Enhancing The Capabilities of IoT Based Fog and Cloud Infrastructures for Time Sensitive Events

Md. Fazla Rabbi Bhuiyan

ID: 2013-1-60-018

Fatema Tuz Zohora

ID: 2013-1-60-044

Md. Rezwanur Rahman Khan

ID: 2013-1-60-055

A thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

Department of Computer Science and Engineering
East West University
Dhaka-1212, Bangladesh

August, 2017

Declaration

We, hereby, declare that the work presented in this thesis is the outcome of the investigation performed by us under the supervision of Amit Kumar Das, Lecturer, Department of Computer Science and Engineering, East West University. We also declare that no part of this thesis has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

. . . .

(Amit Kumar Das)

Supervisor

Signature

. . . .

(Md. Fazla Rabbi Bhuiyan)

(ID: 2013-1-60-018)

Signature

. . . .

(Fatema Tuz Zohora)

(ID: 2013-1-60-044)

Signature

. . . .

(Md. Rezwanur Rahman Khan)

(ID: 2013-1-60-055)

Letter of Acceptance

This thesis report entitled *"Enhancing The Capabilities of IoT Based Fog and Cloud Infrastructures for Time Sensitive Events"* submitted by Md. Fazla Rabbi Bhuiyan (ID: 2013-1-60-018), Fatema Tuz Zohora (ID: 2013-1-60-044) and Md. Rezwanur Rahman Khan (ID: 2013-1-60-055) to the Department of Computer Science and Engineering, East West University is accepted by the department in partial fulfillment of requirements for the Award of the Degree of Bachelor of Science and Engineering on August, 2017.

Supervisor

....

(Amit Kumar Das)

Lecturer, Department of Computer Science and Engineering, East West University

Chairperson

....

(Dr. Md. Mozammel Huq Azad Khan)

Chairperson and Professor,

Department of Computer Science and Engineering, East West University

Abstract

Our life is becoming faster day by day by the blessing of the internet of things *(IoT)*. It plays a vital role to reduce the time to complete a computer-based or *IT* related work. Cloud and fog computing are one of the main elements of *IoT* which helps to provide service accurately in fewer times. A significant portion of the work related to fog and cloud computing indicates that fog computing is better than cloud computing. In this paper, we focus on time sensitive data like brain stroke, heart attack, accident, etc. and use fog computing to send the result to the user as early as possible. We have developed two algorithms in this book. First one is for choosing a fog if the user is in the overlapping portion of fogs. The second one is for user's location change and finding the shortest path among many fogs. At simulation part, we have done experiments on brain strokes data as an example of time sensitive data and also demonstrated the comparison between fog and cloud by doing experiments on execution time, energy consumption cost and network usage. The results of simulation experiments depict that regarding time sensitive things fog computing performs better than cloud computing.

Acknowledgments

As it is true for everyone, we have also arrived at this point of achieving a goal in our life through various interactions and help from other people. However, written words are often elusive and harbor diverse interpretations even in one's mother language. Therefore, we would not like to make efforts to find best words to express our thankfulness other than simply listing those people who have contributed to this thesis in an essential way. This work was carried out in the Department of Computer Science and Engineering at East West University, Bangladesh.

First of all, we would like to express our deepest gratitude to Almighty Allah for His blessings on us. Next, our special thanks goes to our supervisor "Amit Kumar Das" who gave us this opportunity and initiated us into the field of "*Enhancing The Capabilities of IoT bsed Fog and Cloud Infrastructures for Time Sensitive Events*" and without whom this work would not have been possible. His encouragements, visionaries, thoughtful comments, suggestions and unforgettable support at every stage of our BSc study were simply appreciating and essential. His ability to muddle us enough to finally answer our own question correctly is something valuable what we have learned and we would try to emulate, if we get the opportunity ever.

There are numerous other people too who have shown us their constant support and friendship in various ways, directly or indirectly related to our academic life. We will remember them in our hearts and hope to find a more appropriate place to acknowledge them in the future.

Md. Fazla Rabbi Bhuiyan
August, 2017

Fatema Tuz Zohora
August, 2017

Md. Rezwanur Rahman Khan
August, 2017

Table of Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Internet of Things

In recent years, Internet of things IoT took an important place in our daily life. Internet of things (IoT) is a network of Internet attached devices which can accumulate and interchange data operating embedded sensors. Due to its ability to gather, analyze and deliver data which can eventually be turned into essential information and knowledge, it has become one of the most crucial information technologies. In emergency management operation and response times,these technologies offer various advantages [1].

IoT is the essentially set of interconnected things, for example, sensor gadgets, labels, and keen protests over the Internet systems. All these gadgets must have the capacity to gather information and convey the same to every single other gadget sent over the globe. The fundamental concentration of IoT is to get data from the earth which can be shared among various different gadgets. The IoT speeds up awareness and response to events. In industries such as manufacturing, oil and gas, utilities, transportation, mining, and the public sector, faster response time can improve output, boost service levels, and increase safety.

1.1.1 Overview of IoT

The Internet of things covers a huge scope of industries and applications. Such as-

- Provide look at changed approaches to see an IoT innovation stack

- Dive into data organization, information administration and API cloud merchants
- Move into perception of bigger sum developments like machine learning and customer experience feedback and examination

1.1.2 Main System Components of IoT

There are some key components of Internet of Things (IoT):

- Sensing
- The local network
- The Internet
- The cloud
- The thing
- Communication
- Acting
- Analyzing
- Information delivery
- Interconnectivity

1.1.3 Internet of Medical Things

In healthcare, IoT plays a vital role. There is a term Internet of medical things IoMT. IoMT is the collection of medical appliances and application that connect to healthcare IT [2] system through online computer networks [3] [4]. The idea of interconnected healthcare globe gets more attractive with growing number of medical devices getting linked to the Internet. Now IoT devices are efficient enough to transfer significant sign data from a patient home to the hospital staff.

1.2 Cloud Computing

Cloud Computing [5] is one of the essential part of IoT . Cloud computing means accessing and accumulating data over the internet instead of computer's hard drive. A growing number of sensors have been enrolled with the development of the Cloud Computing has been used to accomplish the storage, processing and data analysis to inspect the intelligence of data [6] [7]. For managing IoT resources, it is suitable to use web infrastructure in Cloud. Cloud computing is giving developers and IT departments with the capacity to concentrate on what makes a difference most and dodge undifferentiated work like acquirement, support, and scope quantification. As cloud computing has developed in prevalence, a few distinct models and arrangement techniques have risen to help address particular issues of various clients. Each kind of cloud administration, and organization technique, furnishes you with various levels of control, adaptability, and administration.

Cloud computing [8] is the act of putting away frequently utilized computer information on servers that can be accessed through the Internet. It gives shared PC handling assets and information to PCs and different devices on request. For accessing to a shared pool which can be easily supplied and released with negligible attempts is a model for common empowering. Cloud computing depends on sharing of assets to accomplish intelligence and economy of scale, like a utility over a power arrange. In cloud computing, response time and service availability are two major factors concerning user experience. Relocating super-tasks to cloud assets or growing new super-tasks in a cloud-local environment.

1.2.1 Types of Cloud Computing

There are three key models for Cloud computing. Each model speaks to an alternate piece of the distributed computing stack.

- Infrastructure as a Service, shortened as *IaaS*, contains the essential building obstructs for cloud *IT* and normally give access to Computers,networking features and information storage space. Infrastructure as a Service furnishes you with the largest amount of adaptability and administration control over your *IT* assets and is most like existing *IT* assets that numerous *IT* departments and developers know about today.
- Platforms as a service expel the requirement for associations to deal with the hidden foundation (normally equipment and working frameworks) and enable you to concentrate on the arrangement and administration of your applications. This causes you be more proficient as you don't have to stress over asset acquisition, scope quantification, programming support, fixing, or any of the other undifferentiated truly difficult work associated with running your application.
- Software as a service *SaaS* is a product dissemination display in which an outsider supplier has applications and makes them accessible to clients over the Internet. *SaaS* is one of three fundamental classifications of cloud computing, alongside infrastructure as a service *IaaS* and platform as a service *PaaS*.

1.2.2 Advantage and Disadvantage of Cloud Computing

When data comes from a large number of sensors of heterogeneous devices at that time it's hard to maintain and compute that data streams, a platform is needed where on demand establishment can be done. Cloud gives that request establishment and can boost the computational capacity of IoT applications. As sensors are location depended, resource constrained and expensive. On the other hand, cloud-based platform is not location specified, much cheaper than physical resources. So, adopting cloud infrastructure for the IoT delivery is very much cost and time efficient. Moreover most of the Cloud data centers located far from the proximity of the end devices/users and also many data centers geographically centralized. As a result, the distant Cloud data centers respond the real-time and latency-sensitive computation service requests which are often facing large network congestion, service quality degradation,round-trip delay etc.[9] [10].

1.2.3 Objective of Cloud computing

The objective of distributed computing is to enable clients to take benefit by these movements, without the need for significant finding out about or capacity with each one of them. The cloud intends to cut expenses, and empowers the customers to concentrate on their middle business as opposed to being discouraged by IT tangles. The rule connecting with headway for appropriated registering is virtualization. Virtualization programming isolates a physical enrolling contraption into no less than one gadgets, each of which can be effortlessly utilized and comprehends how to perform planning tries. With working system level virtualization on a very basic level making an adaptable strategy of different autonomous figuring contraptions, now ready assets can be appointed and utilized more profitably.

1.2.4 Limitations of Cloud Computing

One of the main limitations of cloud is that, day by day the amount of data is increasing with the development of IoT which creates a burden on the cloud that is responsible for producing some problems. A new concept called ”Edge Computing” also known as fog computing has recently been suggested to resolve these problems besides centralized cloud computing.

1.3 Fog Computing

Fog computing [11] is a decentralized computing architecture where data is produced and acted upon that crucially extends the cloud computing and fetching the benefit and power of the cloud closer. Fog computing provides limited computing, storing, and networking services in the distributed way between end devices and the traditional cloud computing data centers [2].Fog is observing or analyzing real-time data from network connected things and then producing an action. Generally to get the immediate response we use fog.

The Internet of Things (IoT) is producing a remarkable volume and variety of data. But by the time the data when the information advances toward the cloud for investigation, the chance to follow up on it may be gone. This white paper, proposed for IT and operational innovation experts, clarifies another model for investigating and following up on IoT information. It is called either edge computing or Fog computing [12].

1.3.1 Main Task of Fog

- Examine the most time-touchy information at the framework edge, close where it is made as opposed to sending tremendous measures of IoT data to the cloud.
- Acts on IoT data in milliseconds, in perspective of approach.
- Sends chose to the cloud for undeniable examination and longer-term amassing.

1.3.2 Examples of Fog Applications

Fog applications are as assorted as the Internet of Things itself. What they have in like manner is checking or breaking down continuous information from arrange associated things and after that starting an activity. The activity can include machine-to-machine (M2M) correspondences or human-machine communication (HMI). Examples include locking an entryway, changing gear settings, applying the brakes on a prepare, zooming a camcorder, opening a valve because of a weight perusing, making a bar outline, or sending a caution to an expert to make a preventive repair. The conceivable outcomes are boundless.

1.3.3 Working Procedure of Fog

Engineers either port or compose IoT applications for fog nodes at the system edge. The fog nodes nearest to the system edge ingest the information from IoT gadgets. At that point-and this is urgent-the fog IoT application guides diverse sorts of information to the ideal place for investigation.

- The most time-delicate information is dissected on the fog node nearest to the things producing the information. In a Cisco Smart Grid distribution network, for instance, the most time-sensitive necessity is to confirm that security and control circles are working appropriately. Along these lines, the fog nodes nearest to the grid sensors can search for indications of issues and afterward prevent them by sending control commands to actuators.

- Data that can wait seconds or minutes for activity is passed along to an aggression node for examination and activity. In the Smart Grid case, every substation may have its own particular aggression node that reports the operational status of each downstream feeder and parallel.

- Data that is less time delicate is sent to the cloud for authentic examination, huge information investigation, and long haul stockpiling. For instance, each of thousands or a huge number of fog nodes may send periodic outlines of framework information to the cloud for authentic examination and capacity.

1.3.4 Utilization of Fog in Immediate Response

According to the World health Organization (WHO), 15 million deaths in 2015 are responsible for heart disease and stroke which are the world's biggest killers [13]. Every year there are 32.4 million heart attacks and strokes that lead to 15 million deaths worldwide. These diseases have persisted the leading causes of death in the last 15 years globally [14].

Immediate response is required for time sensitive events like Heart Attack, Brain Stroke, and Accident etc. Oxygen deprivation in the brain which damage the brain cells are responsible for brain stroke. On the other hand, heart attack happens when a section of heart muscle suddenly becomes blocked due to the lack of the flow of oxygen-rich blood. There are mainly two types of strokes. One is Ischemic stroke which typically occurs due to a blood clot that is the reason of the lack of blood flow to the brain. Another is Hemorrhagic stroke which is the result of the ruptured artery that sends blood spilling into brain tissue. These two types of stroke present almost similar symptoms. So it's difficult to find out the precise stroke externally. The more quickly a doctor can detect the stroke types, the more quickly they can provide treatment timely which can save patients from severe long term disability and also from death.

1.3.5 Stroke Finder and Lifepack15

Stroke Finder is an IoT device which helps to find out the particular stroke as soon as possible [6]. By obvious symptoms, it is not possible to determine which stroke has been occurred. Stroke finder can easily find out the precise stroke. On the other hand in terms of heart attack, physio-control is on the leading edge of the innovation in pre hospital emergency care with Lifepak 15. It is a portable heart monitor that can transmit EKG data over the internet [14].

1.4 Our Contribution:

- To process the incoming data even more quickly than cloud fog computing has been introduced.
- A comparison between cloud & fog has been demonstrated in our simulation part.
- Devices like Stroke Finder and Lifepak 15 have been integrated into Fog & Cloud infrastructures for global storage and access and to get immediate result.
- User request is send to either fog or cloud based on the request type that is immediate or non-immediate.
- If User is in the overlapping portion of multiple fogs then based on some criteria those fogs will decide under which fog the user will make request.
- User location change is identified by our proposed model and also the result is send to the user using shortest path.

1.5 Conclusion

Cloud and Fog computing are one of the main elements of IoT which helps to provide service accurately in fewer times. A large portion of the work related to fog and cloud computing, they indicate that fog computing is better than cloud computing. So, because of some limitations of cloud computing we brought fog computing in our architectures. Moreover as we focus on time sensitive data, fog computing perform better then cloud computing.

Chapter 2

Related Works

2.1 Introduction

Cloud [15] and Fog computing are one of the main elements of IoT which helps to provide service accurately in fewer times. A large portion of the work related to fog and cloud computing, they indicate that fog computing is better than cloud computing. In most of the work they used only cloud rather than fog but they do not consider execution time, energy consumption cost, network usage to prove that fog is better than cloud. Besides there were no previous work like if there are multiple fogs cover an area and user is in the overlapping portion at that time which fog will be chose and also if user change his/her location then how can that requested fog send him/her a result in short time.

2.2 Related Works:

In this part, we present some works which are closely linked to ours. We classify the related works to clarify the difference of our work.

2.2.1 Importance of Fog

In [2], Hui Wang, Si Lu, H.Eduardo Roman delivered that It becomes a problem for the latency-sensitive applications when the cloud is relatively far from the customer. Moreover, they also mention that the amount of data and data nodes keep increasing with the evolution of IoT. In this paper, they discuss a new architecture to reduce the problems face in the cloud. So they think that fog computing is a good candidate to solve the problems of the cloud.

2.2.2 Immediate response

To send and receive information immediately, *IoT* is trying to link people and things based on the cloud as a complex cyber-physical system which integrates all kinds of sensing, networking, identification, information, communication and management devices into a system [16]. In this paper, as our topic related to immediate response, where each single second is imperative, we bring fog to send the product to the user immediately.

2.2.3 Stroke Finder an *IoT* Device

In order not to suffer from severe disability a stroke patient with a blood clot in the brain needs treatment within 3.5 to 4 hours. In one study, scientists from Chalmers University of Technology, Sahlgrenska Academy and Sahlgrenska University Hospital [17] [14] introduce an *IoT* device *"Stroke Finder"* which is responsible for determining the precise stroke as soon as possible. Stroke Finder is a device which includes a laptop that uses signal analysis algorithms to differentiate between ischemic stroke and hemorrhagic stroke [17].

2.2.4 *LifePak 15* an *IoT* Device

In [14], the *LifePak 15* is a portable device. It is a defibrillator that can send patient data through direct server-to-server integration to a hospital's secure health-care-information system. These data includes the patient's electronic record. *ER* physicians analyze that patient's record dictate a result whether the patient will be treated in house or must be referred to hospital.

2.3 Conclusion

According to this process, it might not be efficient. So in this book we proposed two algorithms. First one is for choosing a Fog if user is in the overlapping portion of fogs to send request. Second one is for location change and finding the shortest path among multiple fogs to send the response in fewer times to the user.

Chapter 3

Model and Assumption

3.1 Introduction

A large portion of the work related to Fog and Cloud computing, they indicate that fog computing is better than cloud computing. Various authors tried to represent their own established model to prove the comparison of fog and cloud. We have also tried to represent our model and assumptions.

3.1.1 Three phases

In Figure 3.1, there are mainly three phases. In our architecture, each object plays a different role in communication capabilities, computational, and the functionalities.

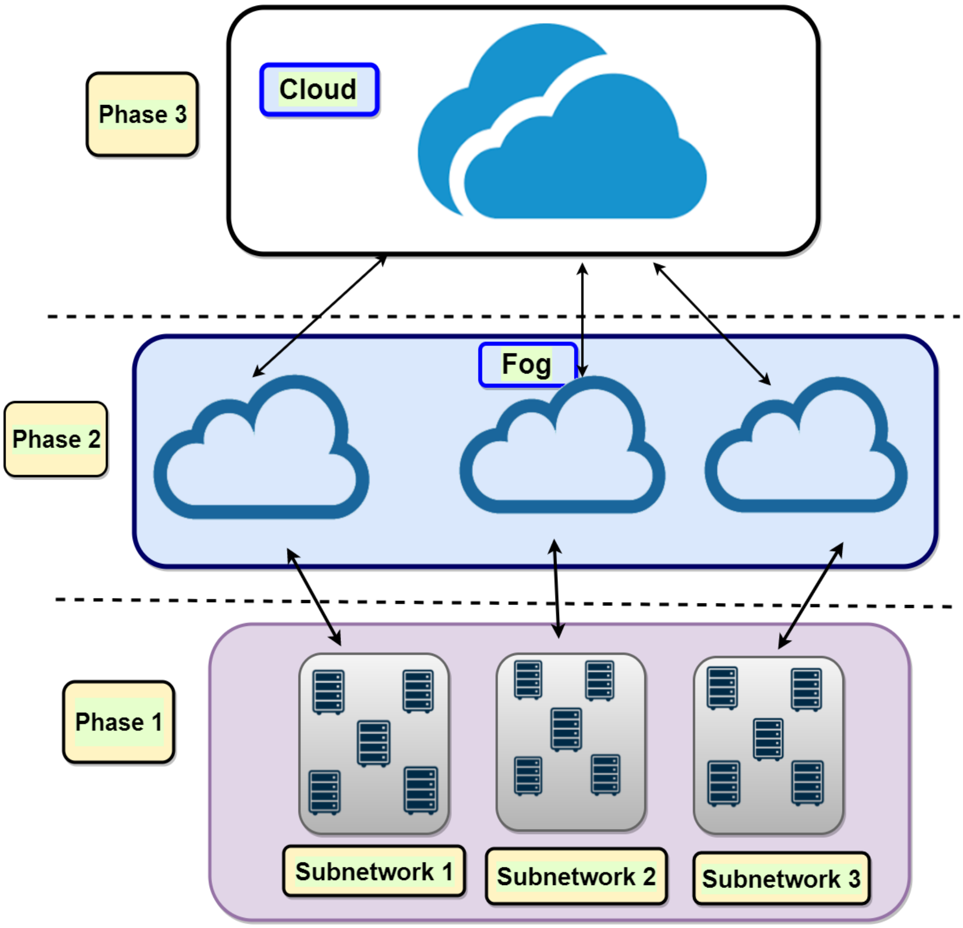


Figure 3.1: Proposed fog and cloud architecture

3.1.2 Description of Proposed Model

In the first step, there are some multiple subnetworks which are consists of some IoT devices and sensors. There is a smart gateway in each IoT devices and sensors which work independently.

In our architecture in the second step, we use fog. Fog computing acts as an intermediate layer in between Cloud and subnetworks which are consist of IoT devices and sensors. Fog computing environment is consists of traditional networking components like proxy servers, switches, routers,etc. [12]. To create broad geographical distributions of Cloud-based services the networking components enable Fog to do that. Regarding power consumption, service latency, network traffic, content distribution, Fog computing can perform efficiently. For this purpose, compared to a merely use of Cloud computing, Fog computing better meets the requirements of IoT applications [13]. For these reasons, we use Fog between cloud and subnetworks in our model.

In our model, in phase three we use Cloud [18] [19] to restore the patient's record which is working as a permanent database. A doctor can easily get the patient's history from the cloud and can provide treatment earlier.

3.2 Conclusion

In our architecture we demonstrated a figure where we can see in case of Immediate response user send data to the fog and fog will compute that result as early as possible. And In case of non immediate data user request will go to the cloud.

Chapter 4

Proposed Section

4.1 Introduction

In the previous chapter, we tried to represent about the structure of our model. In this chapter we will try to explain about how our model works and gives much better performance by giving almost error free result which will be very helpful for immediate response.

4.2 Description of Proposed Model

In case of time sensitive events like brain stroke, heart attack, accident etc. we can use Figure 4.1. A patient who has just faced brain stroke, it's imperative to know that which type of stroke has occurred to take some primary prevention as soon as possible to save that patient from death or severe long-term disability.

4.2.1 Detect the Particular type of disease

To find out the particular type of stroke or the condition of heart, the patient should be taken to the hospital immediately to use Stroke Finder and Lifepak 15, which are very time-consuming. To reduce that time we can send the data (through Sensors and IoT devices) required to know the condition of heart and to find the type of Brain stroke to Fog, then Fog can check if the data is time sensitive or not, if the data is time sensitive then fog will compute the result and send that result to the end user.

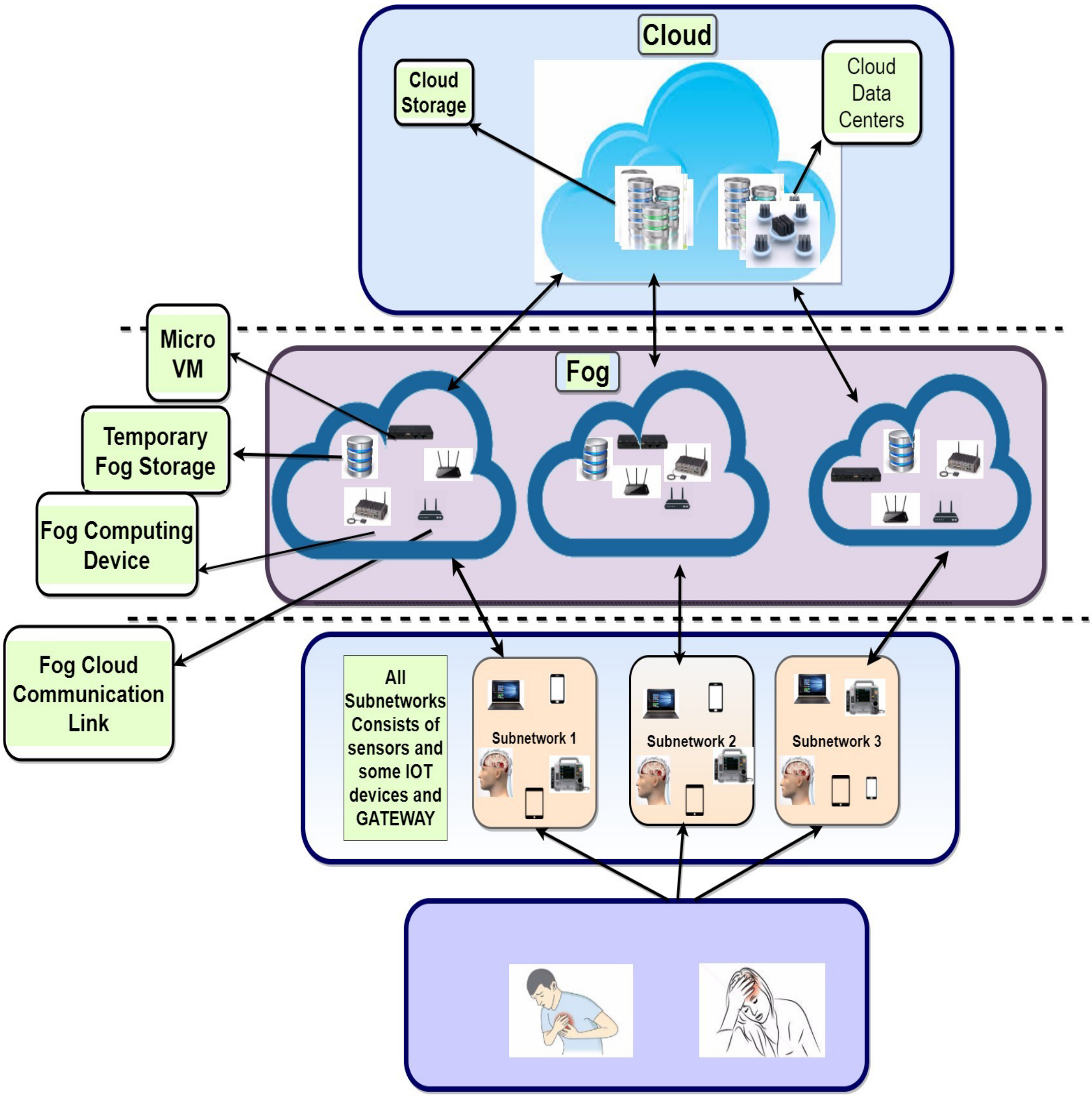


Figure 4.1: Described Architecture of Fog and Cloud using Sensors and IoT devices

4.2.2 Provide the result to the user

Within few minutes, patient's relatives will be able to know the type of brain Stroke and the state of heart. As a result they can take proper primary treatment. If the data is not time sensitive then it will send that data to the cloud and cloud will analyze and compute the result and send it to the end user. Here, cloud is also used as a permanent data storage. As these diseases are very time sensitive and within a few second anything can be happened, so we give priority to reduce the output time. So, in our architecture, we have used Fog computing between cloud and end devices as Fog computing can reduce the network traffic, scalability, Energy efficiency. As the goal of Fog Computing is to process incoming data closer to the data source itself, it reduces the duty of cloud to process all raw data sent to it. So, it saves a lot of time to give the output.

4.3 Flow chart of our proposed model

Figure 4.2 shows, when a request has been made (Step 1), that request will be passed to fog. Fog will collect the data through sensors and gateway (which is integrated with every IoT device) will pass the relevant data (Step 2). Fog will then check whether the data is time sensitive or not. If the request is not time sensitive then fog will pass the request to cloud to compute and store result (Step 5). Cloud will take the request and produce the result and will store it (Step 6 & 7). If fog finds that the request is time sensitive, then fog itself computes the result for that request. After producing the result fog will pass the result to the user and also store it in cloud (Step 4).

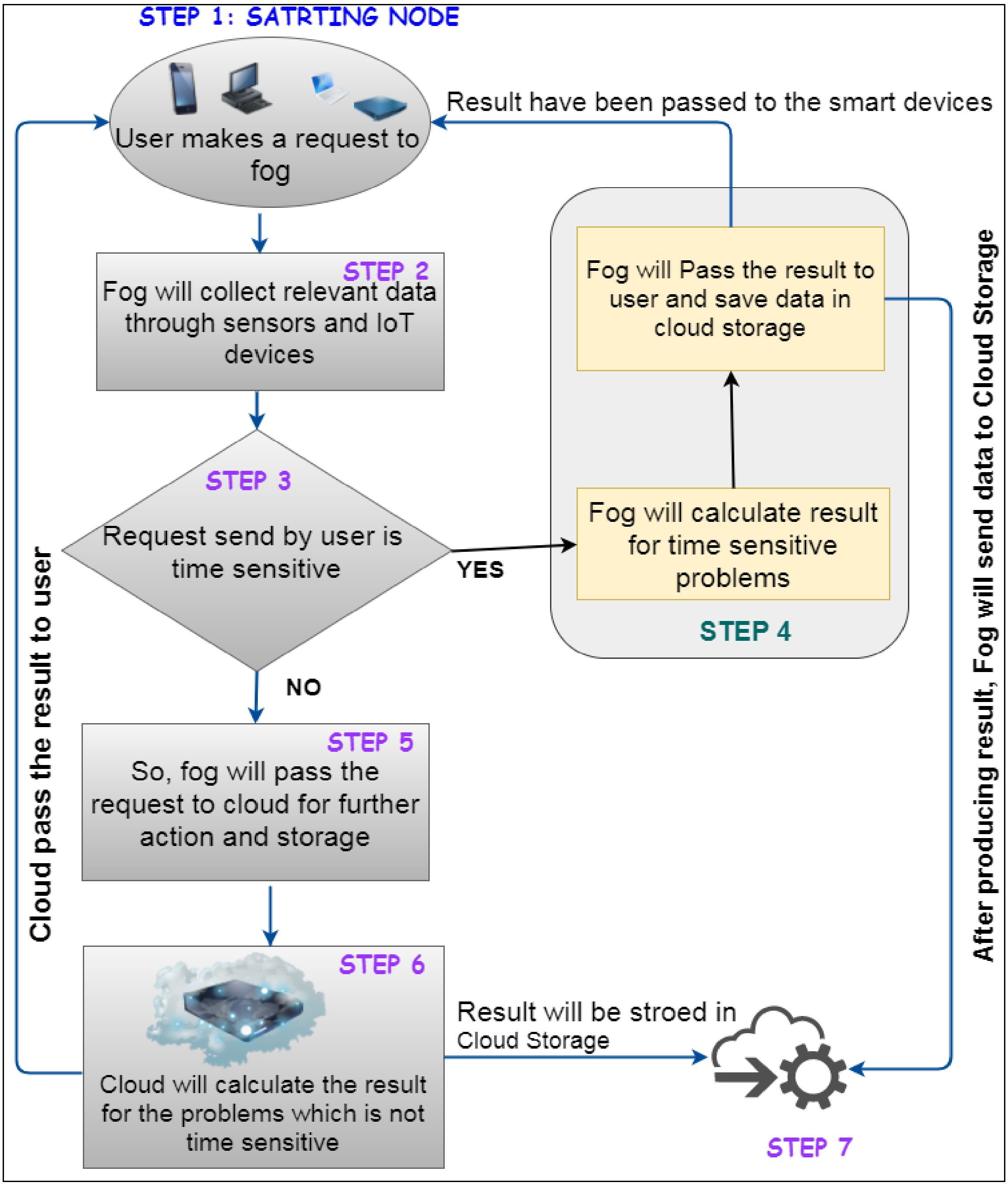
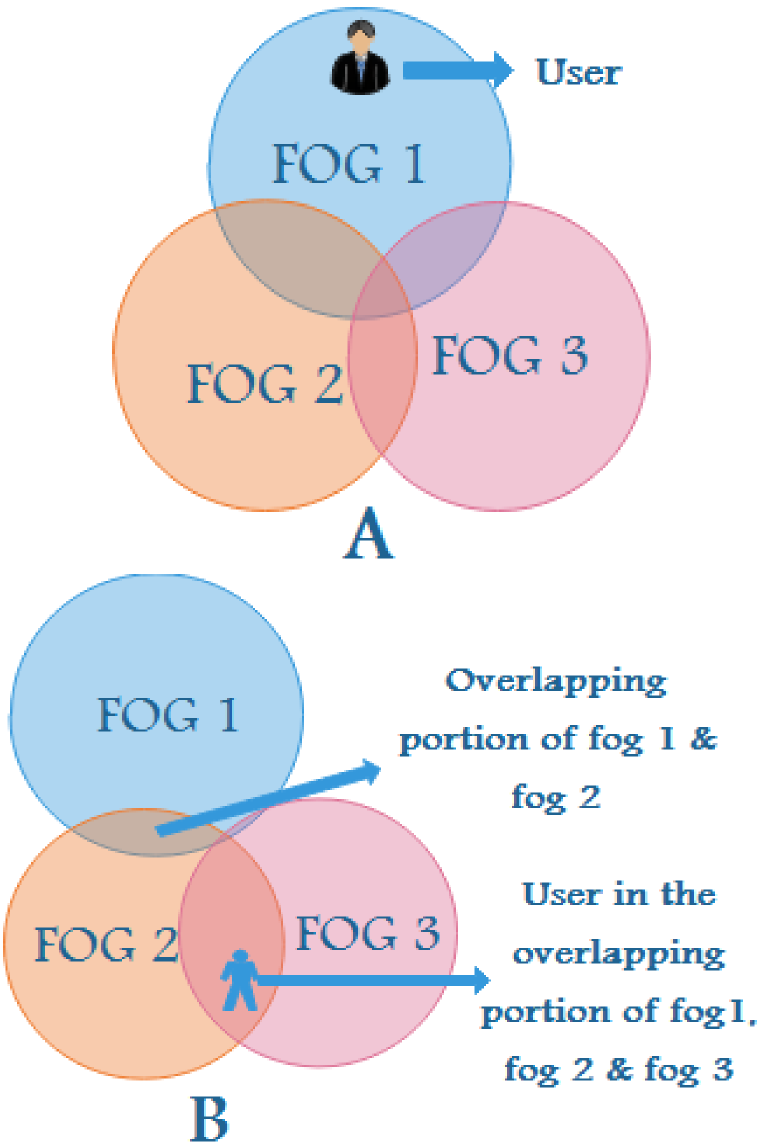


Figure 4.2: Flowchart of our proposed model

4.4 Model and Algorithm for Choosing Fog to Send Request



(A) User in a single fog (B) User in overlapping portion of fogs

Figure 4.3: User's Various Location in Fogs

As we are using fog computing for immediate response, so we need to use several fog systems to cover an area (depending on area size). To cover an area we may need ‘N’ number of fog computing systems. So fogs can overlap each other (Fig: 4.3(B)). In this case two scenarios can be considered: i) user stays in a single fog (Fig: 4.3(A)), ii) user is in the overlapping portion of 2 fogs (Fig: 4.3(B)). For (i) it is clear that user will send request to Fog 1. For (ii) if user is in overlapping portion of fogs, we are considering two things: a) the occupied space of a fog b) the finishing time of current job of a fog.

As per criteria (a) we consider that Fog 1 is 70% occupied with it’s current jobs and Fog 3 is 50% occupied. If request is made from the overlapping portion of Fog 1 & Fog 3 then the request will be forwarded to Fog 3. If Fog 1 & Fog 3 both are 70% occupied, then from criteria (b), request will be forwarded to that fog which will finish it’s current job early.

Algorithm 1 *Choosing Fog to Send Request*

INPUT: SF: User is in a single fog

MF: User is in the common part of multiple fogs

OS1: Occupied space of 1st fog

OS2: Occupied space of 2nd fog

FT1: Finishing time of current job of 1st fog

FT2: Finishing time of current job of 2nd fog

OUTPUT: Forward the request to a fog

```
if SF then
  request forward to that fog
else if MF then
  if OS1 < OS2 then
    Forward request to fog 1
  else if OS1 > OS2 then
    Forward request to fog 2
  else
    if FT1 < FT2 then
      Forward request to fog 1
    else
      Forward request to fog 2
    end if
  end if
end if
```

4.5 Model and Algorithm for Finding Shortest Path between Fogs

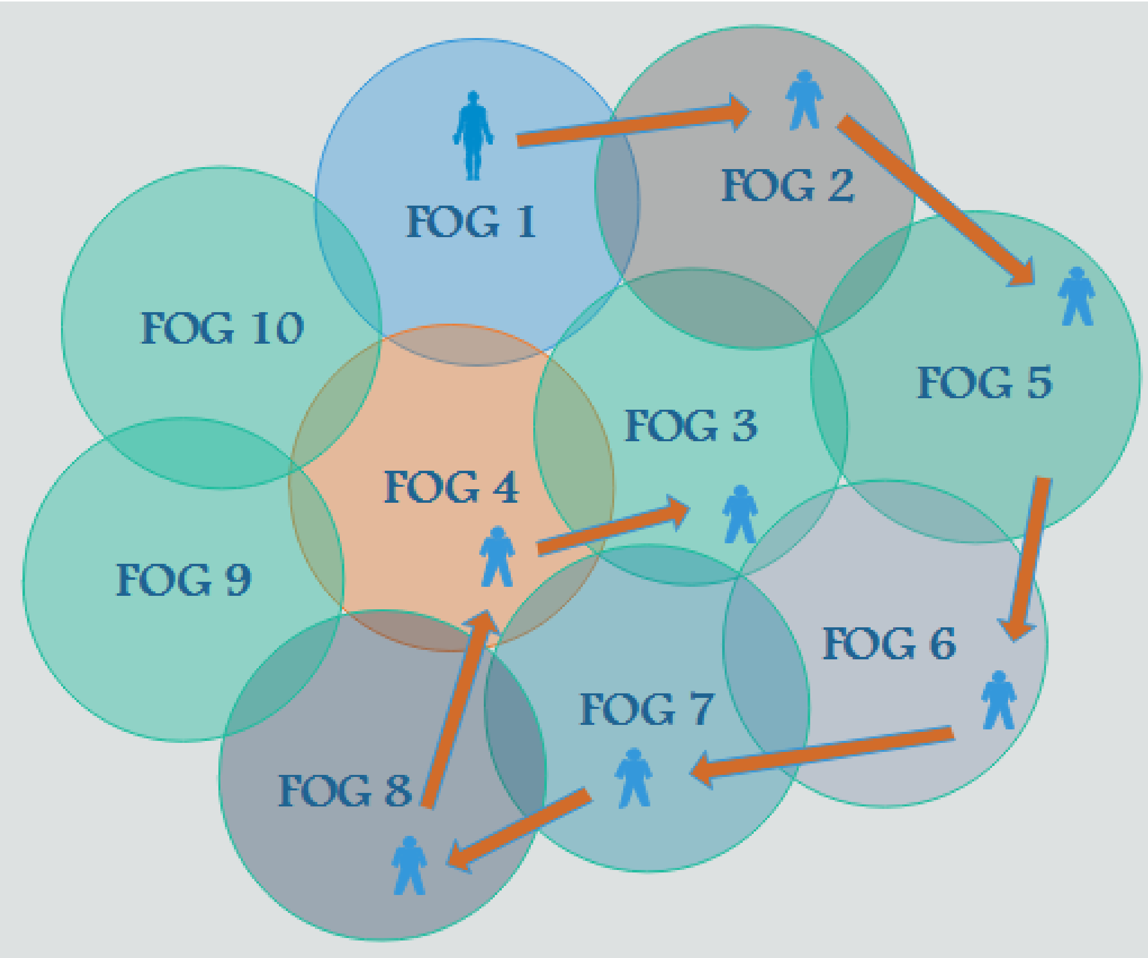


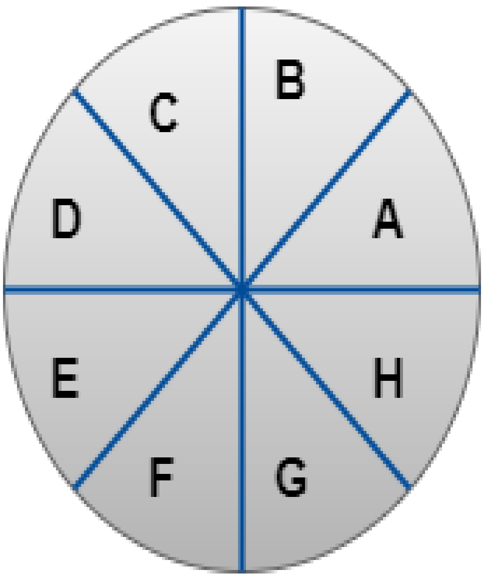
Figure 4.4: User is moving from one fog to another

4.5.1 Track user's current location

After sending request in fog, user may move to another fog. Figure 4.4 shows that user has send request in FOG 1, then moved to another fog. The path of user after sending request in FOG 1 is shown below:

FOG 1 → *FOG 2* → *FOG 5* → *FOG 6* → *FOG 7* → *FOG 8* → *FOG 4* → *FOG 3*

As result is preparing in FOG 1, so to provide that result FOG 1 needs to know the current position of user. After determining the position, FOG 1 needs to find the shortest path to send the result to user. To track user, we need to check every fog. That's why we divided the fog into 8 equal parts (A to H) of 45 degree (anti clockwise) in Figure 4.5, so that we can filter the fog. It will reduce the number of fogs for searching.



PART	ANGLE (IN DEGREE)
A	0 - 45
B	46 - 90
C	91 - 135
D	136 - 180
E	181 - 225
F	226 - 270
G	271 - 315
H	316 - 360

Figure 4.5: A fog is divided into 8 parts

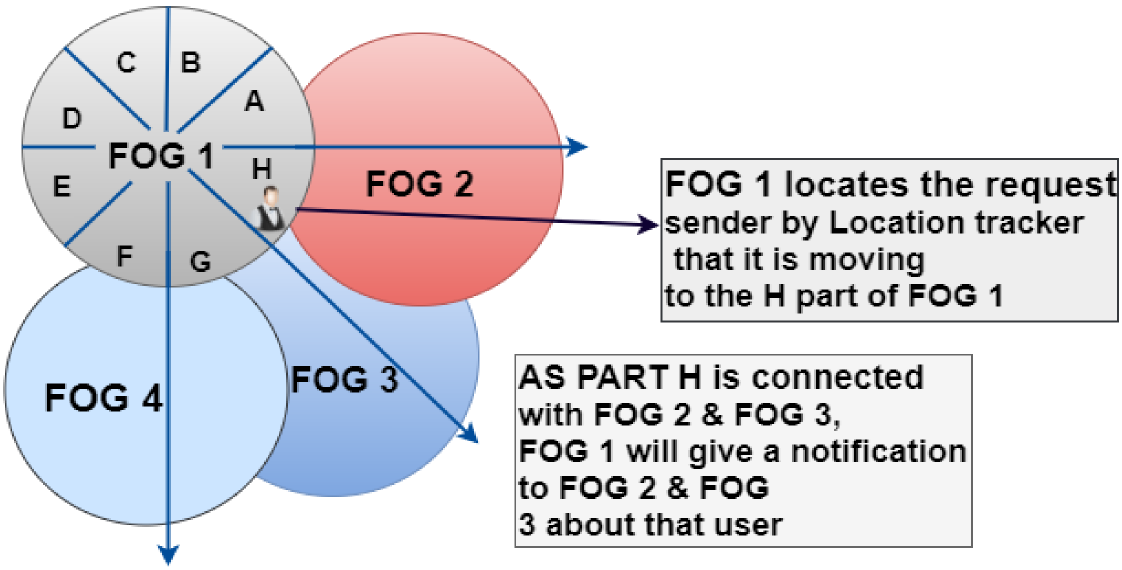


Figure 4.6: Tracking of user using angle

Figure 4.6 illustrates that, every part of a fog is connected to a finite number of fogs. Here, part H of FOG 1 is connected with FOG 2 and FOG 3. If FOG 1 finds that user was in H part last time, then it may go either FOG 2 or FOG 3. So, FOG 1 will push a notification to those fogs about the user. If user goes to FOG 2, then FOG 2 will give an acknowledgment to FOG 1. By this way, FOG 1 can trace the user. The following algorithm works for tracking the user location.

Algorithm 2 *User location tracking*

INPUT: Part: A-H part of a fog

ULP: User's last location Part in a fog

CF: Connected fog with a Part

Ack: Acknowledgment

CFN: Connected fog number

TRACK: Fog cannot track user in it's (fog) network

OUTPUT: User's current location

```
1. while TRACK=0 do
2.   Find ULP
3.   Send notification to the connected fogs
4.   if CF=1 then
5.     Ack=1
6.   end if
7.   Save CFN
8. end while
```

4.5.2 Finding shortest path between source *FOG* and user

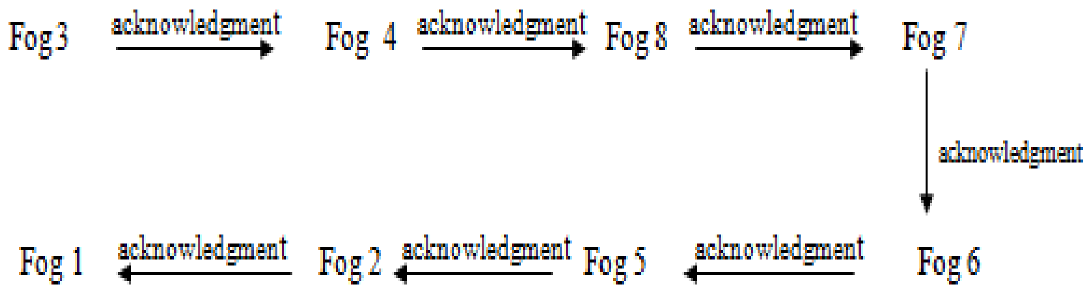


Figure 4.7: Back calculation

After getting user's location, FOG 1 now needs to have the shortest path to give the result to the user. For that from the back calculation illustrated in 4.7, FOG 1 will collect the geometric location of FOG 3. Then draw a imaginary straight line from FOG 1 to FOG 3 (Figure 4.8). After that FOG 1 will find out the connected fogs with that line and pass the result through those connected fogs.

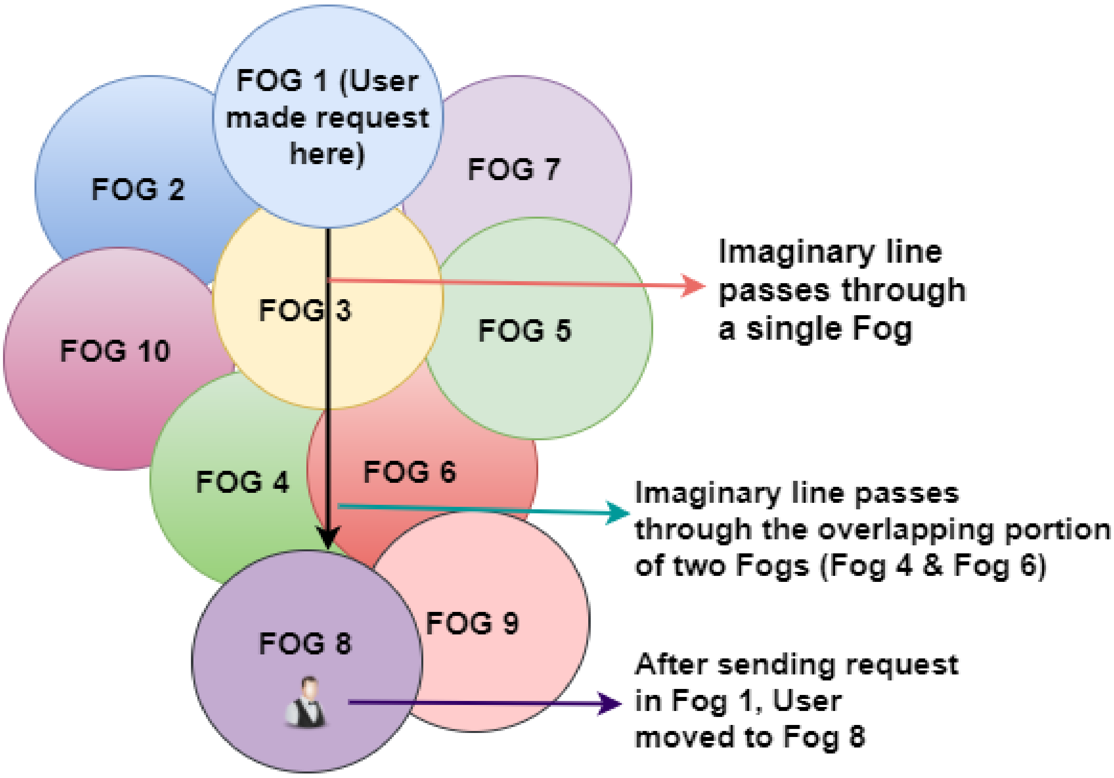


Figure 4.8: Shortest path from source fog to user's current position

Algorithm 3 *Send result using shortest path*

INPUT: SRC: The Fog which Contains the Result

GLU: Geometric Location of User

GLF: Geometric Location of Connected Fogs

DI: Distance from SRC to GLU

OUTPUT: Shortest way to reach the user

```
while DI != 0 do
  Find GLF
  if Fog A & Fog B are connected with that imaginary line then
    if Distance from Fog A to GLU < distance from Fog B then
      Result will pass through Fog A
      SRC = Fog A
      Update DI
    else
      Result will pass through Fog B
      SRC = Fog B
      Update DI
    end if
  else if Only one fog, Fog A is connected then
    Pass the result through that Fog A
    SRC = Fog A
    Update DI
  end if
end while
```

4.6 Conclusion

In this chapter we have explained our proposed model and algorithm. In our model, for time sensitive data we have used fog for computing result. After computing result, fog passes the result to user and store it in cloud storage for future use. This has been explained briefly in our flow chart. After that, we have proposed three algorithms. The first one shows, if a user is in the overlapping portion of multiple fogs, then how the requested is submitted to one fog. We have set two criteria by which request will be submitted to one fog. Base on that two criteria, we have developed an algorithm. Then we consider a critical situation, if a user sends request in $FOG\ A$, but after that user moves to other fog, then how FOG a will track the user and pass the result. The figures are well defined to illustrate this problem. To solve this problem, we have developed two algorithm. One algorithm works to track the user and, second algorithm shows the shortest path to the user from the source FOG. This model and algorithms ensure that user will get result in shortest time.

Chapter 5

Performance Evaluation

5.1 Introduction

Stroke Finder is based on microwave scattering estimations with a head antenna system where the antennas are spread all over the head and a display to show the diagnosis result. Individually, each antenna is employed as a transmitter, with the rest of the antennas as receivers. The generated brain signals are sent to the cloud for processing. But quick processing requires that the application be facilitated near the source of data, ideally on the device itself. However, such a sending would not permit worldwide scope which ordinarily requires the application to be hosted in the cloud.

5.2 Application Model

As represented in Figure 5.1, the Stroke Finder device comprises of two noteworthy modules which perform processing - Client and Scattered Wave Pattern Analyzer. The functions of the above mentioned modules are as follows:

5.2.1 Client

Client module gets raw brain signals by interacting with the sensor. It filters the received signal data and removes any apparently erroneous data. If the received signal data is error free then it sends the data to the Scattered Wave Pattern Analyzer module to get

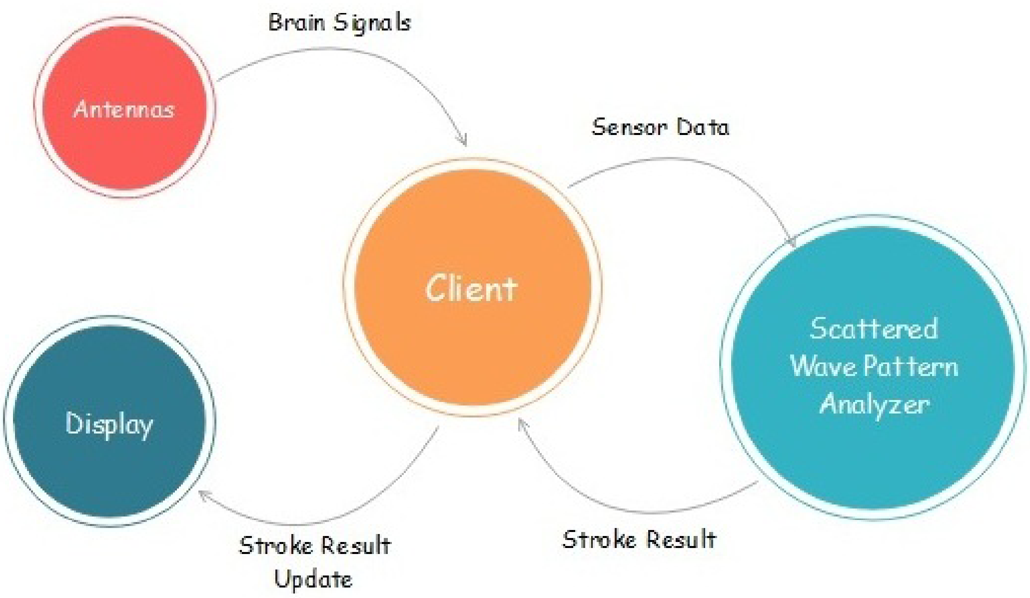


Figure 5.1: Application model of Stroke Finder

the stroke diagnosis result. When it gets the diagnosis result, it displays it to the actuator *DISPLAY*.

5.2.2 Scattered Wave Pattern Analyzer

The Scattered Wave Pattern Analyzer module is responsible for diagnosis of the brain signals. This module sends the diagnosis result to the Client module so that the diagnosis result can be displayed to the user.

Table 5.1 describes the properties of tuples which are carried by edges between the modules in the application.

Table 5.1: Description of inter-module edges in the Stroke Finder device

TUPLE TYPE	CPU LENGTH(MIPS)	N/W LENGTH
BRAIN SIGNAL	3000	500
SENSOR DATA	3500	500
STROKE RESULT	14	500
STROKE RESULT UPDATE	1000	500

5.3 Physical Network of Stroke Finder Device

There are 3 different types of fog devices that we have assumed for our physical topology as showed in Figure 5.2. The configurations of the fog devices that are used in the topology have been showed in Table 5.2. FogDevice, Sensor, PhysicalTopology and Actuator classes are used to design the physical topology in iFogSim [20].

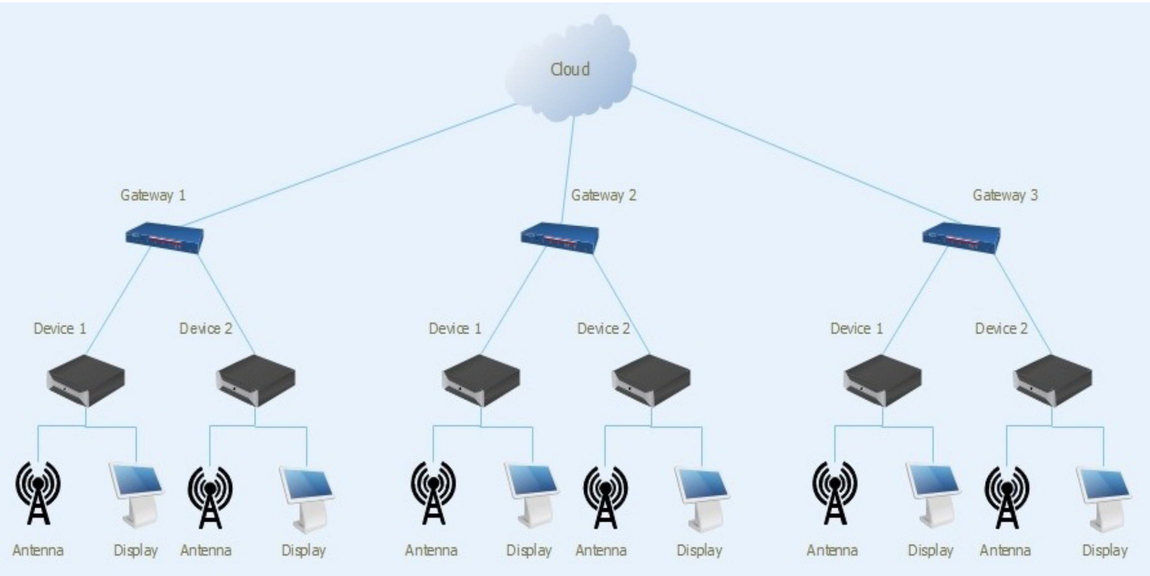


Figure 5.2: Network topology of Stroke Finder

Table 5.2: Configuration of fog devices for Stroke Finder

DEVICE TYPE	CPU(GHz)	RAM(GB)	POWER(W)
Cloud	3.0	40	1648(M) 1332(I)
Wifi Gateway	3.0	4	107.339(M) 83.433(I)
IoT Device	1.6	1	87.53(M) 82.44(I)

5.4 Experiment Results

iFogSim's performance was evaluated through different topology sizes where the number of WiFi gateways were varied keeping the number of Iot devices connected to each gateway fixed. Config 1, Config 2, Config 3, Config 4 and Config 5 having 2, 4, 8, 12 and 16 WiFi gateways respectively, with each gateway connected to 4 IoT devices these are the five configurations of physical topology that have been simulated. Latencies of the links connecting the Fog devices have vital impact on the performance of a fog application. Table 5.3 illustrates various kinds of devices that have various latencies among them.

Table 5.3: Description of network links for Stroke Finder

Source	Destination	Latency (in ms)
Antenna	*IoT* Device	6
IoT Device	Wifi Gateway	2
Wifi Gateway	Cloud	4

5.4.1 Average Latency of Control Loop

In this model, the most vital control loop in terms of latency of response is the loop accountable for altering the brain-signals of the user into stroke diagnosis result on the device's display. This enjoins real-time interaction between the IoT device and the device hosting the Scattered Wave Pattern Analyzer module along with competent processing on the module. The average latency of this control loop is showed Figure 5.3. It shows

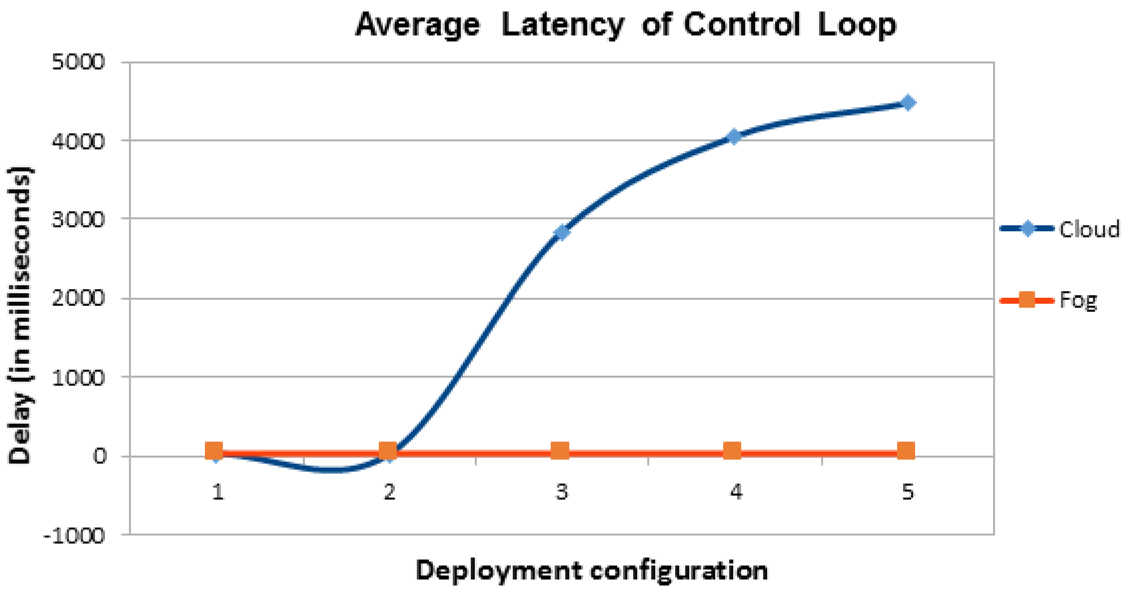


Figure 5.3: Average latency of control loop

that the latency of control loop severely reduces for Edge-ward placement strategy where fog devices are employed for processing. Increase in topology sizes and tuple emission rate makes the reduction even more evident.

5.4.2 Network Usage

Figure 5.4 illustrates the network usage of the Stroke Finder application. The pressure on the network where only cloud resources are used accelerates when the number of devices connected to the application notably accelerates. As Figure 5.4 demonstrates, the network usage significantly reduces when fog devices are taken into consideration.

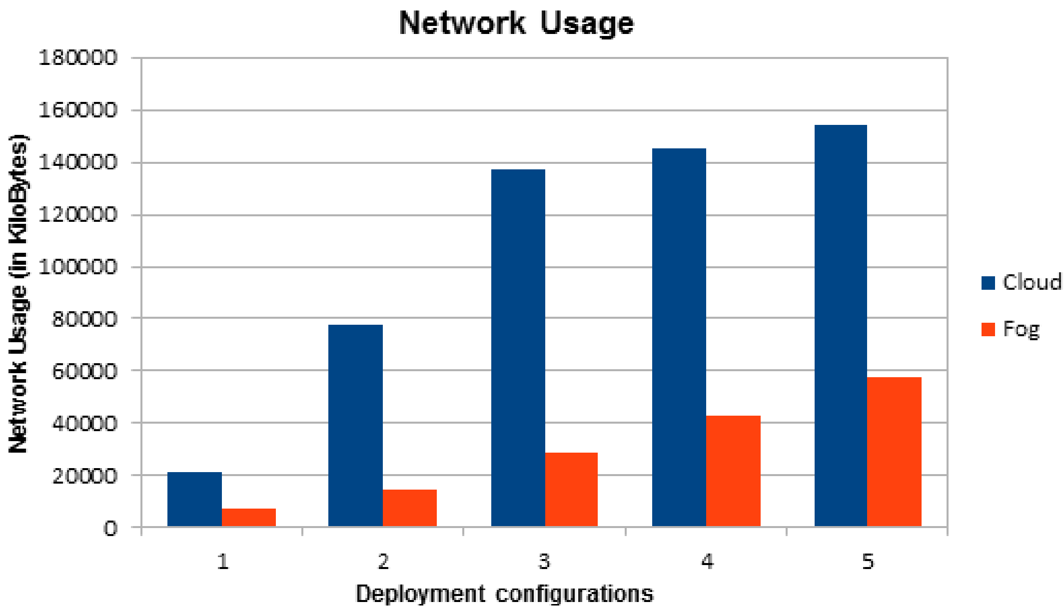


Figure 5.4: Comparison of network usage

5.4.3 Energy Consumption

The energy consumption by various types of devices in the simulation is depicted in Figure 5.5. As Figure 5.5 appears, using Fog devices in Edge-ward placement strategy decreases energy consumption of cloud data centers but marginally increases energy consumption of edge devices.

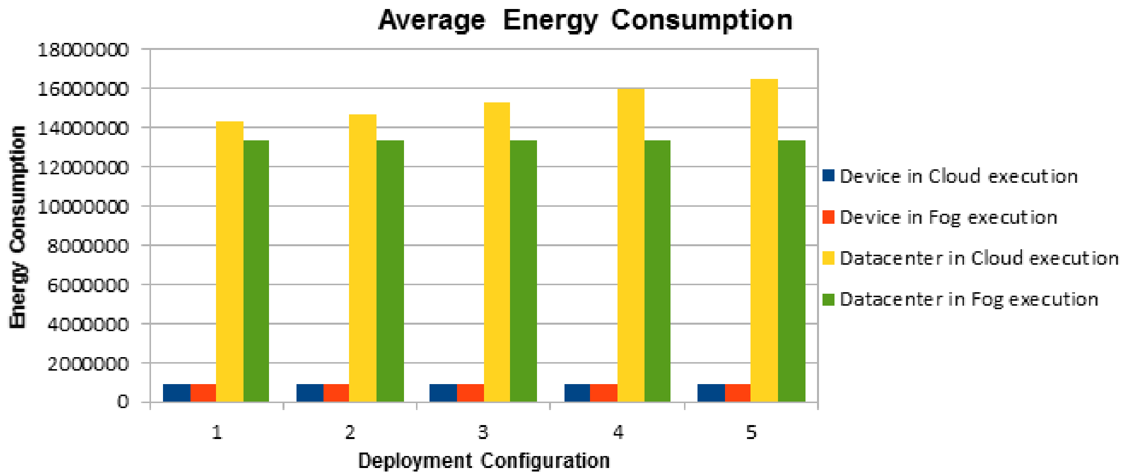


Figure 5.5: Energy consumption of devices in cloud and fog execution

5.4.4 Execution Time

Figure 5.6 illustrates the execution time for various topologies. It also demonstrates that with the increase in the number of devices and transmission rate, the execution time also increases. However, the expansion in simulation is almost linear and as a result simulation can be performed in an acceptable time (25 seconds) regardless of the possibility that a significant number of gateways are added.

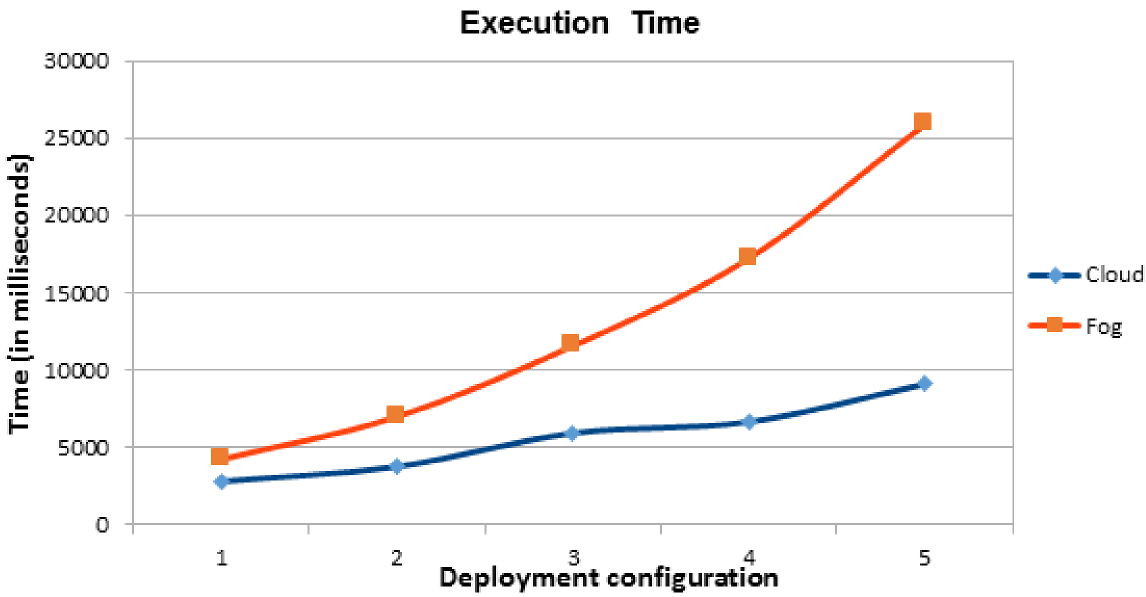


Figure 5.6: Execution time of simulation for varying sizes of topology and input workload

5.5 Conclusion

The efficiencies of the two module placement strategies (i.e. Cloud-only and Edge-ward) have been assessed in terms of latency, network usage, and energy consumption. Finally, the stability of iFogSim has been assessed in terms of execution time for various simulation models. The results of the simulation shows how the network usage and end-to-end latency were influenced by various input workloads and placement strategies.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

In this paper, we have presented the idea of using fog & cloud infrastructures for time sensitive events where immediate result is very much vital. We also used cloud as permanent data storage which enables global access to the result data regardless of time and location. We developed a smart algorithm to assign the user's request to a specific Fog in case user sends request from an overlapping portion of fogs, detect user's location change and send the calculated result to user using the shortest path in case the user moved away from the requested fog. In the performance evaluation section, we have done a comparison between fog and cloud infrastructures for our simulation model and demonstrated that fog is much faster and energy efficient than cloud in most of the cases. So, we have considered using fog to calculate the result for time sensitive events and cloud to calculate the result for events which are not time sensitive.

6.2 Future Works

In future, we have plans to work on the following issues:

- To make our proposed models and architectures more effective and efficient.
- Real life implementation of our proposed models and architectures.
- To cover more areas of interest using fog and cloud architectures.

Bibliography

[1] R. Sharma and N. Kumar, "Qos-alert markov chain based scheduling scheme in internet of things," *2015 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, Dec 2015.

[2] Y. Shi, G. Ding, H. Wang, H. E. Roman, and S. Lu, "The fog computing service for healthcare," *2015 2nd International Symposium on Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech)*, pp. 1–5, May 2015.

[3] "Iomt (internet of medical things) or healthcare iot," last accessed: 8 Aug, 17. [Online]. Available: http://internetofthingsagenda.techtarget.com/definition/IoMT-Internet-of-Medical-Things

[4] "Connected medical devices app: Are they leading the iot revolution-or vice versa," last accessed: 8 Aug, 17. [Online]. Available: https://www.wired.com/insights/2014/06/connected-medical-devices-apps-leading-iot-revolution-vice-versa/

[5] A. K. Das, T. Adhikary, M. A. Razzaque, M. M. Hassan, M. Alrubaian, and A. Alamri, "Big media healthcare data processing in cloud: A collaborative resource management perspective," *Cluster Computing*, vol. 20, p. 15991614, June 2017.

[6] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and it's role in the internet of things," *MCC '12 Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16, August 2012.

[7] A. K. Das, T. Adhikary, M. A. Razzaque, and C. S. Hong, "An intelligent approach for virtual machine and qos provisioning in cloud computing," *ICOIN '13 The International Conference on Information Networking*, January 2013.

[8] T. Adhikary, A. K. Das, M. A. Razzaque, A. Almogren, M. Alrubaian, and M. M. Hassan, "Quality of service aware reliable task scheduling in vehicular cloud computing," *Mobile Networks and Applications*, vol. 21, no. 3, pp. 482–493, Jun 2016. [Online]. Available: https://doi.org/10.1007/s11036-015-0657-5

[9] T. Adhikary, A. K. Das, and M. A. Razzaque, "Energy-efficient scheduling algorithms for data center resources in cloud computing," *HPCC '13 IEEE International Conference on. High-Performance Computing and Communications*, November 2013.

[10] M. S. Hoque, N. Shawkat, A. A. Chowdhury, and A. K. Das, "Triple-e: Energy efficient extended pool management scheme in cloud data centers," *IEEE International Conference on Innovations in Power and Advanced Computing Technologies, 2017 (i-PACT-2017)*, 2017.

[11] R. Mahmud and R. Buyya, "Fog computing: A taxonomy, survey and future directions," *CoRR*, vol. abs/1611.05539, 2016.

[12] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of internet of things," *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, pp. 1–1, 2015.

[13] "The top ten causes of death by world health organization," last accessed: 8 Aug, 17. [Online]. Available: https://www.who.int/mediacentre/factsheets/fs310_2008.pdf

[14] "How the internet of things may help save heart attack and stroke victims," last accessed: 8 Aug, 17. [Online]. Available: https://www.forbes.com/sites/ptc/2014/09/29/where-will-you-buy-the-internet-of-things/

[15] M. Akter, F. T. Zohora, and A. K. Das, "Q-mac: Qos and mobility aware optimal resource allocation for dynamic application offloading in mobile cloud computing," *ECCE '17 International Conference on Electrical Computer and Communication Engineering*, 2017.

[16] Z. Pang, L. Zheng, J. Tian, S. Kao-Walter, E. Dubrova, and Q. Chen, "Design of a terminal solution for integration of in-home health care devices and services towards the internet-of-things," *Enterprise Information Systems*, vol. 9, no. 1, pp. 86–116, 2015. [Online]. Available: http://dx.doi.org/10.1080/17517575.2013.776118

[17] "Stroke finder," last accessed: 8 Aug, 17. [Online]. Available: https://www.chalmers.se/en/Projects/Pages/Strokefinder.aspx

[18] T. Adhikary, A. K. Das, M. A. Razzaque, M. M. Hassan, and B. Song, "Quality of service aware cloud resource provisioning for social multimedia services and applications," *Multimedia Tools and Applications*, vol. 76, pp. 14 485–14 509, June 2017.

[19] A. K. Das, T. Adhikary, M. A. Razzaque, C. S. Hong, and E. J. Cho, "A qos and profit-aware cloud confederation model for iaas service providers," *ICUIMC '14 Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*, January 2014.

[20] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Software: Practice and Experience*, pp. n/a–n/a, spe.2509. [Online]. Available: http://dx.doi.org/10.1002/spe.2509

Appendix A

List of Acronyms

IoT	Internet of Things
IT	Information Technology
API	Application Program Interface
IoMT	Internet of Medical Things
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
QoS	Quality of Service
M2M	Machine-to-Machine
HMI	Human-Machine Interaction
ER	Electronic Record

Appendix B

List of Notations

! =	This is not equal to sign
=	This is equal to sign
<	This is less than sign
>	This is greater than sign
SF	User is in a single fog
MF	User is in the common part of multiple fogs
OS1	Occupied space of 1st fog
OS2	Occupied space of 2nd fog
OS1	Finishing time of current job of 1st fog
OS2	Finishing time of current job of 2nd fog
Part	A-H part of a fog
ULP	Users last location Part
ACK	Acknowledgment
CFN	Connected fog number
TRACK	Fog cannot track user in its (fog) network
SRC	The Fog which Contains the Result
GLU	Geometric Location of User
GLF	Geometric Location of Connected Fogs
DI	Distance from SRC to GLU

Appendix C

List of Publications

International Conference Paper

1. Fatema Tuz Zohora, Md. Rezwanur Rahman Khan, Md. Fazla Rabbi Bhuiyan and Amit Kumar Das, ‘‘Enhancing The Capabilities of IoT Based Fog and Cloud Infrastructures for Time Sensitive Events”, *International Conference on Electrical Engineering and Computer Science* (*ICECOS-2017*), Palembang, Indonesia, 2017. (Accepted)