities [31]. The idea is that mobile devices can dynamically exploit the capabilities of nearby servers, connected to the internet through high-bandwidth networks. These servers are called surrogates and perform computing and data staging which is the process of prefetching distant data to nearby surrogates.

  In cyber foraging, when a mobile device has to perform an intensive computation for accessing a large volume of data, the mobile device offload the complex processing to a surrogate. The latter may stage the database on its local disk and perform the whole or some part of the processing on behalf of the mobile device. It

then delivers the result to the mobile device with low latency, since it is close to the device.

- **Cloudlet paradigm** was proposed next in 2009 [33]. It is also referred to as cloudlet-based cyber foraging [34]. Cloudlets use virtualization on their resource-rich servers that are located in a single-hop proximity of mobile devices. These servers run one or more Virtual Machines (VMs) in which the computation of the mobile devices is offloaded. Similar to WiFi concept, cloudlets provide cloud services to the mobile users instead of proving Internet connectivity as in WiFi. Also, the cloudlets exist as a standalone environment, since VM provisioning of the cloudlets is done without cloud intervention. Hence, Quality of Service (QoS), of the mobile devices is hard to fulfill since the cloudlets are not an inherent part of the mobile network and coverage of WiFi is only local with limited support of mobility [35]. It is worth to note that in several works [36],[37], cloudlets are mentioned as Fog nodes.
- **Mobile/Multi-Access Edge Computing (MEC)** was initiated by the Industry Specification Group (ISG) within the European Telecommunication Standards Institute (ETSI) in 2014 under the name of Mobile Edge Computing, with the focus on mobile networks and VM as virtualization technology. In March 2017, ETSI has expanded the scope to encompass non-mobile network requirements and after that replaced the term “Mobile” by “Multi-Access”. The fundamental goal of the ETSI initiative is to create a “standardized, open environment which will allow the efficient and seamless integration of applications from vendors, service providers, and third-parties across multi-vendor Multi-access Edge Computing platforms [38].

Mobile Edge Computing has been regarded as one of the key enablers of modern evolution of cellular base stations. Mobile Edge Computing provides an IT service environment and cloud computing capabilities at the edge of the mobile network, within the Radio Access Network (RAN) and in close proximity to mobile subscribers. The aim is to reduce latency, ensure highly efficient network operation and service delivery, and offer an improved user experience [39].

Figure 2-5 Provides a Venn diagram that summarizes the similarities and differences between cloudlets, MEC, and fog computing.

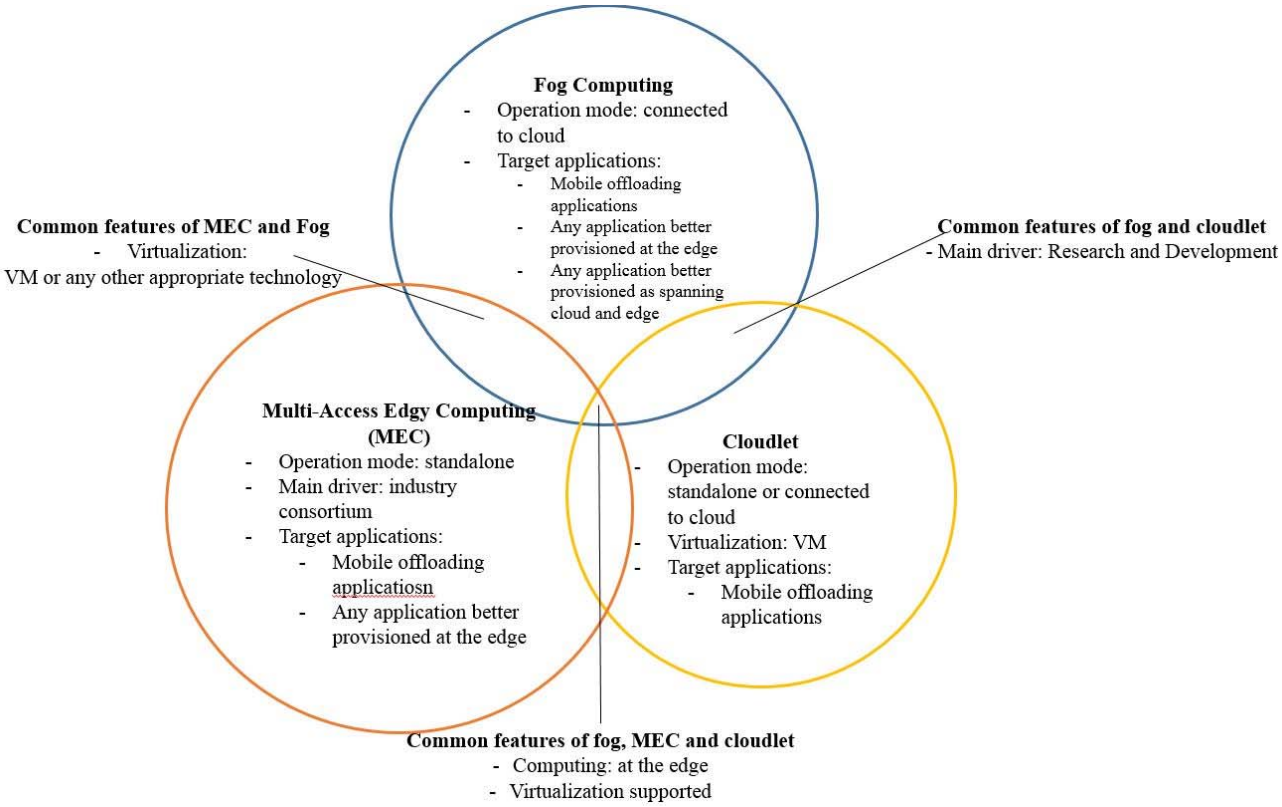


Figure 2-5: A Venn diagram showing the Relationship between Fog Computing, Cloudlet, and MEC (Adapted from [40])

2.3.7 Challenges in Fog Computing

Fog computing is considered as the extension of Cloud computing paradigm to handle IoT-related issues at the edge of network. However, many challenges arise by analyzing the features of Fog computing from structural, service-oriented and security perspectives. These challenges can be categorized as follows [22]:

- **Structural perspective:**
    - Provisioning both the edge and core network components with some sort of computation besides their traditional activities will be very challenging.
    - The selection of suitable nodes and the places of deployment are vital in Fog computing as well.
    - Identification of appropriate techniques, metrics, etc. for node-to-node collaboration and resource provisioning is important.
    - Competency assurance of Fog computing in other networking systems such as a Content Distribution Network (CDN) will be very challenging.
- **Service-oriented perspective:**
    - Potential programming platform for distributed applications development in Fog computing is required to be introduced due to the fact that not all Fog nodes are resource enriched.
    - Specifications of the policies to distribute computational tasks and services among IoT devices/sensors, Fog and Cloud infrastructures are needed.
    - In different scenarios, it is quite difficult to specify the Service Level Agreement (SLA) due to many factors such as, energy usage, network status, service cost, etc.

- **Security perspective:**
    - Fog computing might be vulnerable to security attacks since it's often designed from traditional networking components.
    - It's hard to ensure authentication access to services and maintenance of privacy in a large distributed model.
    - Implementation of security mechanisms for data-centric integrity can affect the QoS of Fog Computing.

2.4 Need for Fog Computing

The increased number of interconnected devices that is estimated to reach 50 billion units by 2020 [41], resulting in an increasing burden on the Cloud as data is getting stored on the cloud; as well as the urgent need to support the computational demand for real-time latency sensitive applications of largely geo-distributed IoT devices/sensors, have given birth to "Fog Computing". Also, according to International Data Corporation (IDC), there will be 222.3 million shipments of wearable devices by 2021 [42] and the worldwide IoT market will grow up to $1.2 trillion in 2022 [43]. We actually need fog computing because of the characteristics mentioned in section 2.3.3.

2.4.1 Drawbacks of Traditional Cloud Architecture

Cloud computing has already evolved as the key computing infrastructure for Internet with complete services and it has significantly changed the landscape of IT industry by providing some benefits to end-user and corporates: on-demand self-services, frees the user from the burden of maintaining large data, scalability, storing large amount of data.

However, Cloud computing experiences many unresolved issues. One such issue is end-to-end delay for real time applications. There are many scenarios where milliseconds can have serious significance, such as emergency and healthcare-related services and vehicle-to-vehicle communications. The other major issues confronted with cloud paradigm are traffic congestion, processing of massive amount of data, and communication cost. Some of these subjects are caused generally due to large physical distance between cloud service provider's Data Centers (DCs) [10] and End-User (EU).

In addition, most of the cloud Data Centers (e.g., AWS, Google Cloud) are geographically far apart from each other (see Figure 2-6), which provokes communication delays between end-user and data centers and QoS degradation, which are not well suited for time sensitive applications.

Figure 2-6: Most of the Cloud DCs (e.g. AWS and Google Cloud) are geographically far apart from each other [23].

2.4.2 Fog Computing Vs Cloud Computing

Fog computing builds upon the capabilities of cloud computing by extending them towards the edge of the network. Fog computing does not replace cloud computing, but supplements the cloud for the most time-critical, delay-sensitive applications. On the other hand latency-tolerant applications can still be performed in the core of the cloud.

Although the Fog and the cloud share many of the same services, mechanism and attributes such as SaaS, PaaS, IaaS, Virtualization, multi-tenancy and use similar resources i.e. Storage, computing and network connectivity, the main key differences are summarized in Table 2-3.

Table 2-3: Comparison between cloud and fog computing (Adapted from [23], [44])

Features	Cloud Computing	Fog Computing
Computing Model	Centralized	Distributed fog nodes are controlled in both distributed and centralized manner
Deployment Cost	High due to sophisticated planning	Low, fog enables ad-hoc deployment with or without planning
Resource Optimization	Global	Local
Size	Cloud data centers are very large in size	Smaller, however, a large number of small fog nodes form a large fog system.
Mobility Management	Easy	Hard
Latency	High	Very Low
Operation	Operated by large companies	Often operated by small companies, however large companies can operate depending on the size
Reliability	High	Low
Maintenance	Operated and maintained by technical expert	Generally requires no or little human involvement
Applications	Cyber-domain application	Support both cyber-domain and cyber-physical applications, most importantly time-critical applications
Latency	High	Low
Hardware	Scalable storage and computing power	Limited storage and computing power
Location of servers nodes	Within the Internet	At the edge of the local network
Distance between client and server	Multiple hops	One hop
Working Environment	Warehouse-size building with air condition systems	Outdoor (e.g. Streets, gardens) or indoor (e.g. Restaurants)
Security Measures	Defined	Hard to define
Attack on Data	Less probability	High probability
Location Awareness	No	Yes

Moreover, Fog Computing brings many benefits for IoT devices and can be summarized as follows:

- Fog computing applications can be quickly developed and deployed. These applications can program the machine to operate according to the customer needs. Besides the achievement of a greater business agility, lower operating expense is

also attained. Fog computing saves the network bandwidth by processing selected data locally instead of sending it to the cloud for analysis [21].

- According to [45], other benefits can be accomplished by extending the cloud closer the things such as:
    - Low latency: The fog has the ability to support real-time services (e.g., gaming, video streaming).
    - Geographical and large-scale distribution: Fog computing provides distributed computing and storage resources to large and widely distributed applications.
    - Scalability: The closeness of fog computing to end devices enables scaling the number of connected devices and services.
    - Flexibility and heterogeneity: Fog computing allows the collaboration of different physical environments and infrastructures among multiple services.

2.5 Fog Applications

According to Cisco [21] Fog Computing is greatly considered when:

- Data is collected at the extreme edge: vehicles, ships, factory floors, roadways, railways, etc.
- Thousands or millions of things across a large geographic area are generating data.
- It is necessary to analyze and act on the data in less than a second.

Some of the applications where Fog computing can play an important role are listed below:

- **Smart Traffic Lights:** Fog computing enables traffic signals to open roads depending on sensing flashing lights of the ambulance. It detects the presence of pedestrians and cyclists and measures the distance and speed of the nearby vehicles. Sensor lighting turns on when it identifies movements and vice-versa. Smart traffic lights may be considered to be fog nodes which are synchronized with each other to send warning messages to nearby vehicles. The interactions of the fog between the vehicle and access points are improved with WiFi, 3G, smart traffic lights and roadside units [46].
- **Connected Vehicles:** There are many beneficial features that can be added to cars such as automatic steering and "hands-free" operation or self-parking features, which means that there is no need for a person behind the wheel to park the vehicle. Fog computing will be the most efficient solution for all Internet-connected vehicles, since it provides a high level of real-time interaction [46]. In [47] the effectiveness of mobile fog with traffic monitoring and vehicle tracking applications are well described.
- **Healthcare and Activity Tracking:** One of the primary objectives of Fog computing is to provide real-time processing. Fog computing can play a vital role in healthcare, in which real-time processing and event response are critical. In addition, the interaction of a large number of healthcare devices for remote storage, processing and medical record retrieval from the cloud requires a reliable network connection which is not always available. Fog computing, however, addresses issues regarding network connectivity and traffic [48].

- **Big Data Analysis in Smart Cities:** Fog computing satisfies the needs of future smart cities where abundant deployment of various kinds of sensors requires a new computing paradigm. A hierarchical fog computing architecture for big data analysis in smart cities is described in [49].
- **Augmented Reality:** Augmented Reality (AR) refers to systems that add virtual information to real world whereas Virtual Reality (VR) refers to systems that simulate the real world [50]. Augmented reality applications are highly latency intolerant since a small delays in the application response can damage the user experience [51]. An augmented brain-computer interaction game (ABCI) based on fog computing has been designed by [52].

Chapter 3: Background and Motivation

3.1 Fog Concept in Literature

In the last few decades, the computational paradigms switched back and forth between a centralized and decentralized computing approach. The first known distributed system was established in 1970s and is known under the name of local area network (LAN), which interconnects multiple computers in order to make applications communicate with each other for developing a collective solution [53].

In the mid of the 2000s, cloud computing has been introduced as a centralized approach. Although cloud paradigm is prospering and is not going to be replaced in the nearest future, there is a force pushing toward a novel decentralized approach to solve the essential problems of centralized systems such as: high latency and missing of location-awareness. Therefore, Fog computing was introduced as a new decentralized computing for the IoT in 2012 [18]. Authors drew the vision and defined the key characteristics of fog computing. They claimed that fog is the appropriate platform for a number of critical IoT services and applications like Smart Grid, Smart Cities, and Connected Vehicles and in general, Wireless sensors and actuator networks [18]. According to a recent survey [40], the total number of the published and reviewed papers in the field of Fog Computing was sixty-eight for the 2013-2017 period. A comprehensive definition and a broad overview of the fog with some highlights of the main challenges faced the emerging of fog computing offered by [16]. The works of [54] have focused on some of the application areas of fog

computing in comparison with cloud computing. Authors in [55] discussed the definition of fog computing, introduced some application scenarios and highlighted the opportunities and challenges that might come up while designing fog computing system. The Impact of Fog computing over 5G networks has been discussed in [56]. The security and privacy challenges of fog computing have been discussed in a recent paper [57].

3.2 Fog Computing for Delay-sensitive Applications and For Healthcare systems

Authors in [47] provided a simplified programming abstraction for latency-sensitive applications. They also analyzed two use cases namely the vehicle tracking using cameras and the traffic monitoring using mobility-driven distributed complex event processing (MCEP) systems.

In [58], authors proposed a service oriented fog computing architecture that allows patient health monitoring in non-clinical settings. The main point of the proposed architecture is a low power embedded fog node that carries out data mining and data analytics on raw data collected from various wearable sensors used for telehealth application. The architecture employs a single fog node at the fog layer for onsite processing to reduce the amount of data to be stored and transmitted to the cloud. A working prototype of the proposed architecture was built and used to carry out two important healthcare problems namely speech disorders and ECG. The obtained results showed substantial improvement in system efficiency using the proposed architecture.

Authors in [59] employ a fall detection application to demonstrate the efficacy of fog computing for health monitoring. In their work, the authors investigated and developed new fall detection algorithms and designed and employed a real-time fall detection system

called U-Fall. However, their work does not substantiate the efficacy of fog over the cloud for delay sensitive applications because they did not compare the response time of the fog with the response time of the cloud.

In [60], authors proposed a fog computing model to easily and quickly notify the caregivers in case of emergency. They only compared the upload delay and synchronization delay of fog and cloud. In [61], authors proposed an enhanced cloud-based Fog Computing system in which bio-signals are analyzed at the fog server side for real-time applications.

Authors in [62] focused mainly on enhancing and flexibly controlling the data privacy issues in healthcare systems. Authors in [63] defined an intermediary layer of intelligence between sensor nodes and cloud. They also implemented an IoT-based Early Warning Score (EWS) health monitoring to practically show the efficiency of the system. In [64], authors proposed a solution for E-health systems to communicate by combining fog computing and Content-Centric Network (CCN) to provide communication efficiency and local storage opportunities. They used the simulation tool ndnSIM – a tool which offers to simulate NDN networks - to evaluate and compare the delay and hit rate packet in the case of CCN network and fog network providing more storage capacities. They have shown that the performances of CCN networks and fog networks are good and adapted to the emergency context of E-health.

Authors in [65] focused on a smart e-health gateway implementation for use in the fog computing layer. They defined the Smart e-health gateways as a distributed network of gateways forming the Fog layer and serving the underlying sensor and actuator network. It hosts a wide variety of services and acts as a bridge to the cloud.

In Literature, the smart gateway can be sometimes considered as an intermediate computing layer between the sensor nodes and the cloud, also known as Fog computing layer [66]. Features of the gateway in fog implementation are discussed and evaluated in the following paragraphs.

3.3 Fog Computing with Smart Gateway (or Broker) for E-Health Systems (previous work in the subject)

To the best of our knowledge, the number of papers that used the concept of fog computing at the smart gateway is very low. In 2014, authors in [24] presented the architecture of a smart gateway with fog computing. In their proposed system, the smart gateway (e. g. a router that is used for data transportation is a gateway node), which is an intermediate layer between IoT and fog node, performs a number of tasks starting from data collection to preprocessing, filtering, reconstruction, uploading only the necessary data to the cloud, keeping check on IoT nodes, and many others. This system, however, lacks the essence of the fog concept as it puts more intelligence on the gateway rather than on the fog node (see Figure 3-1).

On the other hand, authors in [67] analyzed both cloud-only and cloud-fog scenarios in the context of processing delay and power consumption according to increasing number of users. It is worth mentioning that the authors used in their simulation architecture a broker (with simple tasks to perform) instead of a smart gateway with high computation capabilities (see Figure 3-2). The broker used in [67] (which has only 1 instance) is responsible for gathering the messages from all users and clouds/fogs and broadcasting it to all models just like a switch. Furthermore, this model has a map-key data structure which allows it to keep track of the service each fog/cloud offers. Once a message is received

from the user, the broker will decide on how the message should be processed. Once the decision is made, the broker sets a new destination for the request and broadcasts the message over the network. If a message is received from the fog/cloud, the broker will simply change the destination of the user to whom the message is intended and broadcast the message over the network. This is known as all incoming messages from the fog/cloud contain the user ID. The fact that IDs are unique ensures the correct delivery of every message.

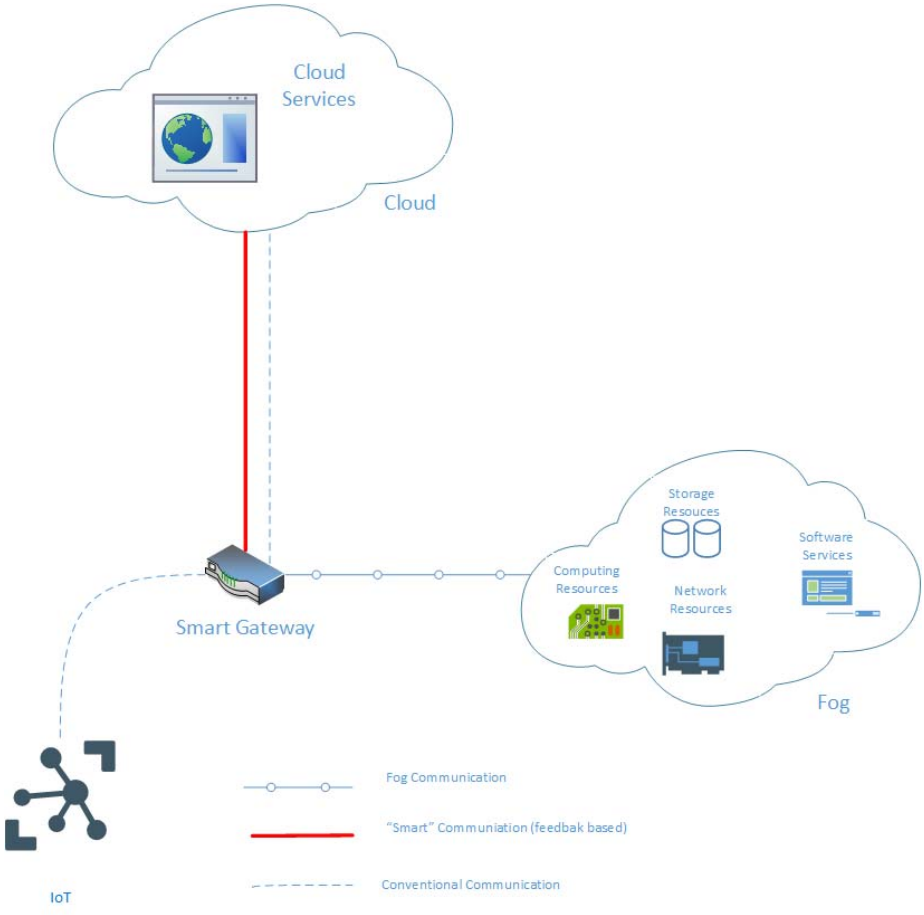


Figure 3-1: Smart Gateway with Fog Computing (Adapted from [24])

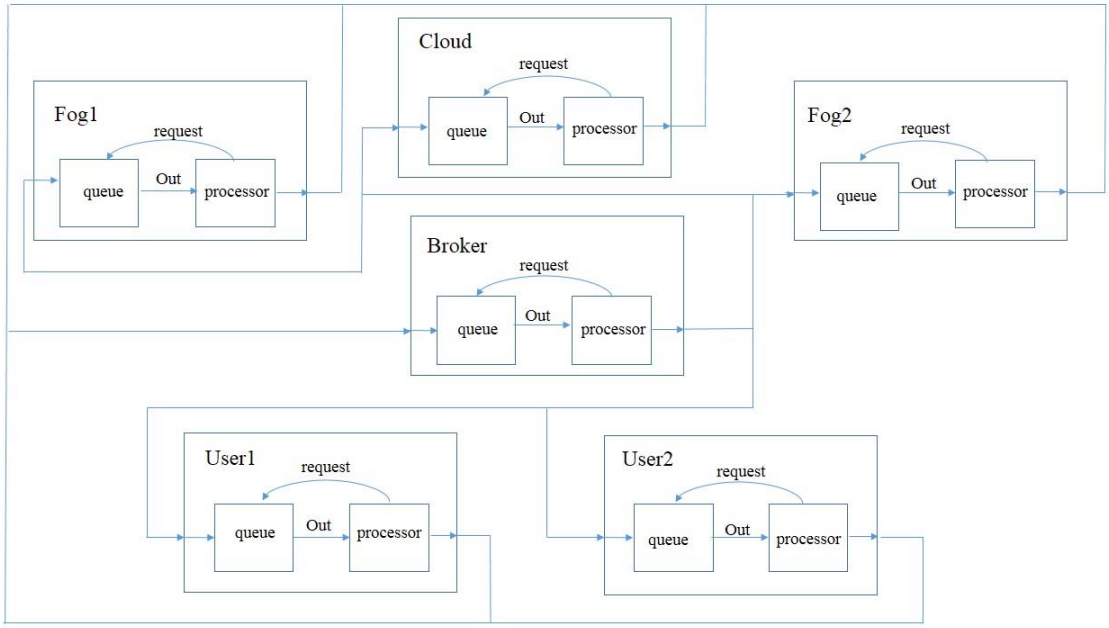


Figure 3-2: Fog with a Broker (Adapted from [67])

3.4 Research Motivation (our idea)

All studies and papers in the literature show the significance of using the fog computing in order to reduce the latency in IoT-fog-cloud paradigm. For instance, researchers in [17] built a proof-of-concept platform to run a face recognition application, and the response time was reduced from 899.970ms to 168.769ms by moving computation from cloud to the edge. Authors in [68] implemented fog computing model for executing the iPokeMon game which is a location-aware online game similar to PokeMon Go. It was noted that, on average, the response time can be reduced by 20% in the edge computing model for the user playing the game and the data transferred between the edge node and the cloud was significantly reduced by 90%. However, none of the literature discussed have explored a robust and significant use of a broker with the fog computing in real-world time sensitive applications. Considering the aforementioned, and after having discussed the background

and the architecture of fog computing, we shall state next the main questions to be addressed in this thesis:

- How much better is it to have a broker node in a healthcare scenario instead of a standalone fog working without a broker?
- What benefits can be gained by using a fast device (broker) instead of a smart device with huge computing capabilities?
- When is it necessary to deploy a broker (smart device) in IoT-Broker-Fog computing?
- What is the optimum value of batch period (T) in IoT-Broker-Fog scenarios?

In order to answer these questions, simulation models are built using MATLAB R2017a.

Firstly, an architecture of IoT-Broker-Fog model has been presented. In this proposed design, a fast broker that handles very basic computation with a specific task for filtering the arrival packets taking into the consideration the user and the status of monitored health (normal or abnormal) and the consecutiveness of abnormality has been implemented as a "layer" between IoT device and Fog network in such a way to reduce the traffic over the network and the delay for most latency-sensitive applications.

Secondly, a set of experiments have been performed to analyze the variation of delay with regard to the existence or absence of a broker between IoT and Fog Server.

Finally, simulations are used to find optimum value of T which is the time of a batch generation – set of packets – to be sent over the network to the fog node.

Chapter 4: Original Work

4.1 Explanation of the Proposed Architecture

This thesis sheds light on a specific area in the IoT-Fog-Cloud architecture where a broker node is to be deployed and configured between the IoT devices and Fog network with an objective to reduce the end-to-end delay and the traffic over the network.

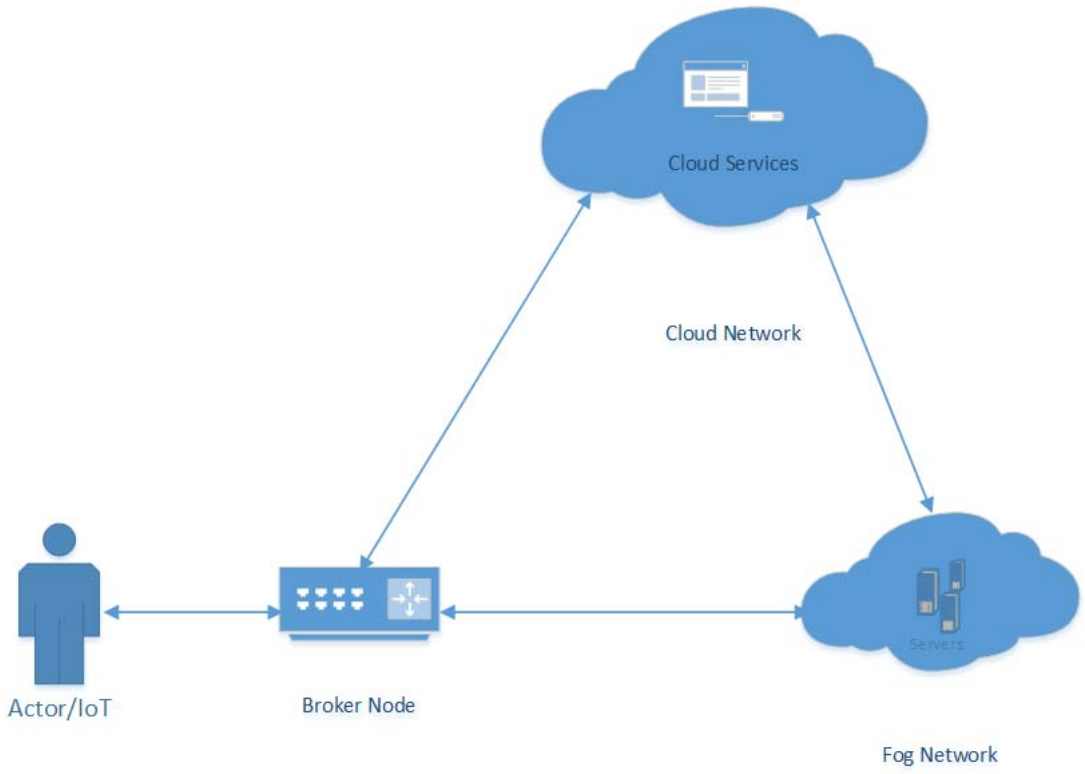


Figure 4-1: IoT-Broker-Fog-Cloud-Model

- Layer 1: This is where the data is generated. This layer consists of sensor nodes/IoT devices. These sensors send their data to layer 2.
- Layer 2: This layer consists of a broker (or a set of brokers). The proposed broker does limited amount of works only to reduce the usual everyday traffic and to speed up the process. It has a small storage capacity and can be any device with very basic computation capabilities, e.g. a raspberry Pi which is considered also a very cheap device. The broker can be virtualized; for example, it can be an application running at an ISP local router. It could be located to any near location to the source of the generation of data. The broker could be added on each phone central which will be closer in general to all the sensors, in contrast to mobile towers where the fog might be located. In addition, it can be located on premises for a specific application such as in a hospital. Also, it can save some data for a specific period of time (e.g. User ID) and some predefined data might be saved (e.g. Heartbeat normal range).
- Layer 3: It consists of a fog node or sequence of fog nodes in the proposed architecture. These nodes have a good storage capacity and computation capabilities. When user wants to access recent data, the data communication would be only between the layer 1 or patient layer up to layer 3 (in most cases), which reduces the communication cost of going to the cloud.
- Layer 4: The layer 4 is the cloud layer where all the data coming from the network are stored permanently. The main advantage of having the cloud is that data can be accessed by users whenever needed, thus helping the analysis and interpretation of the pragmatic data.

4.2 Simulation Details of the Architecture

Figure 4-1 depicts the architecture of an IoT- broker-fog- cloud paradigm. The proposed model is very useful for a large organization such as a hospital and in smart cities where several patients' steps are monitored every second. Data can be quickly accessed and rapid human actions can be taken in case of an emergency. For this thesis simulation purpose, a data generator device, a broker node, as well as a fog node are needed. Hence, if this structure works as per the thesis hypothesis, which is reducing the delay for time-sensitive applications, then it could be applied and extended in a mass scale for a large organization/smart cities environment. Next are the three primary nodes as mentioned before:

- **Data Generator Node -** This is the layer 1 node (represented by an Actor). This node constitutes the source of generation of data. We emulated real-world health sensor data generation scenario at the data generator node using a function that generates data at different times, using Poisson Algorithm.
- **Broker Node -** In the proposed simulation, the broker differentiates the type of request and takes the decision whether to forward the request to Fog network or to simply drop the arrived data if the data is normal. More details about how the broker works will be explored in Section 4.3.
- **Fog Node -** The fog layer is responsible for processing the data for immediate response purpose. Different kinds of processing and different applications can be running simultaneously at the fog nodes. We have created a functions called "server" which works as a fog node or server device. This "server" function is about

a fog node with some processing delay. The data is then sent to the cloud in a scheduled manner for later use or analysis (Out of scope of this Thesis).

4.3 Simulation of the Data Transmission

In this section, the data flows that have been designed to validate the effectiveness of the proposed model are presented.

A comparative analysis in terms of batch delay has been conducted. Here, the batch is about a set of 1 or more packets transferred from the Network Gateway to the fog network. The end-to-end and mean delay of batch creation and arrival have been calculated and evaluated in two different data transfer scenarios. In the first Scenario, IoT device communicates with a Fog server through a simple network device, whereas in the second scenario, a broker with smart gateway functionality has been integrated with a goal to reduce the traffic sent over the network and with a hypothesis that the end-to-end delay will be reduced.

- In Scenario 1: Data is generated at the Patient, and then transmitted to the fog through a simple network device. The network device creates batches of one or more packets based on a given T.

  In our simulation, we are trying to find the optimum value of T where the batches may reach the fog node and be processed with minimum delay. Figure 4-2 shows the architecture and data flow path for scenario 1.

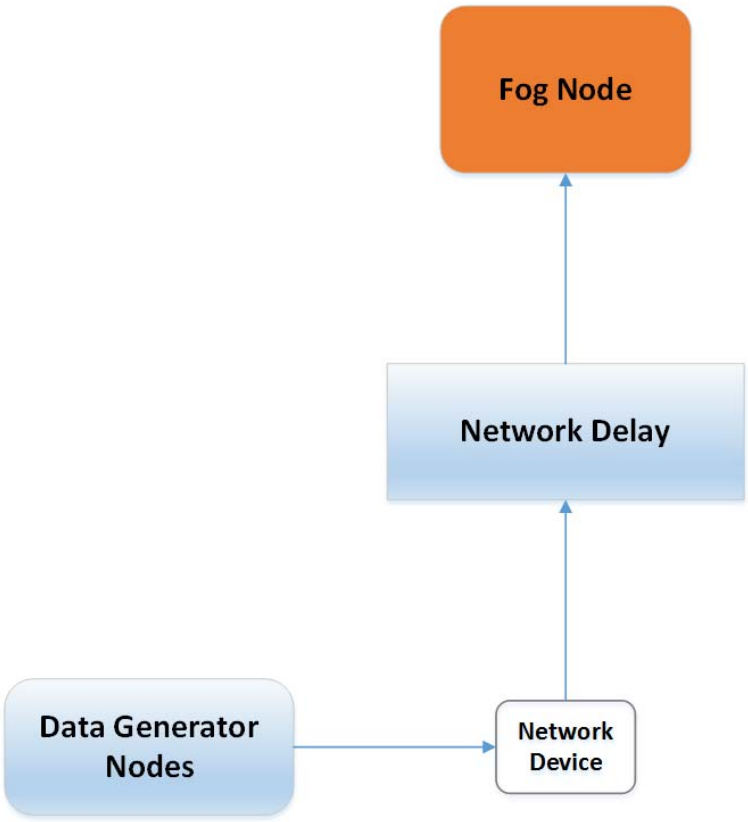


Figure 4-2: Scenario 1 – Data Transmission from Data Generator Node to Fog Node

- In Scenario 2: Data is generated at the Client, and then transmitted to the fog through a smart network device, called broker which has limited computing and storage capabilities.

  The broker generates batches of packets based on T (will find the optimum value of T where the batches may reach the fog node and be processed with minimum delay) and identifies the sender of each packet and its health status whether it's normal or abnormal. The broker sends to the fog node only the packets with abnormal health status in different batches based on T for further analysis and more action to be taken based on the health abnormality. Figure 4-3 shows the architecture and data flow path for scenario 2.

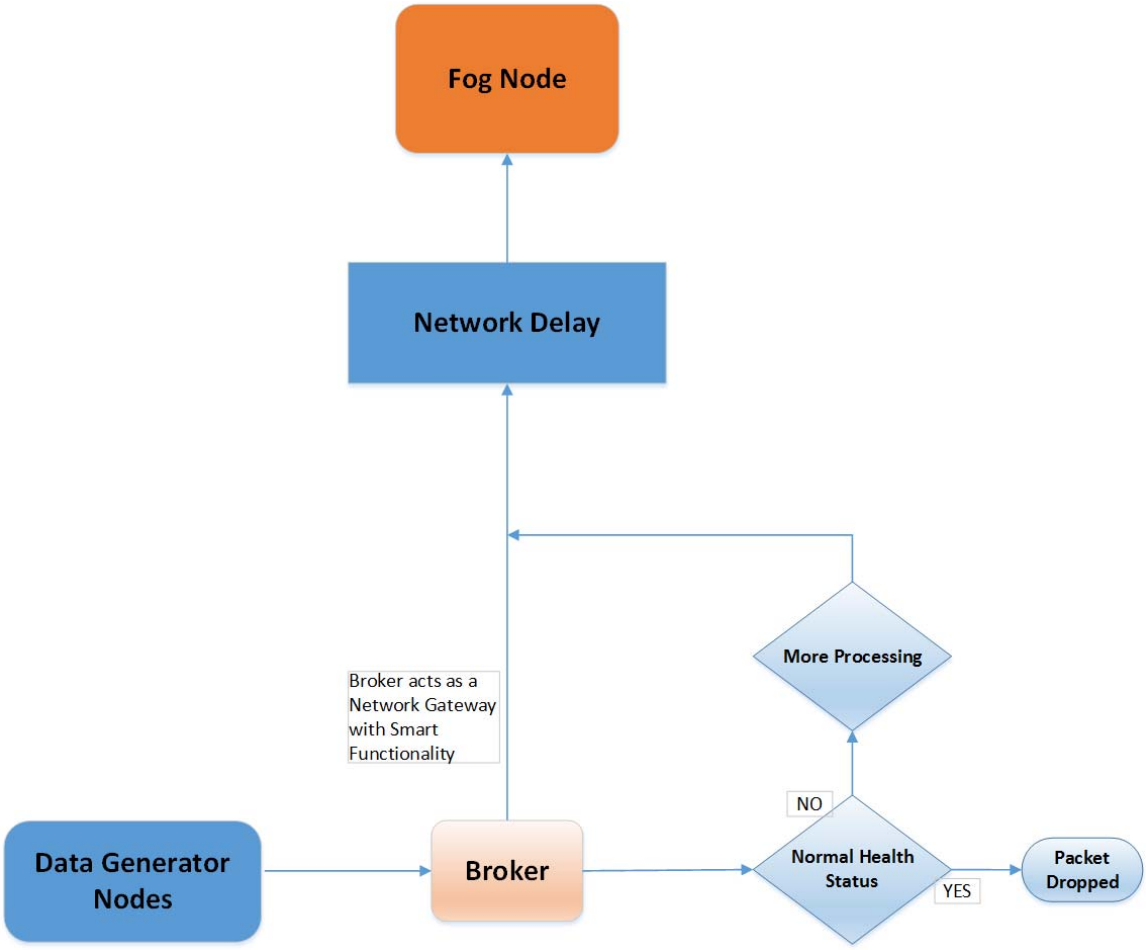


Figure 4-3: Scenario 2 - Data Transmission from Data Generator Node to Fog Node in the Presence of a Broker

4.4 Simulation Methodology

In this thesis, data has been generated at different times as if generated from different sensors or IoT devices connected on patients. However, in order to show the generic nature of the proposed architecture, many experiments have been conducted with different batch periods (T) such as T=2, T=5, T=10. As we aim to compare the performance of a broker with Fog in terms of delay, working with data of different sizes (different number of

packets originally generated) with different batch sizes to travel over the network to the fog network gives us an insight into the efficacy of our fog model.

4.4.1 Data Generation

At the data generator node, we randomly generate dummy data using Poisson distribution based on packet arrival rate K. K is a Poisson variable, and the distribution of K is a Poisson distribution. Different experiments have been conducted with different K values.

Generated Data consists of Patient's ID (we assumed that we have ten patients connected with IoT devices), Patient's Health Status (0: Normal - 1: Abnormal), and the packet departure time based on Poisson distribution.

Listing 1: Poisson Function

```
function [ P] = Poisson(k,pktNum)
    % rand('uniform');
    %arrival rate k=0.1 to k= 0.9;
    P(1)=0;
        for i=2:pktNum
        r=rand;
        pp=r*2/k;
        P(i)=P(i-1)+pp;
    End
```

4.4.2 Batch Creation and Data Transmission

Data is transmitted from the client to the Fog either via a simple network gateway or via a smart gateway called broker. In order to successfully accomplish this transmission and evaluate the end-to-end delay, the network gateway generates a batch of packets every T (time to send the batch to the Fog). In the first scenario, all the packets arrived at the network device are being placed in batches and then sent to the fog over the network. In the second scenario where the broker exists, the packets received with normal status are dropped and the packets with health status equal to abnormal are being collected into batches and then sent to the fog.

4.4.3 Emulation of Delay

In our model, we need to emulate service and distance delay in three locations. We might have delay at the broker where the packets are filtered and collected into different batches. There will be also delays on the network and on the fog node where the data is being analyzed and different actions are to be taken. Therefore, to emulate the processing delay at the broker, a function called SDServ is built.

Listing 2: SDServ Broker Processing Delay

```
function SDServ = serverr(P,s, Z)
    for i=1:Z
        ser(i)=-1/s*log(1-rand);
    end

    Serv(1)=ser(1);

    for k=2:Z
        a=Serv(k-1)-(P(k)-P(k-1));
        if a < 0
            Serv(k)=ser(k);
        else
            Serv(k)=ser(k)+a;
        end
    end
```

Another function called ServerDelay is built to emulate the processing delay at Fog node and a NetworkBatchDelay function is also built to emulate the network delay. All the functions are written using MATLAB scripts based on the exponential service time distribution.

4.5 Results

Different simulation tests have been conducted for different T values, where T represents the time to process a batch (one or more packets) in a trial to find the best T value for the packets to be processed at the fog nodes with different K Values (0.1 - 0.9) where the K

represents the arrival rate of packets in our simulation and at a different network rate (Mu = 1 or 2). The larger K values, the more traffic we will have in the network. Moreover, we assumed we have 10 users connected to IoT devices (e.g. Wearable Health Device) from which the data is generated. In our simulation, we considered that each packet delivered contains the patient identity number and its health status (0: Normal Status – 1: Abnormal status). Hence, we tested our simulation model also with different numbers of packets generated from these devices by using Poisson distribution technique.

As it is mentioned in the experimentation section 4.3, the end-to-end and mean delay of batch processing have been calculated and evaluated in two different data transfer scenarios.

The results of one sample trial of the simulation where T value was set to 2, as T represents the time to process a batch, and K value was set to 0.9, where K is the arrival rate of packets of 10 users generating 100 packets, are shown in Figure 4-4. In the following graph, the x-axis represents the batch index while y-axis represents the end-delay.

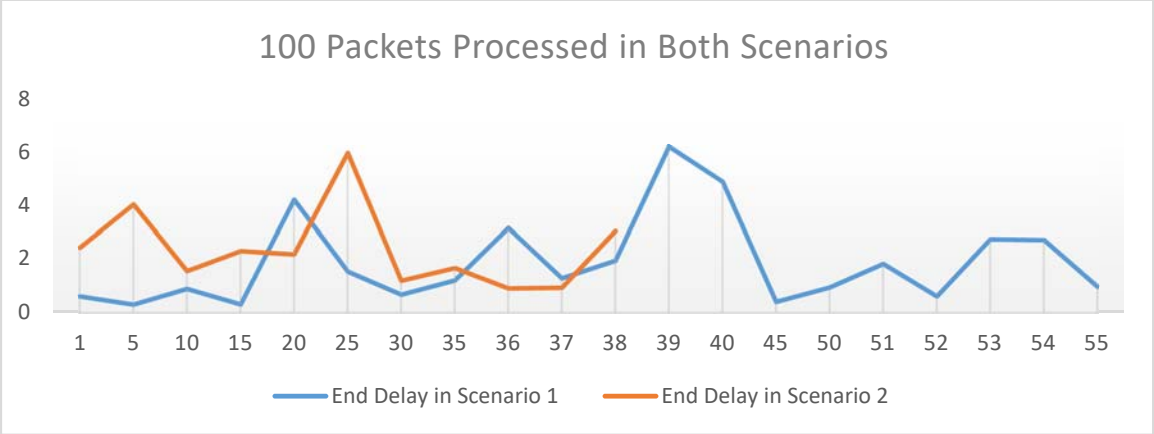


Figure 4-4: Sample Simulation Result with T=2

According to Figure 4-4, the number of generated batches in Scenario 2 has been decreased from 55 batches to 38 batches, a drop by 30% on average. The end-to-end delays for the first 38 batches are higher in scenario 2 in 76% of the time.

Figure 4-5 depicts the results of two different runs for the simulation for T =5 and T=10 on both scenario 1 and scenario 2. The results show that for T = 5, the end-to-end delays for 15 batches out of 25 are lower in scenario 2 whereas for T = 10, 8 batches out of 13 have lower delay (%60 of the Time delays in scenario 2 are better). In addition, the smallest end delay resides in scenario 2.

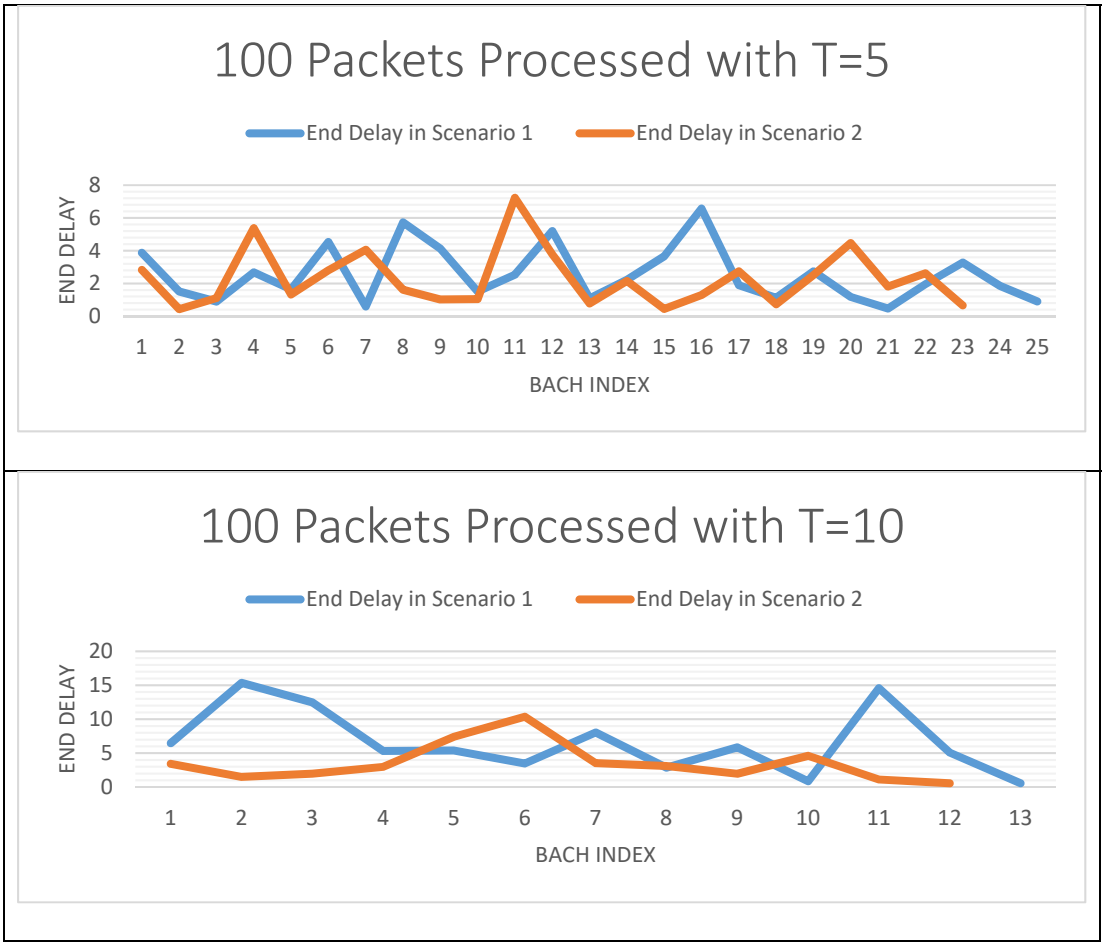


Figure 4-5: Sample Simulation Results for T=5 and T=10 in Scenario 1 and Scenario 2

In all the experiments, we considered, firstly, that 80% of the sent packets have abnormal health status. Secondly, we also considered that the batch processing time on both the Broker and on the Fog Node is equal, whereas in the real-world the processing time in the broker might be less than the processing time of the fog node since the broker has only specific tasks to do and doesn't perform complex computations. The broker will have to identify the health status of each packet arrived. If the packet is normal, then the broker ignores it, but if the health status is abnormal, the broker checks the last health status of the underlined patients. In case if it was abnormal, the packet is put in the processed batch, to be delivered to the Fog Node within the given T value for further analysis and actions to be taken. Therefore, two consecutive abnormal health statuses for the same patient mean processing the last packet with the processed batch of a given T value. The date flow of the broker in our simulation is depicted in Figure 4-7.

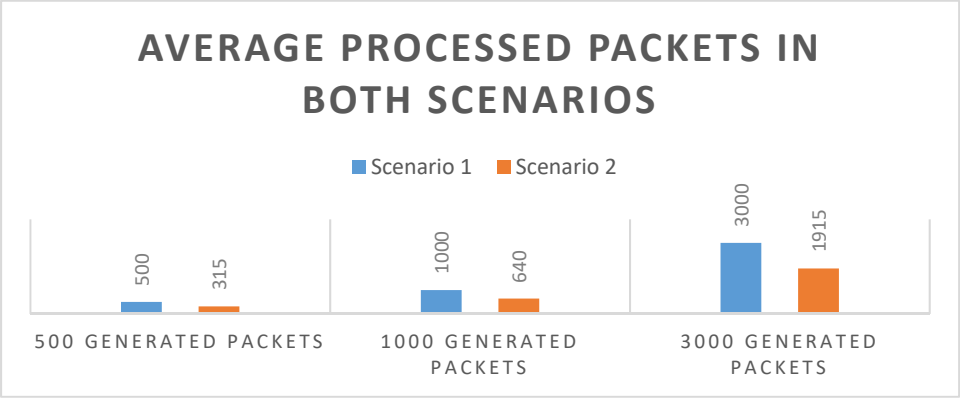


Figure 4-6: Average Processed Packets in Scenario 2

Figure 4-6 shows the average number of processed packets in both scenario 1 and scenario 2. The number of the packets has been decreased by an average of 35% in scenario 2.

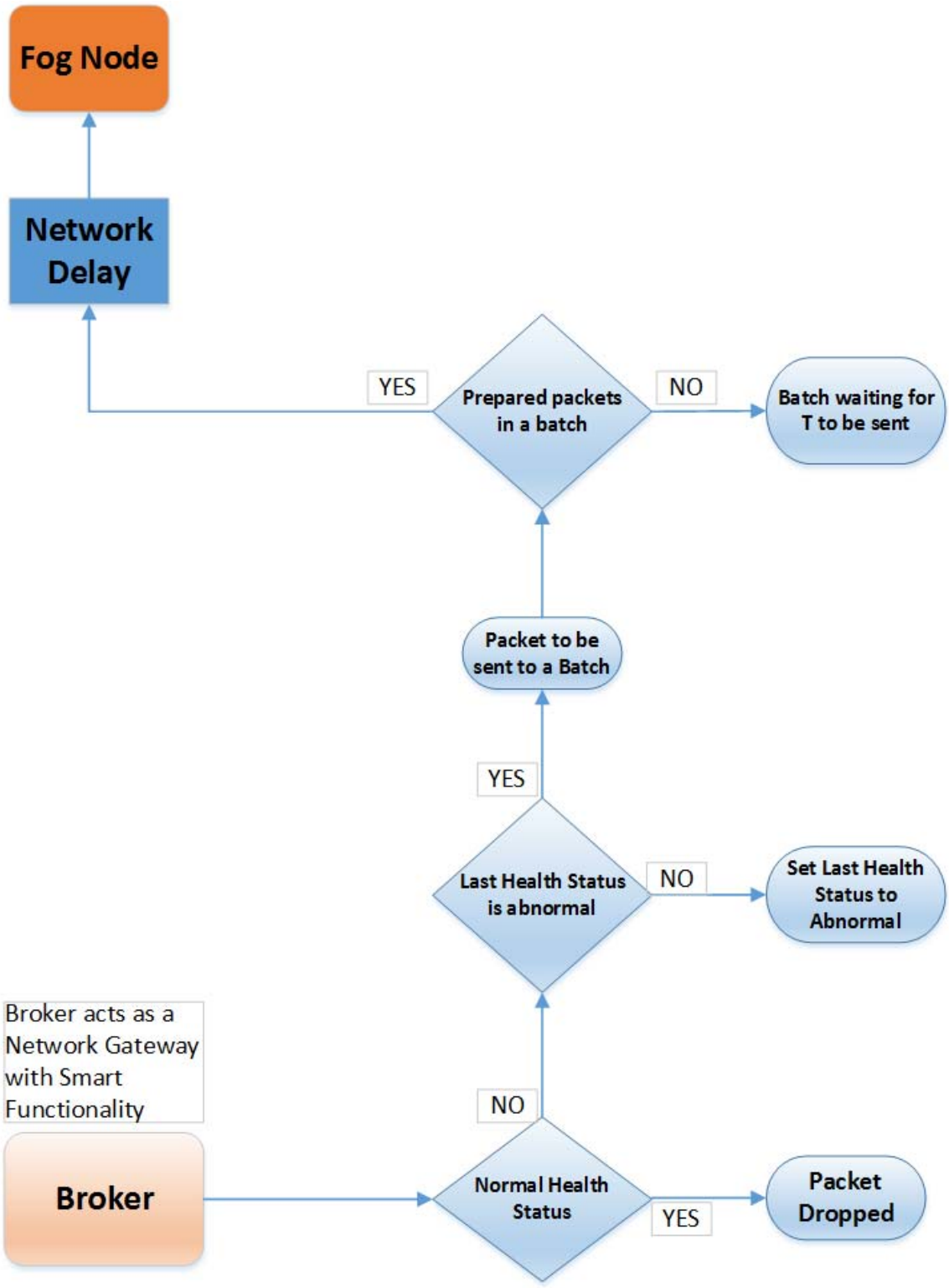


Figure 4-7: Data Flow in Broker

Some other tests were conducted with T=1 through T=10 with 500 and 1000 generated packets for network service rate "Mu" set to 2 (Mu represents the network rate. The higher is Mu the faster is the network) and K set to 0.9. The results are shown in Figure 4-8. In the following graphs, the x-axis represents the T value input and y-axis represents the end-delay.

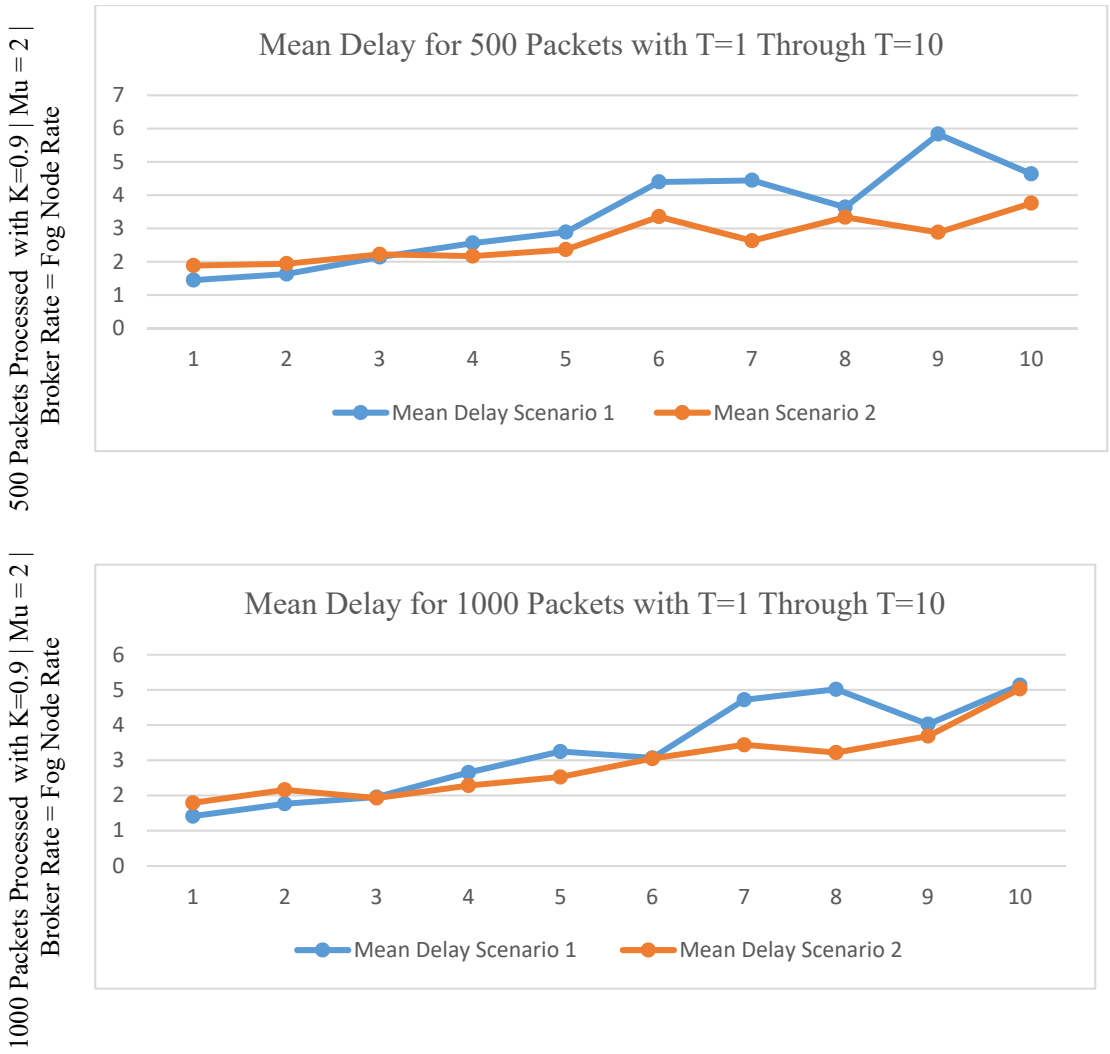
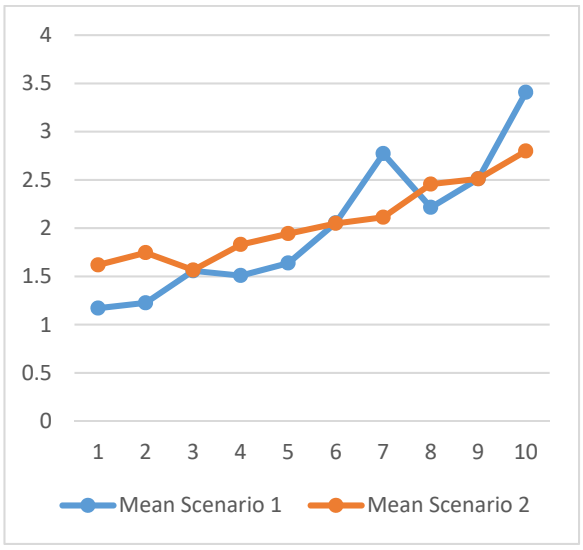
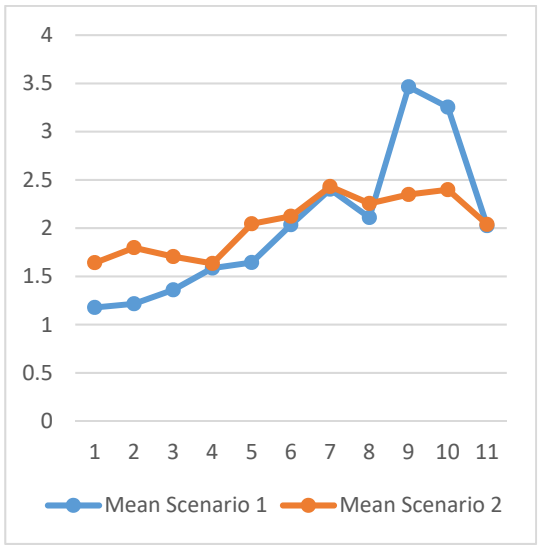


Figure 4-8: Simulation Results for 500 and 1000 Generated Packets

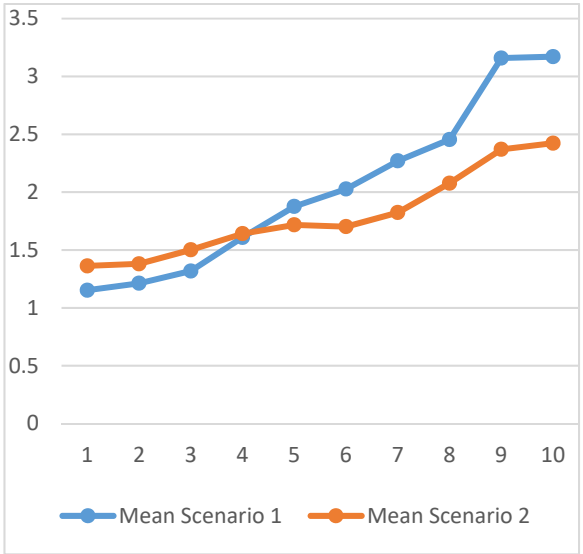
Figure 4-8 shows us that the smallest mean delay always resides in scenario 1 with T value less than 3. However, the mean delays with T value greater than 3 are smaller in scenario 2, which means that in 70% of the Time, the delays are better in Scenario 2.

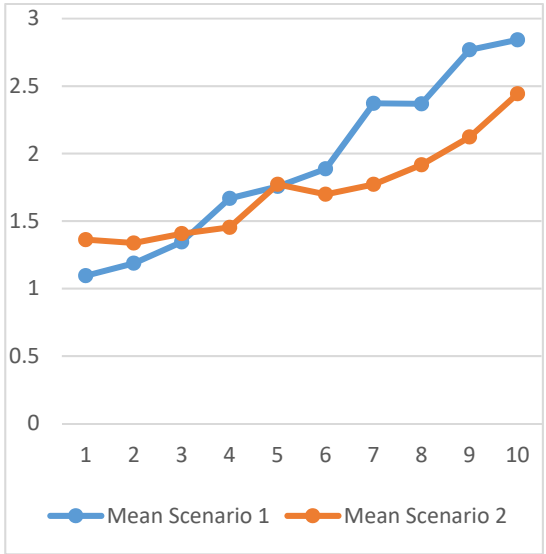
Figure 4-9 depicts the results of two different tests conducted for 500 and 1000 packets but now with K=0.5, which means less traffic in the core network. In the following graphs, the x-axis represents the T values input and y-axis represents the mean of end-delay.





500 Packets Processed in Two Tests with K=0.5 | Mu = 2 | Broker Rate = Fog Node Rate



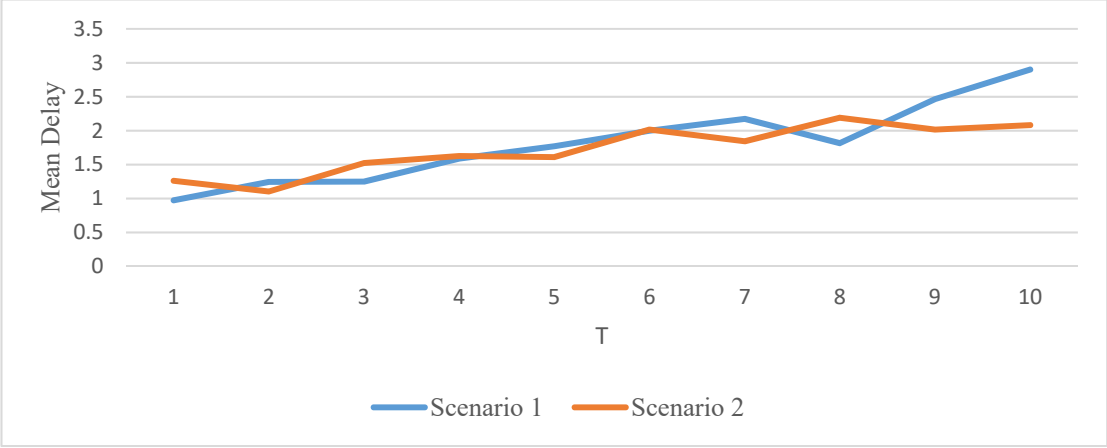


1000 Packets Processed in Two Tests with K=0.5 | Mu = 2 | Broker Rate = Fog Node Rate

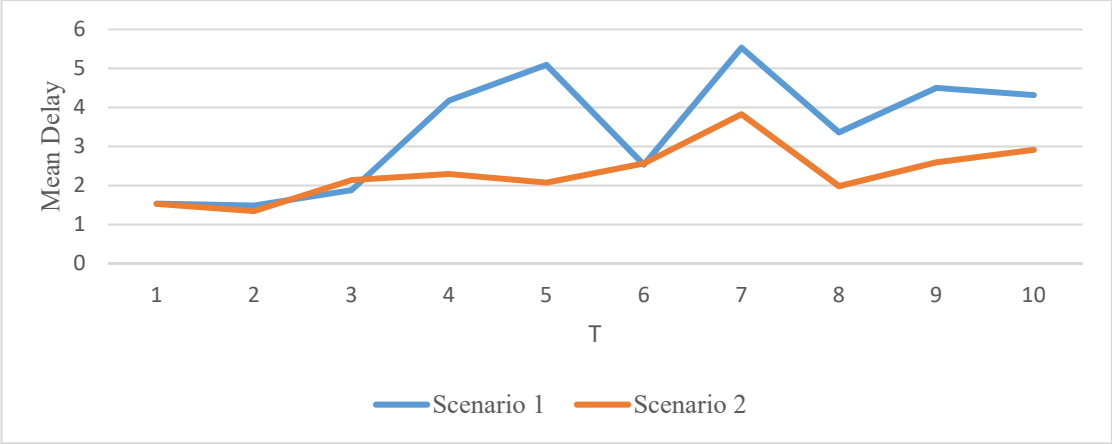
Figure 4-9: Mean Delay for Two Different Tests with K= 0.5 for 500 and 1000 Packets

Graphs in Figure 4-9 show that the mean delays are higher in scenario 2 when the packet arrival rate is slow and the number of generated packets is small (> 70% of the Time, the delays in Scenario 1 are better). As the number of packets increases in the network the means of end delays become lower in scenario 2 (> 60% of the Time, the delays in Scenario 2 are better).

Figure 4-10 shows the results of simulation tests for 100 generated packets with T value equal to 1 through 10, where T is the time to process a batch, and K value was set to 0.5 and 0.9 and Mu was set to 2, knowing that K represents the arrival rate of packets. In these simulation tests, the processing speed of the broker was set to double of the processing speed of the fog node, taking into the consideration that in the real-world scenario, the broker processing time would be faster than that of the fog node. This is simply due to the fact that the broker would not have to perform complex computation as the fog node does. In the following graphs, the x-axis represents the T values input and y-axis represents the mean of end delay.



K = 0.5 | Fog Node Rate = 2 & Broker Rate = 4 | Mu = 2 | Packets = 100



K = 0.9 | Fog Node Rate = 2 & Broker Rate = 4 | Mu = 2 | Packets = 100

Figure 4-10: Simulation Tests with Broker Rate = 4 and Fog Node Rate = 2

Figure 4-10 shows that the mean delays are almost equal for K=0.5 and are getting lower in scenario 2 when the packet arrival rate is high (K=0.9) and the broker rate was set to double the rate of fog node.

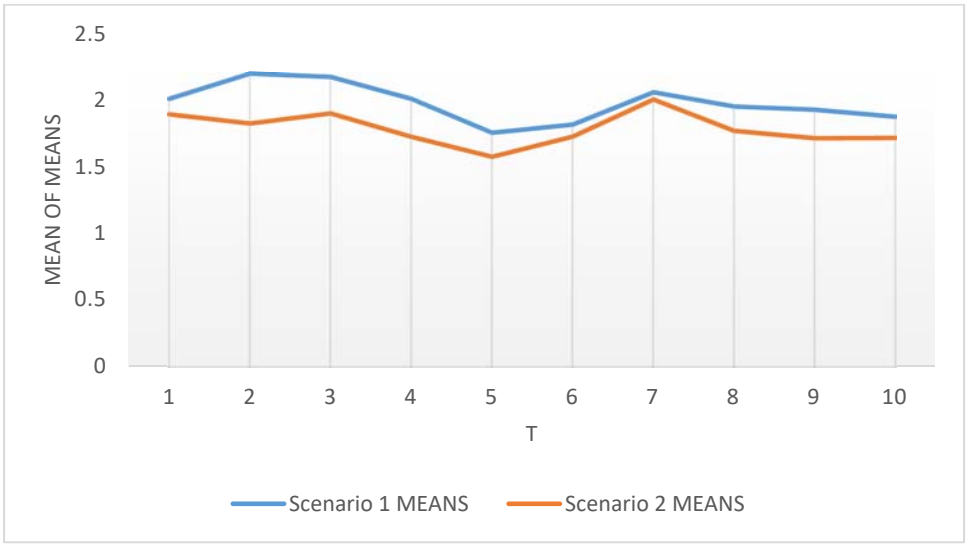
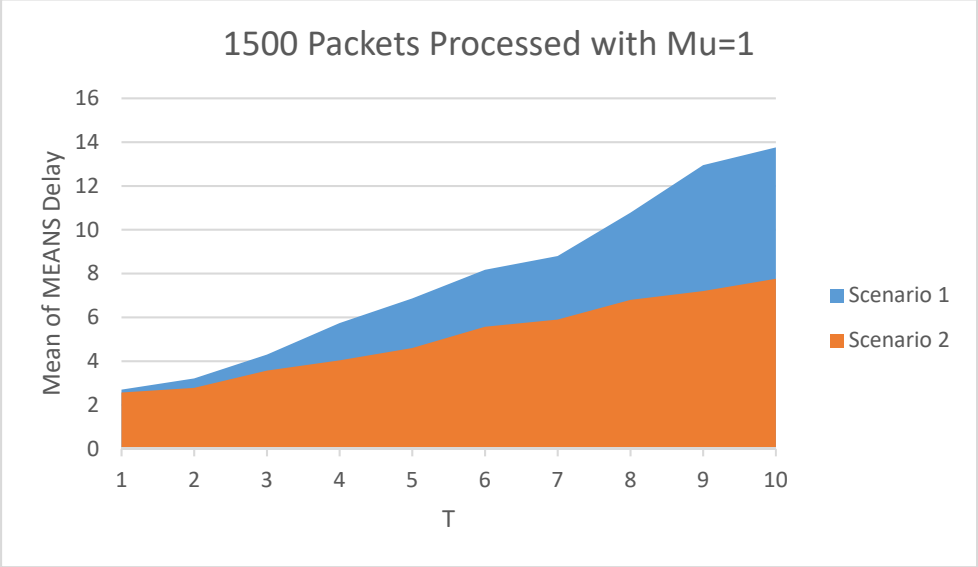


Figure 4-11: Mean of MEANS for 10 Simulation Tests

Figure 4-11 shows that the mean of MEANS in scenario 2 is always smaller in comparison with the mean of MEANS in scenario 1 when Broker rate is set to double the rate of the fog node for K = 0.5.

Some other tests were conducted also with Mu =1 on 1500 packets with packet arrival rate K = 0.9, Fog Node processing rate S = 2 and Broker processing rate = 4 for T=1 through 10. The results are shown in Figure 4-12.



K = 0.9 | Fog Node Rate = 2 & Broker Rate = 4 | Mu = 1 | Packets = 1500

Figure 4-12: Simulation Results for 1500 Packets with Mu=1 and K=0.9

Figure 4-12 shows that the smallest mean delay resides in scenario 2 and that all the mean of MEANS delays for T=1 through T=10 are always lower in scenario 2.

4.6 Analysis

The results show that using a broker in IoT-Broker-Fog model can reduce the delay of sending data by a significant amount compared to the scenario 1 where all the data generated were being transferred to the fog node for further processing. In this section, we will take a close look at the results and analyze their implications.

After conducting more than 100 runs for the simulation with different values of Network Service Rate (Mu), Fog Node Rate (S), Broker Rate (D) and Packet Arrival Rate (K), we can conclude the following:

- First, The smallest end-to-end mean delay always resides in T <= 4 for all the conducted tests, and for all the T<=4 values, the mean of MEANS is always smaller than the mean of MEANS of all the T values between 4 and 10. In addition, the number of delivered packets over the network decreased by 35% in scenario 2 where a broker is deployed (See Figure 4-6).
- Second, when Broker Processing Time is equal to Fog Node Processing Time:
    - The end-to-end mean delay is always the smallest in the scenario 2 when T value is between 4 and 10; However, the smallest mean always resides in T <=3 in scenario 1 (see Figure 4-8).
    - With K=0.5, the means of end delays are higher in scenario 2 when the number of generated packets is small. As the number of packets increases in the network, the means of end delays become lower in scenario 2 (see Figure 4-9).
    - The Mean of MEANS is always smaller in scenario 2 with K=0.9, but with K=0.5, the mean of MEANS is smaller in scenario 2 for Packets >=1000. For a small number of generated packets (<1000), the smallest mean of MEANS resides in scenario 1.
- Third, when Broker Processing Time is set to 4 and Fog Node Processing Time = 2 (See Figure 4-10):

  - The end-to-end mean delays are smaller (>70% of the time) in the scenario 2 when T is between 4 and 10 with K = 0.5 and Mu >= 2; the smallest mean, however, resides in scenario 1 for T <4.
  - With K=0.9 and Mu >=2, the end-to-end mean delays are smaller in the scenario 2 for almost all T (>70% of the time); and the smallest mean resides in scenario 2 (80% of the time) with T <4.
  - Regarding the Means of MEANS, they are always lower in scenario 2 with K=0.5 and 0.9 (See Figure 4-11)
- Forth, with Mu = 1, K= 0.9 and Broker Rate = 4 (See Figure 4-12), the smallest delay always resides in scenario 2 and, the end-to-end delays are always lower in scenario 2.

As per the aforementioned analysis, we can conclude that the deployment of a broker is becoming necessary when we have high traffic in a “traditional” network. In other words, the broker plays a vital role when there is a bandwidth limitation with high network traffic.

Chapter 5: Conclusion

Fog computing is the term coined by Cisco and is also known as Fog Networking or Fogging. Fog computing is an end-to-end horizontal architecture that distributes computing storage, control, and networking functions closer to users along the cloud-to-thing continuum. It is important to note that fog networking complements - not replaces - cloud computing. The main goals of fog computing is to reduce data transition delay and the amount of data conveyed to the cloud for processing, analysis and storage.

The Fog network consists of devices called fog nodes and can be deployed anywhere with a network connection such as on a factory floor, on top of a power pole, alongside a railway track, in a vehicle, etc. Switches, routers, embedded servers, and surveillance cameras are some examples of fog nodes. In fact, any device with computing, storage, and network connectivity can be a fog node.

Many researchers use the terms for computing and edge computing interchangeably, as both involve bringing intelligence and processing closer to where the data is generated. However, the fundamental difference between the two is where the intelligence and compute power is placed. In a fog environment, intelligence is at the local area network. Data is transmitted from endpoints to a gateway where it is then transmitted to sources for processing and returning transmission. In edge computing, intelligence and power of the edge gateway or appliance are in devices such as programmable automation controllers.

Fog computing is more scalable and gives a better big-picture view of the network as multiple data points feed more data into it than the edge computing does.

As billions of devices get connected to the Internet, gigantic data is generated daily, which would be costly and time-consuming to send to the Cloud for processing and analysis. It is expected that 43 trillion gigabytes of data will be generated daily by 2020. Fog computing is often used, firstly, to reduce the back-and-forth communication between IoT devices and the cloud and apparently the bandwidth. Secondly, it is used because of the fact that latency in data transmission in many real-world IoT cases can be life-threatening such as in telemedicine and vehicle-to-vehicle communication systems, where milliseconds matter. Fog Computing plays an important role in the IoT time-sensitive applications. In literature, the majority of the IoT-Fog-Cloud paradigm are built in a tree-tiers architecture, where Layer 1 consists of the devices that generate data (e.g. IoT devices, sensors). Layer 2 consists of the Fog nodes and Layer 3 is about the Cloud services tier.

Up to the time we are writing this research, the number of papers that tackle the existence of a Smart Device between the IoT and fog network is very small; In addition, none of these studies have evaluated the efficiency of the deployment of a smart device within the IoT-Fog-Cloud computing paradigm. In this thesis, a simulation was built and many experiments were conducted in order to evaluate the efficiency of a broker in IoT-broker-Fog model.

5.1 Summary of the Main Results

Table 5-1: Main Results

Mu = 1 → Slow Network and K = 0.9 → High Traffic			
Broker Rate = Fog Node Rate		**Broker Rate > Fog Node Rate**	
Packets < 3000	Packets > 3000	Packets < 1500	Packets > 1500
Delays in Scenario 2 are better (70% of the time)	**Delays in Scenario 2 are always better (100%)**	Delays in Scenario 2 are better (80% of the time)	**Delays in Scenario 2 are always better (100%)**

The deployment of a Broker is absolutely beneficial when the arrival rate of the packets is fast, when there is high traffic in the network, in a slow network. In our simulation when packets arrival rate was set to 0.9 (maximum value → maximum traffic) and network rate was set to 1 (minimum value → network is slow), the scenario 2 always has the minimum end-to-end delay for all the batch period value (T=1 through T=10).

5.2 Main Contribution of the Thesis

Our provided experimental results justify the deployment of a smart device in IoT-Fog-Cloud computing to reduce the delay for time-sensitive applications. What benefits and when to use a smart device between IoT and Fog network had been successfully revealed.

5.3 Possible Extensions and Future Work

While our proposed model and experiments give a good insight into IoT-Broker-Fog computing model, there are opportunities for further research and several improvements can be made such as:

- The number of brokers to deploy in a specific application and the communication protocols between them need to be discovered.
- The model we have proposed is a very simplified structure which deals with one Broker and one Fog Node. In a real-world scenario, there might be several Fog Nodes. Therefore, as future work, an efficient node algorithm that could find the near-optimal node quickly and effectively without causing much delay is to be discovered.

Appendix A: Simulation Environment

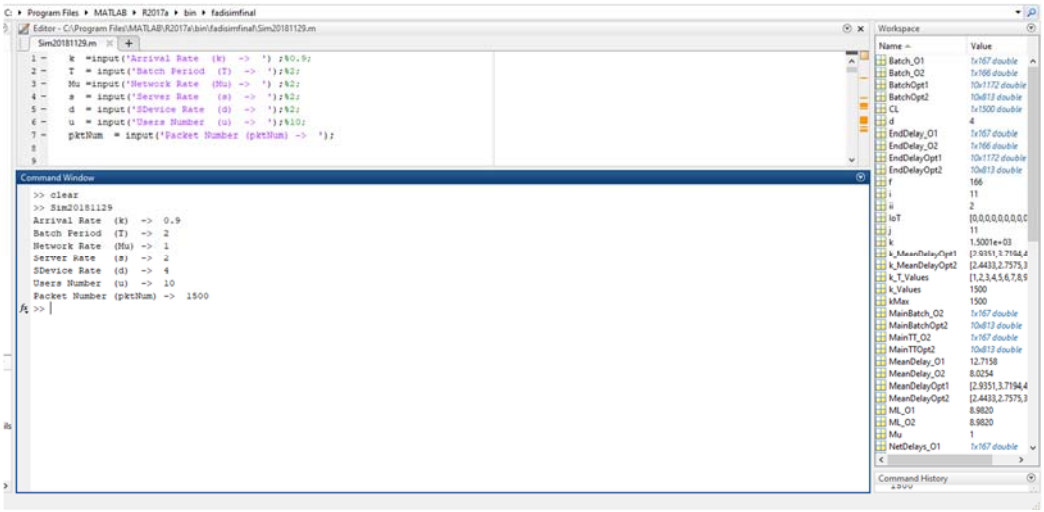


Figure Appendix A- 1: Sample Simulation Environment

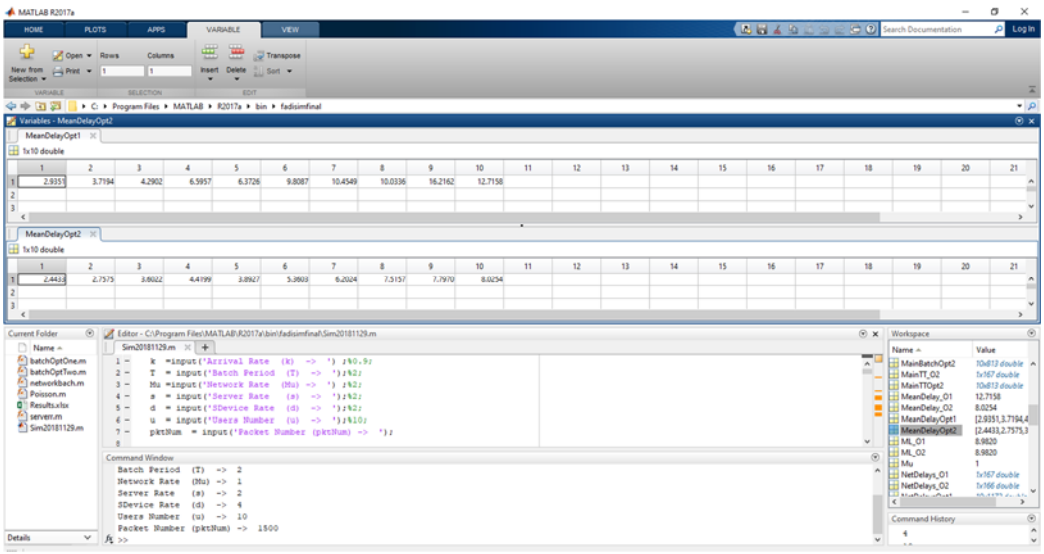


Figure Appendix A- 2: Mean Delay Results for 1500 Packets for T=1 through T=10 in both Scenarios

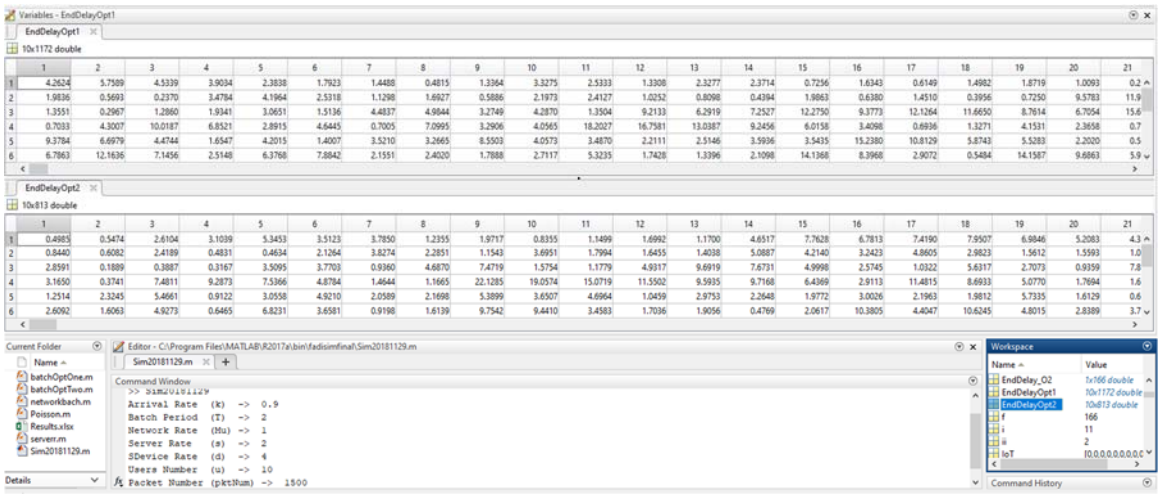


Figure Appendix A- 3: Sample of End Delays in Both Scenario 1 and Scenario 2

Appendix B: Sample Simulation Code

```
k  =input('Arrival Rate  (k)  ->  ') ;%0.9;
T  = input('Batch Period  (T)  ->  ');%2;
Mu =input('Network Rate  (Mu) ->  ') ;%2;
s  = input('Server Rate   (s)  ->  ');%2;
d  = input('SDevice Rate  (d)  ->  ');%2;
u  = input('Users Number  (u)  ->  ');%10;
pktNum  = input('Packet Number (pktNum) ->  ');%1500

%2-generating POISSON
P = Poisson(k,pktNum);

%3-initializing CL
CL = ones(1,pktNum);


%4-initializing PT
Z=pktNum;
for k = 1:Z
   PT(k) = k;
end


%5-initializing IoT users and packets
PKT = zeros(2,Z);
IoT = zeros(1,u);

vZeros = zeros(1,Z*0.2); % 20percent are normal
vOnes = ones(1,Z*0.8); % 80percent are abnormal
vTmp = [vOnes, vZeros]; %concatenate the two vectors
vStatus = randsample(vTmp, length(vTmp)); %shuffle the element of the vector

for i=1:Z
   PKT(1,i) = randi(u);
   PKT(2,i) = vStatus(i);
end


k=0.9;
kMax=k;
```

```
TMax=10;

ii=1;
while k<=kMax
   i=1;

   T=1;
   while T<=TMax
      T_Values(i)=T;

      %Option 1 Call
      [TT_O1,PT_O1,Z_O1,ML_O1,Batch_O1] =
batchOptOne(Z,T,P,PT,IoT,PKT,pktNum);
      NetDelays_O1 = networkbach(Mu, TT_O1, Z_O1, Batch_O1);
      Tnow1_O1 = NetDelays_O1 - TT_O1(1:Z_O1);
      Serv_O1 = serverr(NetDelays_O1,s,Z_O1);
      EndDelay_O1 = Tnow1_O1 + Serv_O1;
      MeanDelay_O1 = mean(EndDelay_O1);

      for f=1:Z_O1
         BatchOpt1(i,f) = Batch_O1(f);
         TTOpt1(i,f) = TT_O1(f);
         NetDelaysOpt1(i,f) = NetDelays_O1(f);
         Tnow1Opt1(i,f) = Tnow1_O1(f);
         ServOpt1(i,f) = Serv_O1(f);
         EndDelayOpt1(i,f) = EndDelay_O1(f);
      end

      MeanDelayOpt1(i) = MeanDelay_O1;

      %Option 2 Call

      [TT_O2,PT_O2,Z_O2,ML_O2,Batch_O2,MainBatch_O2,MainTT_O2] =
batchOptTwo(Z,T,P,PT,IoT,PKT,pktNum);
      SDServ_O2 = serverr(TT_O2,d,Z_O2);
      NetSendTT_O2 = SDServ_O2 + TT_O2;
      NetDelays_O2 = networkbach(Mu, NetSendTT_O2, Z_O2, Batch_O2);
      Tnow1_O2 = NetDelays_O2 - NetSendTT_O2(1:Z_O2);
      Serv_O2 = serverr(NetDelays_O2,s,Z_O2);
      EndDelay_O2 = SDServ_O2 + Tnow1_O2 + Serv_O2;
      MeanDelay_O2 = mean(EndDelay_O2);


      for f=1:Z_O2
         MainBatchOpt2(i,f) = MainBatch_O2(f);
```

```
        MainTTOpt2(i,f) = MainTT_O2(f);
        BatchOpt2(i,f) = Batch_O2(f);
        TTOpt2(i,f) = TT_O2(f);
        SDServOpt2(i,f) = SDServ_O2(f);
        NetSendTTOpt2(i,f) = NetSendTT_O2(f);
        NetDelaysOpt2(i,f) = NetDelays_O2(f);
        Tnow1Opt2(i,f) = Tnow1_O2(f);
        ServOpt2(i,f) = Serv_O2(f);
        EndDelayOpt2(i,f) = EndDelay_O2(f);
      end
      MeanDelayOpt2(i) = MeanDelay_O2;


      i=i+1;
      T=T+1;
    end

    k_Values(ii)=k;

    j=1;
    for j=1:length(T_Values)
      k_T_Values(ii,j)=T_Values(j);
      k_MeanDelayOpt1(ii,j)=MeanDelayOpt1(j);
      k_MeanDelayOpt2(ii,j)=MeanDelayOpt2(j);
      j=j+1;
    end



    ii=ii+1;
    k=k+0.1;
end
```

References

[1] *Industrial internet of things*. New York, NY: Springer Berlin Heidelberg, 2016.

[2] "Cisco Global Cloud Index Projects Cloud Traffic to Quadruple by 2019." [Online]. Available: https://newsroom.cisco.com/press-release-content?type=webcontent&articleId=1724918. [Accessed: 05-Dec-2018].

[3] "Data Never Sleeps 4.0 | Domo." [Online]. Available: https://www.domo.com/blog/data-never-sleeps-4-0/. [Accessed: 05-Dec-2018].

[4] L. Yehia, A. Khedr, and A. Darwish, "Hybrid Security Techniques for Internet of Things Healthcare Applications," *Adv. Internet Things*, vol. 05, no. 03, pp. 21–25, 2015.

[5] "Ashton: That 'internet of things' thing - Google Scholar." [Online]. Available: https://scholar.google.com/scholar_lookup?title=That%20%E2%80%98internet%20of%20things%E2%80%99%20thing&author=K.%20Ashton&publication_year=2009&pages=97-114. [Accessed: 22-Sep-2018].

[6] "IERC-European Research Cluster on the Internet of Things." [Online]. Available: http://www.internet-of-things-research.eu/about_iot.htm. [Accessed: 09-Sep-2018].

[7] M. Muntjir, M. Rahul, and H. Alhumiany, "An Analysis of Internet of Things(IoT): Novel Architectures, Modern Applications,Security Aspects and Future Scope with Latest Case Studies," *Build. Serv. Eng. Res. Technol.*, vol. 6, 2017.

[8] H. Atlam, A. Alenezi, R. J. Walters, and G. B. Wills, "An Overview of Risk Estimation Techniques in Risk-based Access Control for the Internet of Things," 2017, pp. 254–260.

[9] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[10] C. Everett Koop *et al.*, "Future delivery of health care: Cybercare," *IEEE Eng. Med. Biol. Mag. Q. Mag. Eng. Med. Biol. Soc.*, vol. 27, pp. 29–38, Jan. 2009.

[11] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," p. 7.

[12] "2018 Top Providers: Data Center, Colocation, Cloud Companies." [Online]. Available: https://www.datacenters.com/providers. [Accessed: 11-Sep-2018].

[13] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of Cloud computing and Internet of Things: A survey," *Future Gener. Comput. Syst.*, vol. 56, pp. 684–700, Mar. 2016.

[14] M. Iorga, L. Feldman, R. Barton, M. J. Martin, N. Goren, and C. Mahmoudi, "Fog computing conceptual model," National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 500-325, Mar. 2018.

[15] "Resources | OpenFog Consortium." .

[16] L. M. Vaquero and L. Rodero-Merino, "Finding your Way in the Fog: Towards a Comprehensive Definition of Fog Computing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, Oct. 2014.

[17] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog Computing: Platform and Applications," in *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, Washington DC, DC, USA, 2015, pp. 73–78.

[18] F. Bonomi and R. Milito, “Fog Computing and its Role in the Internet of Things,” *Proc. MCC Workshop Mob. Cloud Comput.*, Aug. 2012.
[19] “What is fog computing (fog networking, fogging)? - Definition from WhatIs.com,” *IoT Agenda*. [Online]. Available: https://internetofthingsagenda.techtarget.com/definition/fog-computing-fogging. [Accessed: 12-Sep-2018].
[20] S. Sarkar, S. Chatterjee, and S. Misra, “Assessment of the Suitability of Fog Computing in the Context of Internet of Things,” *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 46–59, Jan. 2018.
[21] “computing-overview.pdf.” .
[22] R. Mahmud, R. Kotagiri, and R. Buyya, “Fog Computing: A Taxonomy, Survey and Future Directions,” in *Internet of Everything*, B. Di Martino, K.-C. Li, L. T. Yang, and A. Esposito, Eds. Singapore: Springer Singapore, 2018, pp. 103–130.
[23] M. Mukherjee, L. Shu, and D. Wang, “Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges,” *IEEE Commun. Surv. Tutor.*, vol. 20, no. 3, pp. 1826–1857, thirdquarter 2018.
[24] M. Aazam and E. Huh, “Fog Computing and Smart Gateway Based Communication for Cloud of Things,” in *2014 International Conference on Future Internet of Things and Cloud*, 2014, pp. 464–470.
[25] Y. Ai, M. Peng, and K. Zhang, “Edge computing technologies for Internet of Things: a primer,” *Digit. Commun. Netw.*, vol. 4, no. 2, pp. 77–86, Apr. 2018.
[26] “whitepaper-fog-vs-edge.pdf.” .
[27] T. O. Consortium, “10 ways fog computing extends the edge,” *RTInsights*, 21-Aug-2017. .
[28] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos, “Challenges and Opportunities in Edge Computing,” in *2016 IEEE International Conference on Smart Cloud (SmartCloud)*, New York, NY, USA, 2016, pp. 20–26.
[29] open-idea, “Edge Computing in a brief - Open Ideas Ideation Blog - IBM Services Assets,” 03-Jun-2017. [Online]. Available: https://www.ibm.com/developerworks/community/blogs/open-idea/entry/Edge_Computing_in_a_brief. [Accessed: 16-Sep-2018].
[30] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.
[31] M. Satyanarayanan, “Pervasive computing: vision and challenges,” *IEEE Pers. Commun.*, vol. 8, no. 4, pp. 10–17, Aug. 2001.
[32] R. Balan, J. Flinn, M. Satyanarayanan, S. Sinnamohideen, and H.-I. Yang, “The case for cyber foraging,” in *Proceedings of the 10th workshop on ACM SIGOPS European workshop: beyond the PC - EW10*, Saint-Emilion, France, 2002, p. 87.
[33] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The Case for VM-Based Cloudlets in Mobile Computing,” *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct. 2009.
[34] A. M. M. Ali, N. M. Ahmad, and A. H. M. Amin, “Cloudlet-based cyber foraging framework for distributed video surveillance provisioning,” in *2014 4th World Congress on Information and Communication Technologies (WICT 2014)*, 2014, pp. 199–204.

[35] P. Mach and Z. Becvar, “Mobile Edge Computing: A Survey on Architecture and Computation Offloading,” *IEEE Commun. Surv. Tutor.*, vol. 19, no. 3, pp. 1628–1656, thirdquarter 2017.

[36] C. Dsouza, G. Ahn, and M. Taguinod, “Policy-driven security management for fog computing: Preliminary framework and a case study,” in *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*, 2014, pp. 16–23.

[37] V. Cardellini, V. Grassi, F. L. Presti, and M. Nardelli, “On QoS-aware scheduling of data stream applications over fog computing infrastructures,” in *2015 IEEE Symposium on Computers and Communication (ISCC)*, 2015, pp. 271–276.

[38] S. Dahmen-Lhuissier, “Multi-access Edge Computing,” *ETSI*. [Online]. Available: https://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing. [Accessed: 16-Sep-2018].

[39] M. Patel, D. Sabella, N. Sprecher, and V. Young, “Contributor, Huawei, Vice Chair ETSI MEC ISG, Chair MEC IEG Working Group,” p. 16.

[40] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, “A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges,” *IEEE Commun. Surv. Tutor.*, vol. 20, no. 1, pp. 416–464, Firstquarter 2018.

[41] D. Evans, “How the Next Evolution of the Internet Is Changing Everything,” p. 11, 2011.

[42] “IDC Forecasts Shipments of Wearable Devices to Nearly Double by 2021 as Smart Watches and New Product Categories Gain Traction,” *IDC: The premier global market intelligence company*. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prUS43408517. [Accessed: 18-Sep-2018].

[43] “Internet of Things Ecosystem and Trends,” *IDC: The premier global market intelligence company*. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=IDC_P24793. [Accessed: 18-Sep-2018].

[44] H. Atlam, R. J. Walters, and G. Wills, “Fog Computing and the Internet of Things: A Review,” *Big Data Cogn. Comput.*, vol. 2, Apr. 2018.

[45] G. Peralta, M. Iglesias-Urkia, M. Barcelo, R. Gomez, A. Moran, and J. Bilbao, “Fog computing based efficient IoT scheme for the Industry 4.0,” in *2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*, 2017, pp. 1–6.

[46] “FOG_Computing_and_Its_Real_Time_Applicat.pdf.” .

[47] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, “Mobile fog: a programming model for large-scale applications on the internet of things,” in *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing - MCC ’13*, Hong Kong, China, 2013, p. 15.

[48] A. V. Dastjerdi and R. Buyya, “Fog Computing: Helping the Internet of Things Realize Its Potential,” *Computer*, vol. 49, no. 8, pp. 112–116, Aug. 2016.

[49] B. Tang, Z. Chen, and G. Hefferman, “A Hierarchical Distributed Fog Computing Architecture for Big Data Analysis in Smart Cities,” p. 6.

[50] S. J. J. Kim, “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 *2012 International Symposium on Ubiquitous Virtual Reality*, 2012, pp. 1–5.

[51] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, “Fog Computing: principles, architectures, and applications,” in *Internet of Things*, Elsevier, 2016, pp. 61–75.

[52] J. K. Zao *et al.*, “Augmented Brain Computer Interaction Based on Fog Computing and Linked Data,” in *2014 International Conference on Intelligent Environments*, 2014, pp. 374–377.

[53] S. B. Nath, H. Gupta, S. Chakraborty, and S. K. Ghosh, “A Survey of Fog Computing and Communication: Current Researches and Future Directions,” *ArXiv180404365 Cs*, Apr. 2018.

[54] K. P.Saharan and A. Kumar, “Fog in Comparison to Cloud: A Survey,” *Int. J. Comput. Appl.*, vol. 122, no. 3, pp. 10–12, Jul. 2015.

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

[56] S. Kitanov, E. Monteiro, and T. Janevski, “5G and the Fog — Survey of related technologies and research directions,” in *2016 18th Mediterranean Electrotechnical Conference (MELECON)*, 2016, pp. 1–6.

[57] M. Mukherjee *et al.*, “Security and Privacy in Fog Computing: Challenges,” *IEEE Access*, vol. 5, pp. 19293–19304, 2017.

[58] H. Dubey, J. Yang, N. Constant, A. M. Amiri, Q. Yang, and K. Makodiya, “Fog Data: Enhancing Telehealth Big Data Through Fog Computing,” p. 6.

[59] Y. Cao, P. Hou, D. Brown, J. Wang, and S. Chen, “Distributed Analytics and Edge Intelligence: Pervasive Health Monitoring at the Era of Fog Computing,” in *Proceedings of the 2015 Workshop on Mobile Big Data - Mobidata ’15*, Hangzhou, China, 2015, pp. 43–48.

[60] M. Aazam and E. Huh, “E-HAMC: Leveraging Fog computing for emergency alert service,” in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, 2015, pp. 518–523.

[61] T. N. Gia, M. Jiang, A. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, “Fog Computing in Healthcare Internet of Things: A Case Study on ECG Feature Extraction,” in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, pp. 356–363.

[62] M. Ahmad, M. B. Amin, S. Hussain, B. H. Kang, T. Cheong, and S. Lee, “Health Fog: a novel framework for health and wellness applications,” *J. Supercomput.*, vol. 72, no. 10, pp. 3677–3695, Oct. 2016.

[63] A. M. Rahmani *et al.*, “Exploiting Smart E-Health Gateways at the Edge of Healthcare Internet-of-Things: A Fog Computing Approach,” *Future Gener. Comput. Syst.*, Feb. 2017.

[64] D. Guibert, J. Wu, S. He, M. Wang, and J. Li, “CC-fog: Toward content-centric fog networks for E-health,” in *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2017, pp. 1–5.

[65] B. Negash *et al.*, “Leveraging Fog Computing for Healthcare IoT,” in *Fog Computing in the Internet of Things: Intelligence at the Edge*, A. M. Rahmani, P. Liljeberg, J.-S. Preden, and A. Jantsch, Eds. Cham: Springer International Publishing, 2018, pp. 145–169.

[66] A. Rahmani *et al.*, “Smart e-Health Gateway: Bringing intelligence to Internet-of-Things based ubiquitous healthcare systems,” in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, 2015, pp. 826–834.

[67] M. Etemad, M. Aazam, and M. St-Hilaire, “Using DEVS for modeling and simulating a Fog Computing environment,” in *2017 International Conference on Computing, Networking and Communications (ICNC)*, 2017, pp. 849–854.

[68] B. Varghese, N. Wang, D. S. Nikolopoulos, and R. Buyya, “Feasibility of Fog Computing,” *ArXiv170105451 Cs*, Jan. 2017.