SECURITY AND SCHEDULING IN FOG COMPUTING

Project report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Rishav Sharma (151249)

Under the Supervision of

Dr. Geetanjali

to

Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

CERTIFICATE

I hereby declare that the work presented in this report entitled **“Security and scheduling in Fog Computing”** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2018 to May 2019 under the supervision of **Dr.Geetanjali**(Assistant Professor Computer Science and Engineering). The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature)
Student Name: Rishav Sharma
Roll no: 151249

(Supervisor Signature)
Supervisor Name: Dr. Geetanjali
Designation: Assistant Professor (Senior Grade)
Department name: Computer Science and Engineering and Information technology
Dated:

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all those who have been helping me throughout the project and without whom this project would have been a very difficult task. I would like to thank all of them.

I highly indebted to Dr Geetanjali for her guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in doing my project. I would like to express my gratitude towards members of JUIT for their kind co-operation and encouragement which helped me in doing this project. My thanks and appreciations also go to our colleagues who have helped me out with their abilities in developing the project.

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

CONTENTS

Chapter1 Introduction

1.1 Introduction……………………………………………………………………1

1.2 Problem statement…………………………………………………………..6

1.3 Objective……………………….………………………………………………7

Chapter 2 Literature survey

2.1 Previously existing literature surveys on fog computing…………………………...9

2.2Previously existing literature surveys in security……….…………………………...9

2.3 Previously existing literature surveys in scheduling………………………………10

Chapter 3 System Design

3.1 cyber security system and algorithm …….…………………………………… 11

3.2 system design for scheduling……………………………………………………14

3.3How queuing and Priority works…………………………………………………...16

3.4How the priority model is going to work…………………………………………...17

Chapter 4 Performance analysis……………………………………………………...21

Chapter5 Conclusion…………………………………………………………………..49

References………………………………………………………………………….51

LIST OF THE FIGURES

Figure1.1: Fog network…………………………………………………………2

Figure1.2: Architecture of the fog computing ………………………………….3

Figure1.3: Fire detection and firefighting system……………………………....4

Figure1.4: Working of the Fog…………………………………………………..5

Figure3.1: Elliptic Curves………………………………………………………12

Figure3.2: Elliptic curve observation……………………………………….....12

Fig.ure3.3: Server Virtualization in cloud and fog layer environment…………15

Figure4.1: window for creating the fog environment (topology)………………22

Figure4.2: Topology with 7 fog nodes and 12 users……………………………23

Figure4.3: Initial screen after the topology making window…………………...24

Figure4.4: Main configuration window……………………………………….25

Figure4.5: Data Centre configuration window………………………………….25

Figure4.6: Advanced window of the configure simulation……………………..26

Figure4.7: Configure internet characteristics window………………………….27

Figure4.8: Region Plot for topology with 7 fog devices and 12 users………….28

Figure4.9: overall Budget and time summary…………………………………..29

Figure4.10: Response time for no of task………………………………………29

Figure4.11: scheduling Time for case 1……………………………………...31

Figure4.12: Cost and security measure for case 1………………………………32

Figure4.13: Total Data center cost for case 1…………………………………..32

Figure4.14: Region graph for case1 without priority……………………………33

Figure4.15: overall Budget and Time summary without priority……………….33

Figure4.16: Scheduling time for system without priority……………………….35

Figure4.17: cost and security measure for system without priority……………..36

Figure4.18: main configuration for case 2……………………………………….37

Figure4.19: data center configuration for case 2………………………………..37

Figure4.20: advanced window for case2……………………………………….38

Figure4.21: Topology for the second case……………………………………...38

Figure4.22: main configuration window for case2 with priority……………….39

Figure4.23: data center configuration window for case 2 with priority………...39

Figure4.24: Advanced settings window for case 2 with priority……………….40
Figure4.25: Region map for case 2…………………………………………...40
Figure4.26: Summary for case with priority…………………………………41
Figure4.27: Scheduling time for fog nodes case 2 with priority ……………….42
Figure4.28: Cost and security measure for case with priority…………………..44
Figure4.29: data centers cost for case 2 with priority…………………………..44
Figure4.30: Overall Budget summary for case without priority………………..45
Figure4.31: Scheduling time for case2 without priority………………………..47
Figure4.32: Cost and security measure case2 without priority…………………48

LIST OF THE GRAPHS

Graph4.1: Task base average response time…………………………………………….30

Graph4.2: Load Balancing Rate for each Fog node in case1……………………………31

Graph4.3: Task base hourly average response for system without priority……………..34

Graph4.4: Load balancing rate for the system without priority…………………………35

Graph4.5: task base hourly average response time for case 2 with priority……………..42

Graph4.6: load balancing graphs for fog nodes case 2 with priority…………………….43

Graph4.7: Task base average response time for each user case2 without priority………46

Graph4.8: Load balancing rate case2 without priority…………………………………..47

ABSTRACT

Nowadays, We all are aware of cloud computing and its capabilities, cloud provides many services to the end users such as users can use cloud for storage purpose where the end user can store its data and can access that data from anywhere or as a platform to compile or run the programs but sill cloud has many limitations to the latency sensitive applications that is where latency is an issue and any delay can't be tolerated in the application. If any delay would occur in the application then there would be a major problem for example in the disaster management application if there is any delay in the transmission and processing of the data then the results might be dangerous. To deal with such kind of problems a new field is emerging in which a virtual layer between the cloud and the end devices known as the fog layer is introduced. There are many security issues that occur in the fog layer and an appropriate security algorithm is needed to deal with those security issues. For this we have to keep many things in mind such as the processing capabilities of the fog nodes that if a particular scheme is applied in the fog layer and the encrypted message size is too large then the processing of that packet in the fog nodes may take a large amount of time and the processing cost of the network may increase, also the overall time may increase. Our first goal is to analyze the various challenges in terms of cyber security principles and provide an appropriate encryption scheme of the fog computing. The overall cost of the real time applications is going to reduce because of the virtual layer between the cloud and the IOT/end user devices, fog layer is going to reduce the overall transmission time and the processing cost of the application, If an appropriate scheduling algorithm is implemented in the fog layer to schedule the tasks that is which request from the end devices is passed to cloud or which request is handled in the fog layer. There are many algorithms for the fog computing to reduce the overall cost of the fog network but our second goal is to implement the priority-based scheduling algorithm in the fog environment to show that this is going to reduce the overall cost of the application.

CHAPTER1
INTRODUCTION

1.1 Introduction

Cloud computing enables the end user with many capabilities such as storage, processing and provides many other services to the user. Suppose there is an application which require some processing power which our personnel computer is unable to provide for that purpose we can request cloud to use its processing power to complete our task. We can request to a use a software without installing it in our personal computer from cloud. Our devices have some limited space capacity we can buy cloud storage for storing our files, videos, photos etc. on the cloud and access those photos, files from anywhere in the world by logging into the cloud there are companies which provide these kinds of services such as AWS (amazon web services), microsoft Azure these two companies provide the infrastructure like servers, OS, virtual machine etc. Apprenda, Red hat openshift provides platform-based services for maintaining, testing and developing software's and software as a service are provided are provided by companies like google, salesforce in which user can connect to applications through internet. Due to some of the drawbacks of the cloud computing a field is emerging nowadays that is edge computing or the fog computing which brings the cloud and end devices together. In the fog computing a virtual layer is added between the cloud and the end devices. In other words, fog computing is not replacing the cloud computing but adding an addition layer to the cloud environment to overcome the weakness of cloud computing by providing real time services that support low latency.

Since in the last few years many IOT devices are getting linked to the fog layer and their security is becoming a big issue. The confidentiality of the data that is sent from end users to the fog nodes and the authentication or authorization is also an issue that is a particular client can access particular information while designing of an fog application we have to keep in mind these points in this chapter we are going to discuss about various security challenges, requirements and possible cryptography solutions for the fog computing , also discuss about scheduling problem in fog computing.

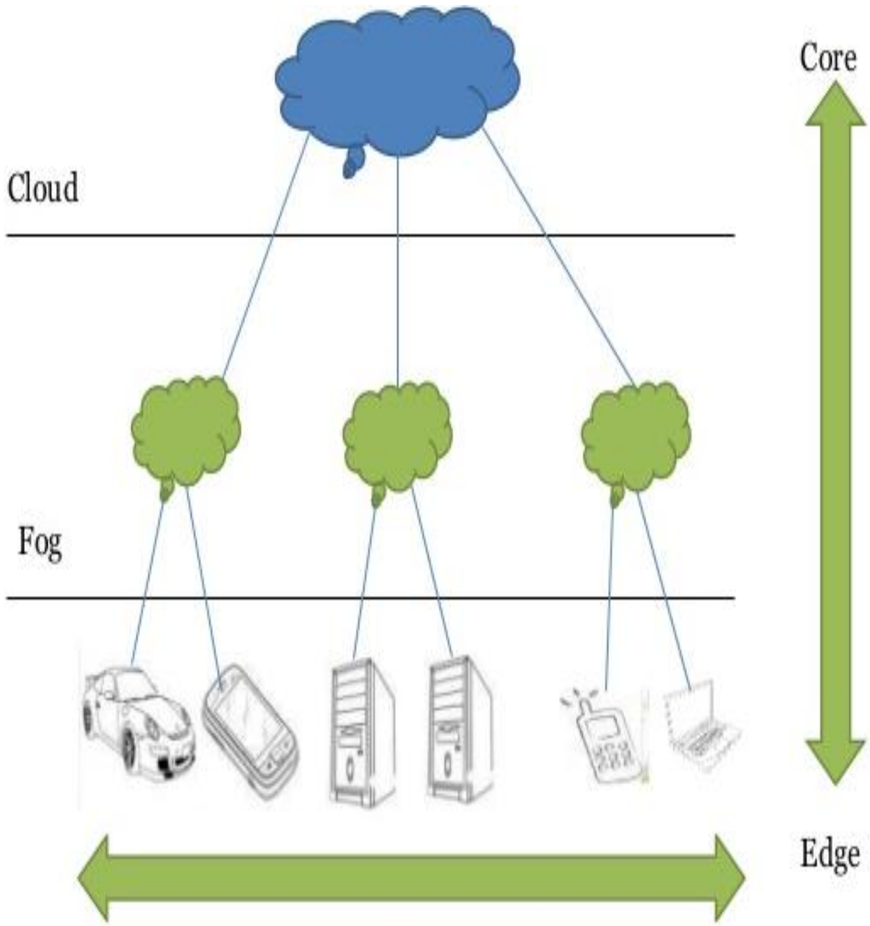


Figure1.1: Fog network

Above figure is showing the three layered network of the fog environment in this network the bottom layer consists of IOT or the user devices such as the mobile, CCTV cameras etc and the middle layer is the fog layer consisting of the fog nodes such as routers, switches ,embedded servers or we can say that any device that has computing, storage and network connectivity can be a fog node in the fog layer. The characteristic of the fog nodes are as follows:

1.Storage: The fog nodes have some storing capability and the storage in the cloud is not permanent.

2. Computation: The fog nodes should have some processing power to process the data before sending it into the cloud layer to reduce the latency in the network.

3.Network connectivity: The fog nodes should have the network connectivity means that the fog nodes can connect with other fog nodes and also with the cloud so efficient working of the network should be possible.

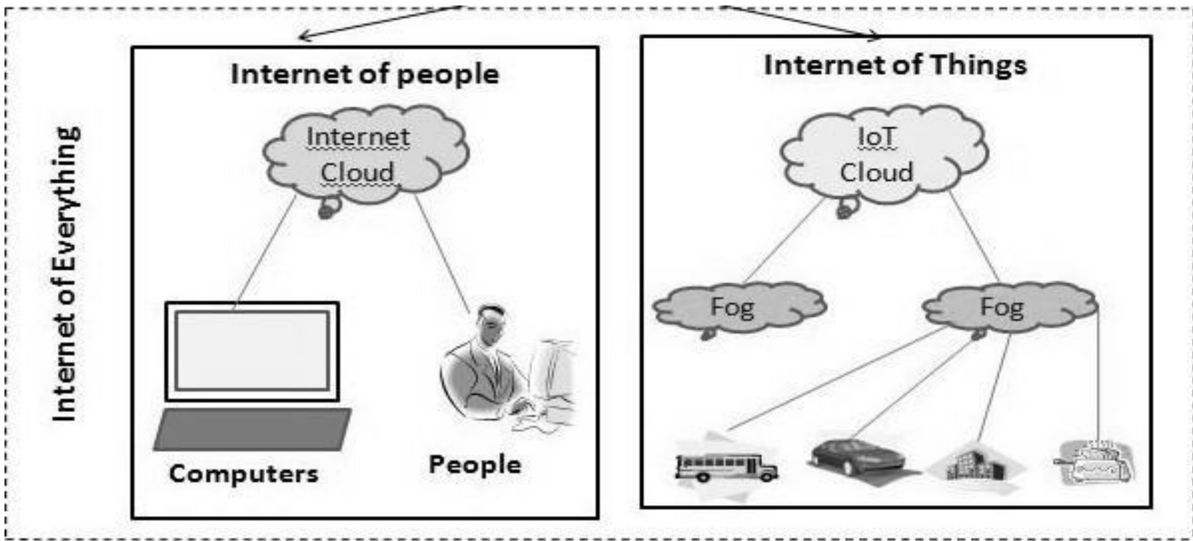


Figure1.2: Architecture of the fog computing

In the fog computing architecture, we are not going to remove cloud but a fog layer is added. In this we are going to bring storage, computation closer to IOT layer through introduction to fog.

Firstly, data is processed in fog layer if the data is time sensitive otherwise in the cloud layer. This architecture reduces latency, save bandwidth because we are not flooding the entire network with different packets, we are sharing bandwidth and storage at cloud because everything is not sent to cloud. Now let us discuss a use case for fog computing a fire detection and firefighting system.

This system is going to detect the fire in an environment by monitoring various factors such as the wind speed, moisture and temperature and the mechanism will dispatches a large group of the robots when fire is detected. Fire detection component, firefighting strategies module and Robot dispatcher module are the three main modules of the above use case system and these components are mainly located in the cloud that might be at a large distance from the IOT devices such as sensors which are present on the other end of the network. There might arise a problem of the delay in the transmission between the IOT devices and the components of the application.

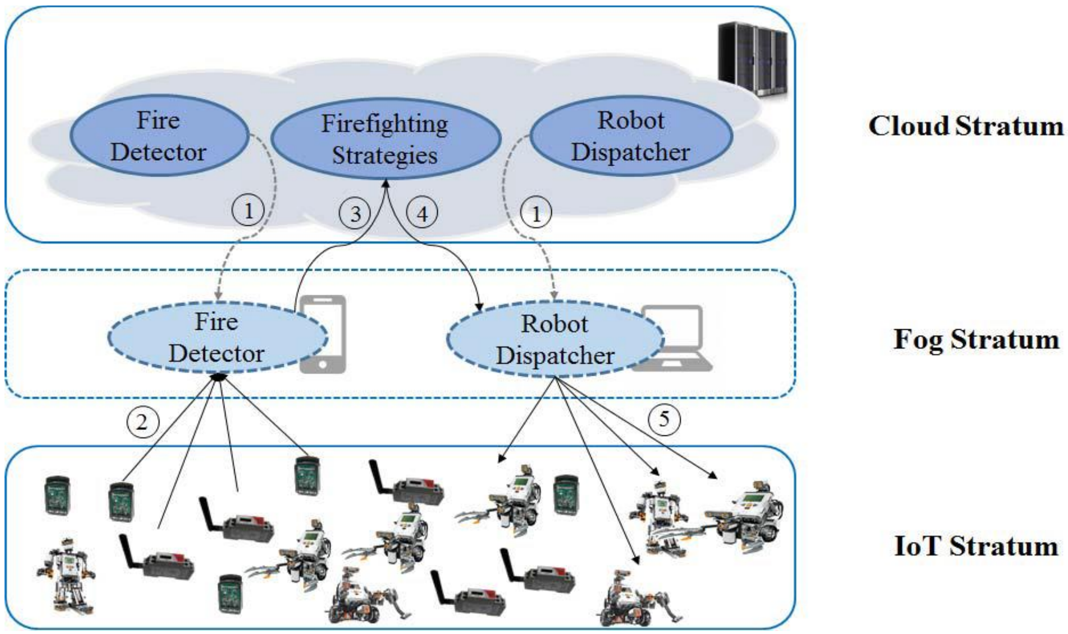


Figure1.3: Fire detection and firefighting system

Above figure shows the use case for the above discussed use case. Since there 3 components of the fire Detection and fighting system, 2 modules i.e. fire detector module and robot dispatcher module are moved from cloud stratum to the fog layer out of the three components then will helps in getting end to end latency constraints.

Real time data can be collected from the different sensors deployed in the edge of the fog computing environment and detect the fire if to locate the fire detector module at the fog layer and if this component detects the fire then this component notifies the firefighting strategies module which is present in the cloud layer to perform the firefighting strategies. After this firefighting strategy set up a communication with the robot dispatcher which will release the robots to extinguish the fire.

Now let us discuss about the working of the fog computing, The IOT devices sends 3 types of data:

- The data which is very time sensitive.
- The data which is less time sensitive.
- Third one is the data which is not Time Sensitive.

To the fog nodes and the fog nodes worked accordingly to the type of data they receive and for this fog node application is installed in the fog layer devices to handle this kind of data.

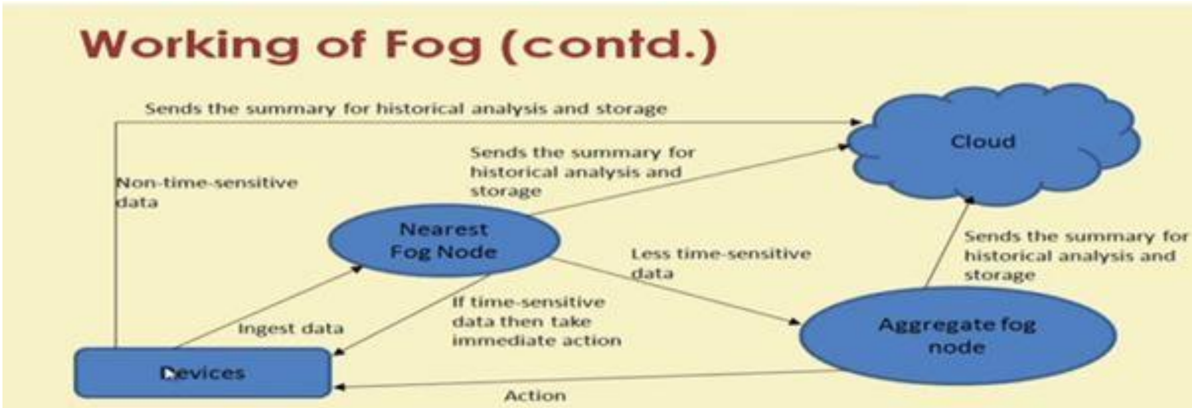


Figure1.4: Working of the Fog

- The data which is very time sensitive:

The nearest fog node takes the data from the devices. In case of most time-sensitive Data i.e. the data in which any kind of delay can't be tolerated. Firstly, it analyzes at the nearest node itself and then sends the decision or action to the devices and at last a copy of it sends and stores the summary to cloud for future analysis.

- The data which is less time sensitive:

Data which can be analyzed after sec or minutes. Data sent to the aggregate node for analysis. After analysis the aggregate node send the decision or action to the device through the nearest node then the aggregate node sends the detailed summary to cloud for storage purpose and for future use.

- Third one is the data which is not Time Sensitive:

Data which can be waited for hours, days, weeks is sent to cloud for storage and future analysis and then summarizes from fog nodes can be considered as less time sensitive data.

Now let us discuss the advantages of the fog computing:

- Security:
    I. Provides better security
    II. Fog nodes can use the same security policy

- Low operation cost:
    I. Processing of the data takes place at the fog node before sending this data to cloud.
    II. Fog computing reduces the consumption between the cloud and the client

- Reduces unwanted accidents:
    I. At the time of decision-making latency is reduced.
    II. Fog computing provide quicker decision making as comparison to the cloud computing.

- Better privacy:
    I. If the industry wants to analyze their data locally then they can do so.
    II. Confidential data can be stored in the local servers.
    III. Users can data that data which he/she wants to store in the cloud.

- Support mobility:

I. Nodes that are present in the fog environment are mobile since these fog nodes can join or leave the fog layer where ever they feel the need of it.

1.2 Problem statement

Cloud computing is a powerful tool but still it has some problems. In cloud the distance between cloud servers and the end users is a major problem which may lead to the increase in the delay and sometimes there may be the chances of shortage of resources for execution of the tasks and there may chances that some of the resources may remain idle these are some of the problems with the cloud computing. One of the major problems that comes in cloud is with the latency sensitive applications where any kind of delay is intolerable. To handle these problems in cloud computing fog computing helps a lot by adding a layer in cloud environment between cloud server and end users. Suppose we have an application for fire detection and handling in which any kind of delay would cause a major problem. In this application let us say that there are three main parts first part is for sensing the atmospheric changes in the environment, second part makes decision regarding decision making that is which action is needed to deal with the fire and the third part is robot dispatcher to dispatch the robots to extinguish the fire. Suppose second and third parts are present in the cloud and first part in end devices layer. First part senses the changes in environment like temperature, pressure changes etc. and send it to the cloud then decision is made there but if there is any kind of delay then this would lead to a major problem. To deal with this kind of problem we can put the decision maker part in the fog layer to resolve the latency problem.

After the fog layer is introduced in the fog computing environment the security challenges that are present in the traditional internet also present in fog computing environment. Some of the attacks that can happen in the fog environment are the impersonation, M-in-M, injection and the DOS attack. In the impersonation attack the third party can take the identity of a particular node and try to modify the message or can cause many kinds of privacy concerns in the network, the impersonation attack is one the common attack in the wireless network. Man, in the middle is the other kind of attack that is possible in the IOT use case, it is also a type of impersonation attack in which a mobile can take the identity of a sensor and can transmit any kind of information or we can say false information, in simple words in this attack the a 3rd party node intercepts the common between two nodes to capture the data. So, if we want to deal with security and privacy issues a strong identity management and encryption schemes are needed. There are security requirements for the cyber security such as confidentiality, integrity and availability that we have to keep in mind while developing a cyber security system. In confidentiality we have to ensure that the authorized user can only access the data that is transmitted and the unauthorized user can access the data in this way the privacy of the data is maintained. The integrity of the data makes sure that the data remains consistent, accurate. At last availability of data makes sure that the data always remains available even after the denial of service or we can say even after the system failure.

1.3 Objective

Nowadays large amount of data is generated by real time applications and many requests are made by the end users to the cloud at the same time. To handle these requests in an efficient manner a proper algorithm is required. Our first objective is to allocate the cloud resources efficiently to achieve the maximum throughput from the system. In the fog computing environment, a limited amount of the resources is available and request are made in a large amount from the end users to get maximum profit we have to allocate resources to the requests according to their priority levels. Just like in operating system proper scheduling algorithms are required in a similar way a scheduling algorithm is required in fog layer to handle the request such that no request go to starvation and each request gets a proper attention based on its priority. In simple words we can say that a request is made by end user if that request can be fulfilled in fog layer then it will not be proceeded to cloud layer but if fog devices are not able to fulfill its requirement then that request will be sent to the cloud. If all the resources of the fog layer are in use and a request is coming from the end users then that request have to wait till the resources are going to free but if proper scheduling algorithm is applied in fog layer then that request is diverted to the cloud. Our aim of the project is to implement a prioritized scheduling algorithm in the fog layer. This algorithm is based on the system model where a fog layer is present between end devices and cloud datacenters.

This model is best suited for those applications where high latency can't be tolerated. The fog layer many fog devices are present and with each fog devices micro datacenters are associated. In this model the fog devices can communicate with each other for the efficient resource allocation to the request that is made by the end user's and to balance the load on the fog layer. First of all, the request made by user is processed in the fog layer based on their priority levels but if all the micro datacenters in fog layer are saturated then this request is passed to the cloud layer. Our next objective is giving a proper light weight cryptography solution for the fog computing environment. One of the challenges in the security of the fog nodes is that the environment of the fog computing is distributed and we have to deploy a security scheme in the distributed environment and processing capability of the fog nodes is also very less as comparison to the cloud computing in which the processing is done by the centralized cloud. The implementation of the security mechanisms in the fog nodes is an ideal solution for security in the fog computing because fog computing is distributed network and the fog nodes are closer to the IOT or end devices. The resource limitation of these fog nodes is a major concern that we have to consider while designing a security mechanism for the fog computing environment.

CHAPTER-2
LITERATURE SURVEY

In this chapter we are going to discuss about the related works in fog computing and scheduling in the fog computing environment.

2.1 Previously Existing Surveys on fog computing

In recent years a large amount of work and research is conducted in this area. In 2010 Q. Zhang, L. cheng and R. Boutaba [1] have given a research which is based on cloud computing and various research challenges in it. X.zhu et al [2] have published a research paper in 2014 which shows how we can improve the video quality and its overall performance with the help of edge servers in the fog computing environment.

A three-layer architecture have been given by Agarwal, Yadav and Yadav [3] in year 2016. In this architecture a layer is introduced between cloud and end devices. Fog server, fog manager and virtual machines these three functional components are present in the fog layer and they also proposed an ERA (efficient resource allocation algorithm).Chin Tsai and Melody Moh in 2017[4] proposed the load balancing in 5G cloud radio access networks supporting IOT communication for smart communities. Most recently in 2018 the fog computing can help in 5G, the research paper by Gurpreet kaur and Melody Moh [5] have an algorithm in cloud Radio Access networks for efficient cache management.

2.2Previously Existing Surveys in security

M.Dakshayini and H.S Guruprasad [6] these two have given anoptimal model in 2011 for priority based service scheduling policy for cloud computing environment, this algorithm provides the real time results, provides the quality of services (QOS) and also helps in getting maximum throughput . An optimized algorithm for task scheduling based on activity costing in cloud computing is given by A. Ingole, S. Chavran and U. Pawde [7] in 2011. A latency aware resource allocation algorithm proposed by J.Xu, B.Palanisamy ,H. Ludwig and Q. wang for edge computing[8]. A task scheduling algorithm by R.singh,S.Paul [9] they have divided scheduling in two types first one is static and second is dynamic. A real time Efficient scheduling algorithm for balancing the load in fog computing environment by M.verma, N.bhardwaj[10].

C.Pawar and R.wagh[11] based on the parameters such as CPU time, memory and network bandwidth they have given a priority based dynamic resource allocation in cloud computing which is actually based on preemption mechanism. E. Elghoneimy ,O.Bouhali have given a resource allocation and scheduling algorithm in cloud computing[12]. Optimization of First come first server (FCFS) based algorithm by A. Marphatia, A.Muhnot ,T. sachdeva[13].

2.3 Previously Existing Surveys on Scheduling

Now we are going to discuss and analyze the related work in the cyber security schemes. In 2006 G.Ateniese, K.Fv,M.Green and S.Hohenderger[14] have improved the proxy re-encryption to secure the application that support the distributed storage. Fast proxy re-encryption for the publish systems is given by the Yuriy et al [15], since in the embedded systems the IOT devices the resources are limited that is why the proxy re-encryption is proved to be efficient. Key proxy re-encryption in [16] has proposed in 2009 but this scheme has an limitation, this scheme is inefficient for the embedded systems in which the resources are limited. Identity Proxy based re-encryption has been studied in [17] in which plaintexts encrypted is transformed from on identity to another. Z.wang [18] proposed an id based proxy re-encryption scheme for access control in the fog computing environment, in this the key leakage is protected from the channel side. A summary of Secure data sharing based on proxy re-encryption in fog computing environment is given by Y-J song and J-M kim [19] but in this paper there is deficiency of the results and the evaluation.

CHAPTER-3

SYSTEM DESIGN

3.1 The cyber security system and algorithm

Since our main is to design a security a scheme for the fog computing embedded systems in which the processing capabilities of the IOT devices is very less and also we have to deal with various security challenges that we have discussed in the chapter 1.One of the way to protect the data is the cryptography in which the plain text is first encrypted into the cipher text and after that the data is transmitted through the channel with a key and this message is received by the end side after that this message is decrypted into simple text using a key ,the receiver will then able to read the text. There are many algorithms for security such as RSA, ECC etc. but we are going to discuss about the ECC algorithm that is the elliptic curve cryptography algorithm which is going to use the properties of the elliptic curve to generate the algorithm.

Elliptic curve cryptography (ECC)[21] is a asymmetric cryptography this means that the encryption and decryption keys are going to be different in the system .We have chosen this algorithm for our system because this system is going to provides more security as comparison to the traditional RSA algorithm that is most widely used algorithm for security in the cloud computing because the key size that is used in ECC is much smaller than the key size of the RSA algorithm. If we want to define the elliptic curves then we can use functions with the degree three or we can say the cubic functions. The equation for the cubic function is given below:

$$y^2=x^3+ax+b$$

Elliptic curve cryptography can be used for many purposes such as the key exchange, encryption, decryption or the digital signature. The elliptic curves can be shown in the following figure:

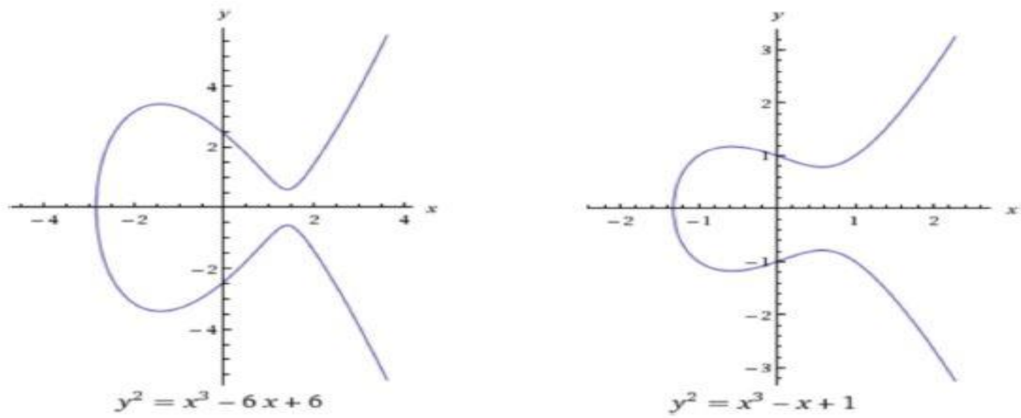


Figure3.1: Elliptic Curves

One of the properties of these elliptic curves is that these elliptic curves are symmetric in the horizontal direction (x-axis). And another property of the elliptic curves is that the line drawn on the elliptic curves is going to touch the curve at three different points, the line can be drawn up to infinity. But for our analysis we have to a limit to the curve suppose we are giving a limit of n to the curve for simplification and the line that we have drawn is touching the curve at points P, Q and R as shown as in the following figure:

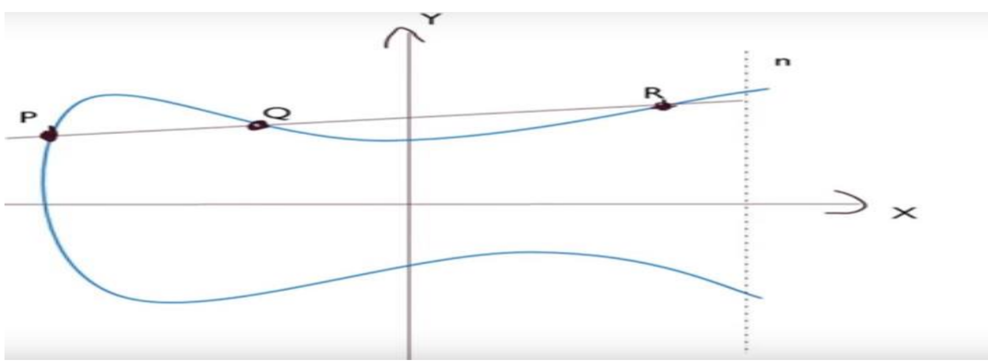


Figure 3.2: Elliptic curve observation

Let us say that the above elliptic curve is represented with $E_p(a,b)$.Also consider the equation

Q=KP where Q, P belongs to $E_p(a,b)$

And the term K is going to be less than our limit that we have set in our elliptic curve i.e. $k<n$. From above we can conclude that this function is a one way function or we can say a trap door function because in this function it is easy to find the value of the Q if the values of K and P are given but if we go in the opposite direction then it will be difficult for us to find the value of K given Q and P, this kind of problem is known as the discrete logarithm problem.
Now we are going to discuss about the ECC key exchange method, ECC encryption and the ECC decryption.

So, firstly we will start with the key exchange method let us consider an elliptic curve $E_q(a,b)$ where q, a and b are the parameters for the elliptic curve where q should always be in the form of 2^mq is a integer or a prime number.

G would be that point of the elliptic curve that can generate all the points of the elliptic curve and this point is known as the generator point of the elliptic curve. Let us consider two persons A and B both to them are going to exchange the keys.

For Person A:
Let us say private key for this person is n_A and this can be calculated by $n_A<n$ where n is our limit. And the public key P_A is calculated as:

$P_A = n_A * G$ where G is the generator point

In the similar way the private and the public key for person B can be calculated:
Let us say the private key of person B is n_B and public key is P_B.
Here $n_B<n$i.e less than our limit and $P_B = n_B * G$
Here public key is that key that is shared with another person. The secret key calculation for both person A and B can be done as follows:
Since both the persons have their private key when their own private key is multiplied with another person's private key then the secret key is calculated.

For person A:

Secret key $k=n_A*P_B$

For Person B:

Secret key $k=n_B *P_A$

Now we will discuss the ECC encryption and decryption:

For this purpose, the message is not encrypted but the message is encoded into a point on the elliptic curve and after that this point on the elliptic curve is encrypted and sent to the other end. Let us consider the message that we want to send is M first step in the algorithm is to encode this message M into a point on elliptic curve let us say that point is P_M on elliptic curve then in the next step P_M is encrypted then the corresponding cipher point which is sent to the receiver is c_m.

$c_{m=}(KG,P_{M+}KP_B)$ where K is an random positive integer that is chosen for the encryption. When this encrypted point is sent to the receiver the decryption takes place, they're by multiplying the first point in the pair with the receiving end's secret key

i.e $KG*n_B$

after that this value is subtracted from the second point i.e

$P_M + kP_B – (kG \text{ x } n_B)= P_M +kP_B – (kP_B) =Pm$

which is the original point. After this the receiver is going to decode the message from this point.

3.2 System design for scheduling

Our main aim is to allocate resource in the fog layer in such a way that the efficient use of the resources in the fog layer is possible. Three-layer model is going to use in our system in which three layers are present in our environment that is cloud layer middle one is the fog layer and the client layer or end user's layer. The main problem is that in our fog environment there are many clients are present and each client can make several requests according to their need but the resources are present in the limited amount, for dealing with such kind of problem server virtualization is an important concept. In server virtualization we break the actual server into many different virtual servers or we can say that a particular resource is going to be divided into many parts each part can deal or handle an individual request from the end user client. In this way we can maximize the server resources. The following figure can show the server virtualization concept on cloud and fog layer.

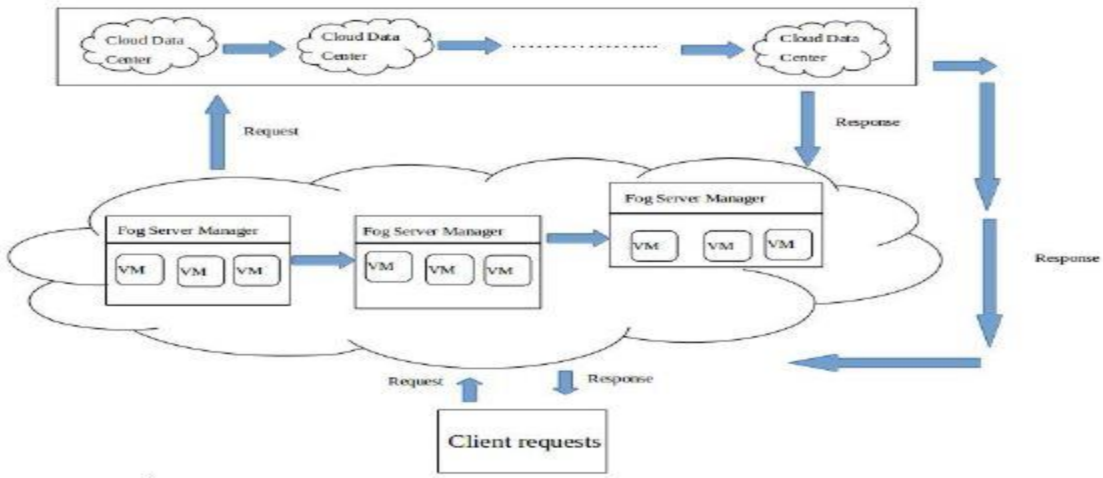


Figure 3.3: Server Virtualization in cloud and fog layer environment

With the help of the server virtualization the overall cost system usage and response time of the system is going to decrease. The algorithm that we are going to implement is on the fog layer that is the middle layer between client layer and cloud layer.

The bottom layer is the client layer or the end user's layer in which different IOT devices are present these devices can make any type of request to the server. First of all the request made by the user goes to the fog layer where they are assigned their priorities and after that resources are checked whether the resources are present in fog layer to satisfy the client's request if the resources are available then the client's request is served in the fog layer otherwise if all the resources in the fog layer are exhausted then the request is diverted to the cloud layer. Our main motive to serve the requests in fog layer is achieve the low latency. The system model is described as follows:

As shown in fig.3.3 there are three layers are present in our cloud-fog environment.

- First layer is the cloud layer that is the top most layer of the fog computing environment in this layer we have many cloud data centers.
- Second layer is the fog layer that is the middle layer of the fog computing environment, in this layer various fog nodes are present or we can say that various fog devices are present or the Fog servers are present (FS). The fog servers present in the fog layer have various micro data centers. For managing the various resources such as processor, storage present in the fog layer fog server manager (FSM) is present.
- The end layer is known as the client layer or the end devices layer. In the client's devices are present which makes the requests to the fog layer for various kind of services according to their needs.

The fog layer receives the requests from the client layer. It depends on the fog nodes to accept the request or they will reject the request. Depending on the various factors if the request is completed in the given parameters or we can say the given deadlines then the fog layer devices are going to accept the request otherwise the fog nodes are going to decline the clients request. If the client's request is satisfying the deadline parameters then fog nodes accept the request and fog server manager is going to put that request into the priority queue and server the request according to their assigned priority levels. On the basis of the priority of requests the fog server manager servers the request and for serving the requests first of all it checks the availability of the resources present in the fog layer. If all the resources of the particular fog node are free or are available for serving then the request is completed and the results are sent back to the client. But sometimes there are chances that the resources of the fog nodes are not available that is the resources are acquired by the other requests then that fog node can communicate with other fog nodes for serving the clients requests but if all the resources of the fog layer are busy, not able to serve user's requests the requests are redirected to the cloud.

3.3How queuing and priority works

In order to serve the client requests certain deadlines, have to be met. So that the task is completed in time and in efficient manner [22].

Some of the parameters that we are going to use in our algorithm are discussed as follows:

- Q_H, Q_M, Q_L High, medium, low priority queues
- T_1, T_2 Threshold for deadline at levels 1 and 2
- req_i the i^{th} user request
- μ_i total service time of req_i
- $delay_i^T$ maximum allowed delay of req_i
- W_i^Q total time user req_i spends in the queue
- C total number of servers
- $ST_{est.i}$ estimated service time required by req_i
- ST_{est}^T estimated service time required by all tasks

As mentioned in the above parameters we have three priority levels 1 means high 2 means medium and 3 means low.

From the client layer the client makes a request and the total time the client's request is going to spend in the fog layer is equal to the sum of the time that the request takes in the queue and the time that the request will take to complete on the fog nodes that is the request total service time.
i.e
$W_i = W_i^Q + \mu i$

The delay that the request can handle is specified in the service level agreement.
But our main aim to achieve the Quality of service (QOS) for this purpose the time that the request of the client spends in the fog layer should be less than the maximum allowed delay for the request that is specified in the service level agreement (SLA).
i.e
$W_i < delay_i^T$

3.4 How the priority model is going to work

The priority algorithm is going to work in the following three points.

if $delay_i^T == ST_{est,i}$
Return (H)

First of all, the maximum delay that is allowed to a task is compared with the total estimated service time of the request if both of them are found equal then the request is assigned a higher priority.

In the algorithm we are going to use the two variables T1 and T2, these two requests are used to order the requests in the system according to their timeline.

else if T_1< delay $_i^T$<= T_2
if SB_{CAT} ==1
Return (H)
else if SB_{CAT} ==2
Return (M)
else if SB_{CAT}==3
Return(L)

If the first condition is not satisfied in the algorithm then the algorithm is going to jump to the next condition i.e stated above .In this condition we will check that if the total delay of the request falls between the parameters $T_1$and T_2 then the priority of the request is given according to the original priority of the request. If the original priority is 1 then the assigned priority level is going to be H (high priority), elseif the original priority is 2 then the assigned priority level will be medium (M), and elseif the original priority level is 3 then the assigned priority level would be low(L).

The last condition is given by:

Else delay $_i^T$ > T_2
if SB_{CAT} ==1
if Q_H is not full
Return (H)
else
Return (M)
else if SB_{CAT} ==2
If Q_M is not full
Return (M)
else
Return (L)
else if SB_{CAT} ==3
Return (L)

If the above stated two conditions are not satisfied then the last condition is checked in which the maximum allowed delay of the task is compared with the variable T_2 if this condition is fulfilled then the priority level is assigned according to the original priority if the corresponding queue that is to be assigned is not filled otherwise one lower priority queue is assigned. If the original priority is 1 and the Q_H queue is not full then assigned priority level is H but if the queue is full then M assigned as the priority queue. Similarly, if the original priority is 2 and Q_M is not full then assigned priority level will be M otherwise if the queue is full then lower priority level is assigned i.e (L) low is assigned to the user's request and if original level is already 3 then the assigned priority level will be L.

Now we are going to discuss the algorithm that we have used in our implementation of our project. The implemented algorithm is going to work in two steps:

Step1:

For each req_i
1. req_i is sent to nearest FS.
2. FS calculates delay $_i^T$
3. if ($((ST_{est}^T/C) + ST_{est,i}) > delay_i^T$),
Reject req_i
4. else
(Pri) = Priority ($delay_i^T$, ST_{est}^T)
if (Pri== H)
Place req_i in Q_H
else if (Pri == M)
Place req_i in Q_M
else if (Pri == L)
    Place req_i in Q_L

The user in the client layer makes request and that request is sent to the fog layer. In the first step we will assign that request to a particular fog server that is nearest to that client and after that the fog server (FS) will calculate the delay. There is a deadline associated with each client's request if the resources are available to meet the maximum allowed delay then only the request is accepted other the request is going to be rejected. If the request is accepted then the priority queue is assigned to the request according to priority module that we have discussed in the chapter 3.

Step2:

1. for x = H, M, and L,
2. while (Qx ≠ NULL)
3. for each request in Qx
4. if sufficient resources in current FS
Serve the request in the current FS
5. else if sufficient resources in fog layer
Allocate resources from the remaining FS, following
Step A, divide task to sub-tasks if necessary.
6. else
Send the request to the cloud.

In the second step after the requests are assigned a particular priority queue, we will take out the requests one by one from each priority queue until each queue will be null. For a particular request that is taken out from the queue from a fog server, if that fog server has sufficient resources to serve that request then that request is served in that fog server only. But if the fog server is unable to server that request because of inefficiency of the resources then that request will be sent to another fog server and the step1 is followed for that fog server. Some if the task is large then we have to divide that task into small parts and assign that parts of the task to other fog servers. Finally, if the request or the task from the client can’t be served from any fog server then the request is sent to the cloud.

CHAPTER-4

PERFORMANCE ANALYSIS

For the implementation of our algorithm we are going to use the ifogsim. The ifogsim is a Simulation toolkit that is going to simulate the fog environment for us. The software requirements for our implementation are jdk 1.8 and above, net beans 8.2 and above and ifogsim.

The steps for our Implementation are as follows:

Step 1: We create fog computing environment with cloud datacenters, fog nodes and number of client's devices (Cloud Customer).

Step 2: we deploy each cloud data centers, fog nodes and Cloud Customer in different regions, regions are represented in world map.

Step 3: Next the cloud customers gives the request for the job to fog nodes or cloud datacenters.

Step 4: we implement the priority-based algorithm for Scheduling task and load balancing policy for balancing the load in the cloud server.

Step 5: Run the simulation

Step 6: We Plot the graph results are

1) Response time for Number of Tasks
2) Scheduling time/length for Number of Tasks (Seconds)
3) Load balancing rate
4) Delay
5) Energy Consumption (KJ)

The simulation duration we have set for 60 minutes. The requests per user per hour is set to 60,data size per request is set to 100 bytes, peak hour start is 3 GMT and peak hour end is 9 GMT, average peak users are 1000, average peak off users are 100. Every data Centre configuration goes like this arch X86, operating system Linux, cost per VM per hour is $0.1,memory cost /s is $0.05,storage cost /s is $0.1 ,data transfer cost per GB $0.1 and the physical hardware units is 1. Number of virtual machines per data center are 5.
First of all, we are going to create the fog computing environment with cloud data centers, fog nodes and number of client devices or we can say the cloud customer. In simple words firstly we are going to create topology for the fog environment. When we run are project following window will display where we will create our topology.

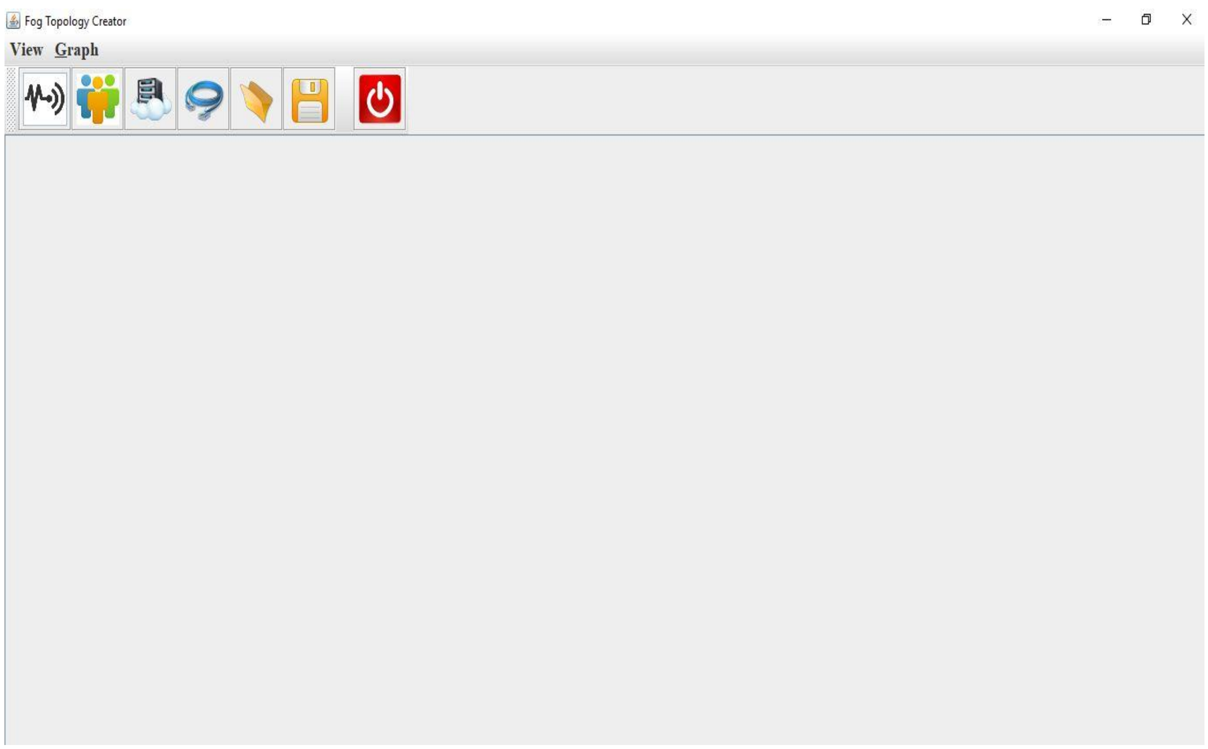


Figure4.1: window for creating the fog environment (topology)

We are going to analysis our results in two ways by putting the users and fog data centers in one region i.e. all the users and fog devices are present in one region and the results are compared for same topology but with the different configurations, with priority scheduling and without priority scheduling. So, firstly start by putting all the users and fog nodes in one region.

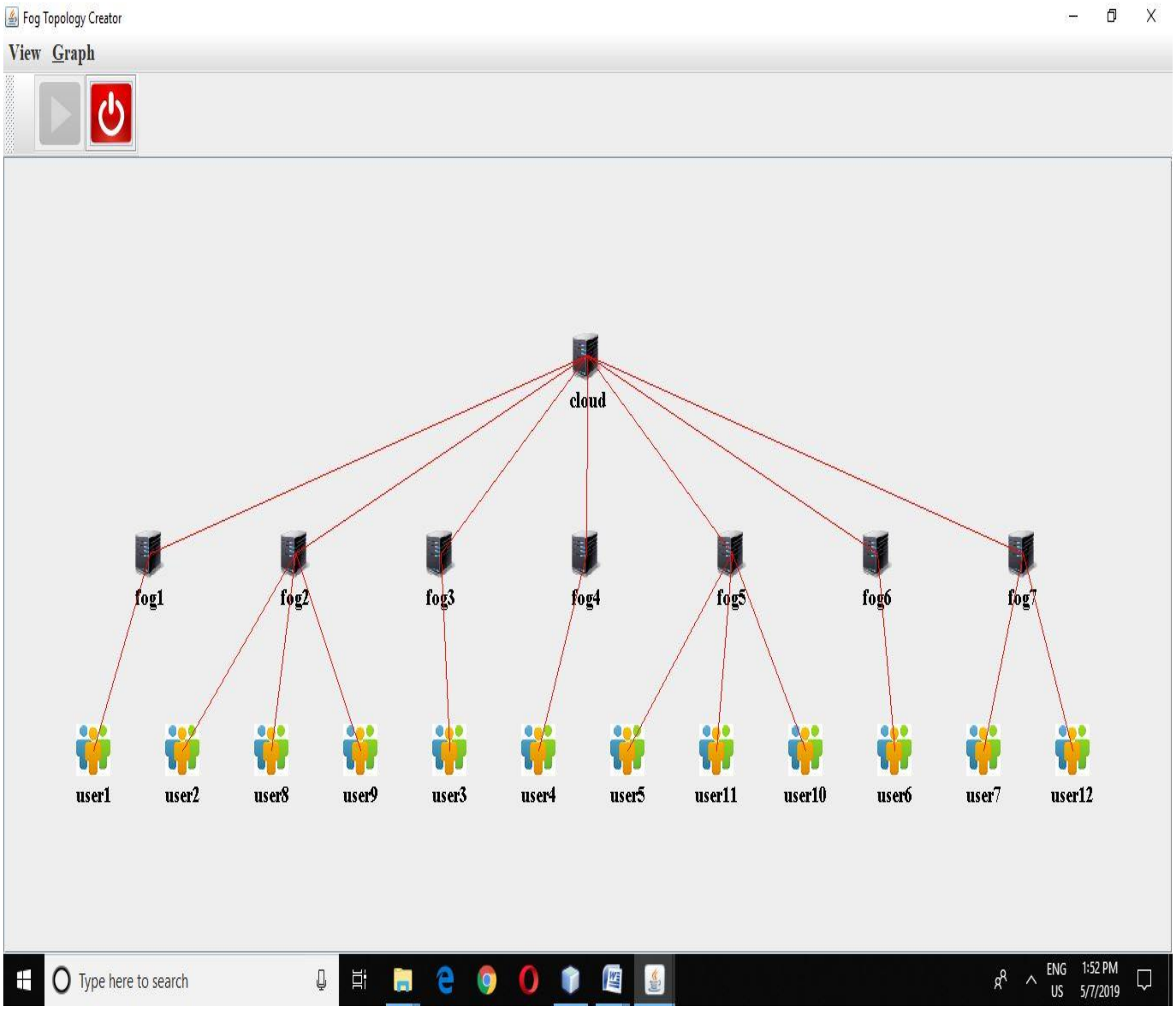


Figure4.2: Topology with 7 fog nodes and 12 users

In the above topology we have 7 fog nodes and 12 users. After making the topology click on the execution button.

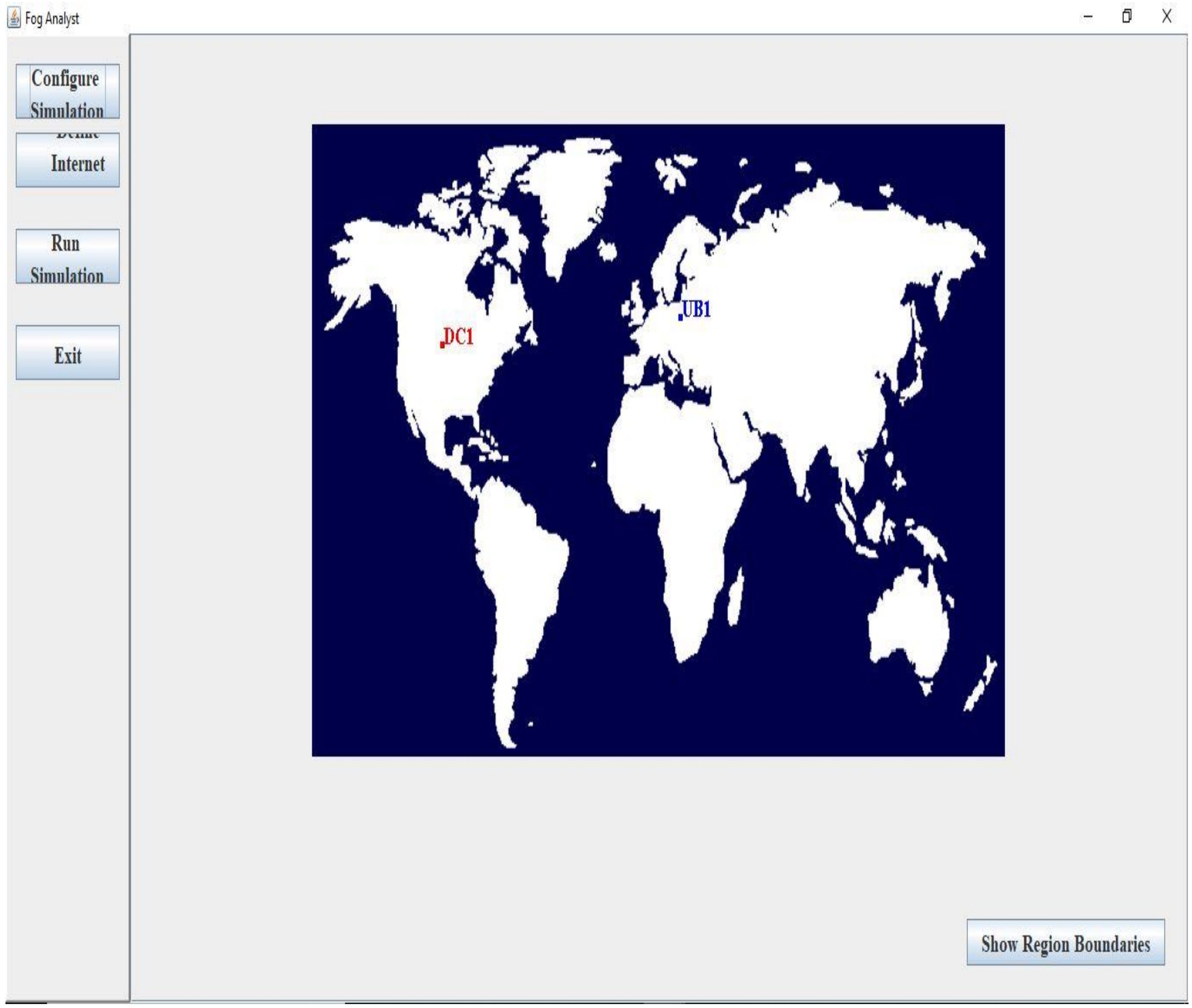


Figure4.3: Initial screen after the topology making window

Initially as you can see on this screen that there is only one data Centre and one user base After this we have to click on the configure simulation button that is shown in the upper left corner of the screen to place the users and fog nodes in different regions and after that you will get three tabs main tab where you can configure the users settings, next tab is data center configuration tab where you can set the data centers configuration and last tab is the advanced tab where you can select whether to run the application with priority or without priority scheduling method. As shown in the following figures:

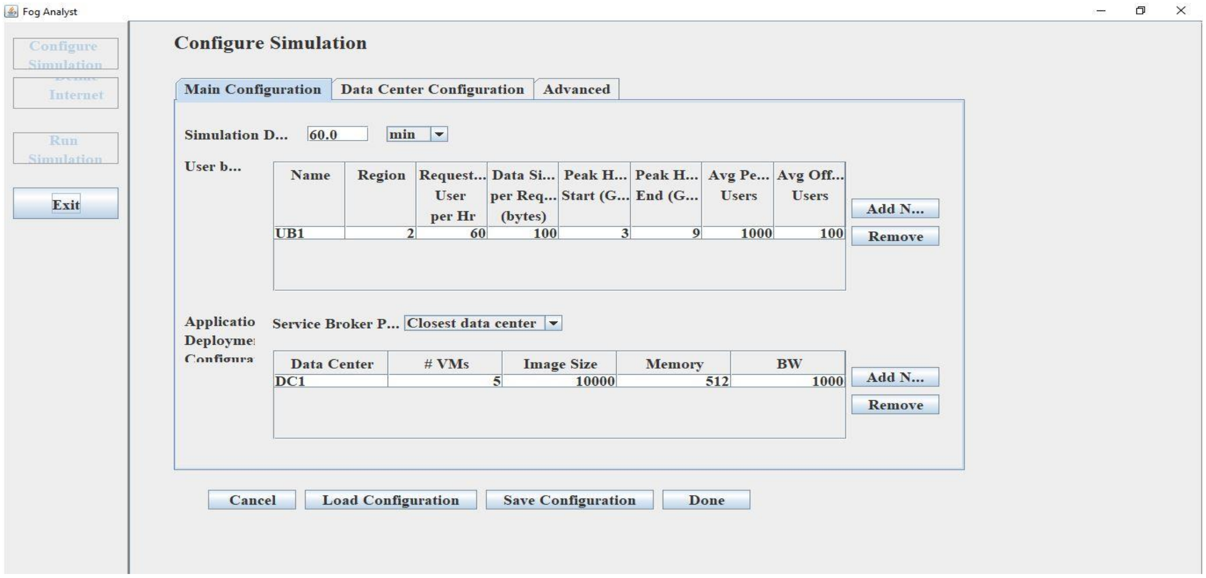


Figure4.4: Main configuration window

In this window you can set the simulation time that the time up to which our simulation is going to run, requests that the users can send per hour, data size per request, peak hours start (GMT), peak hours end (GMT), average peak users, average off peak users and can also add and remove the user bases. And can also set the application deployment configuration i.e. the service broker type either closest data center, optimize response time, fuzzy etc.

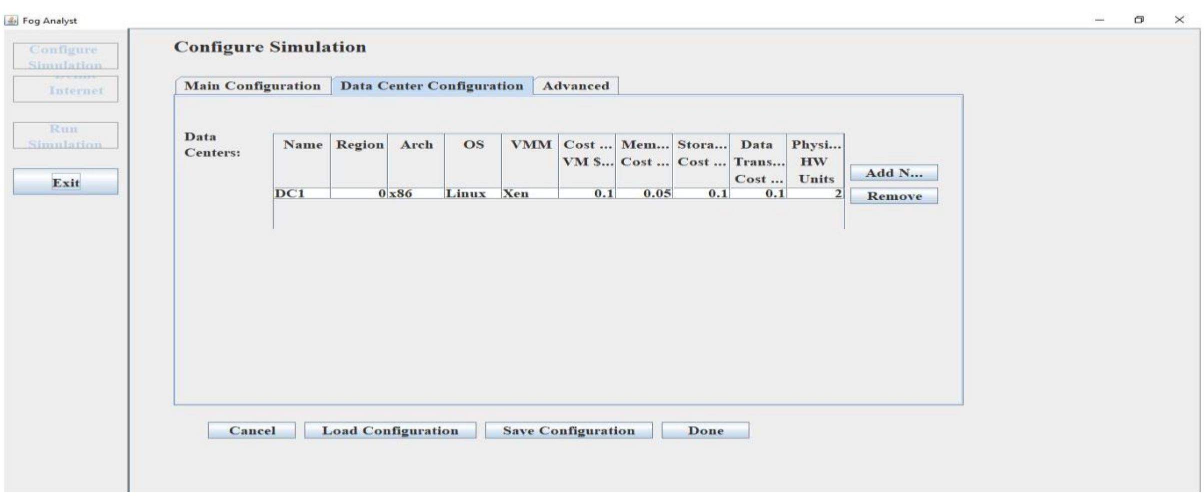


Figure 4.5: Data Centre configuration window

Using the data Centre configuration window, you can set the data centers region, add or remove the data centers, set the Arch, VMM, cost of the VM, memory cost, storage cost, data transfer cost and the physical hardware units.

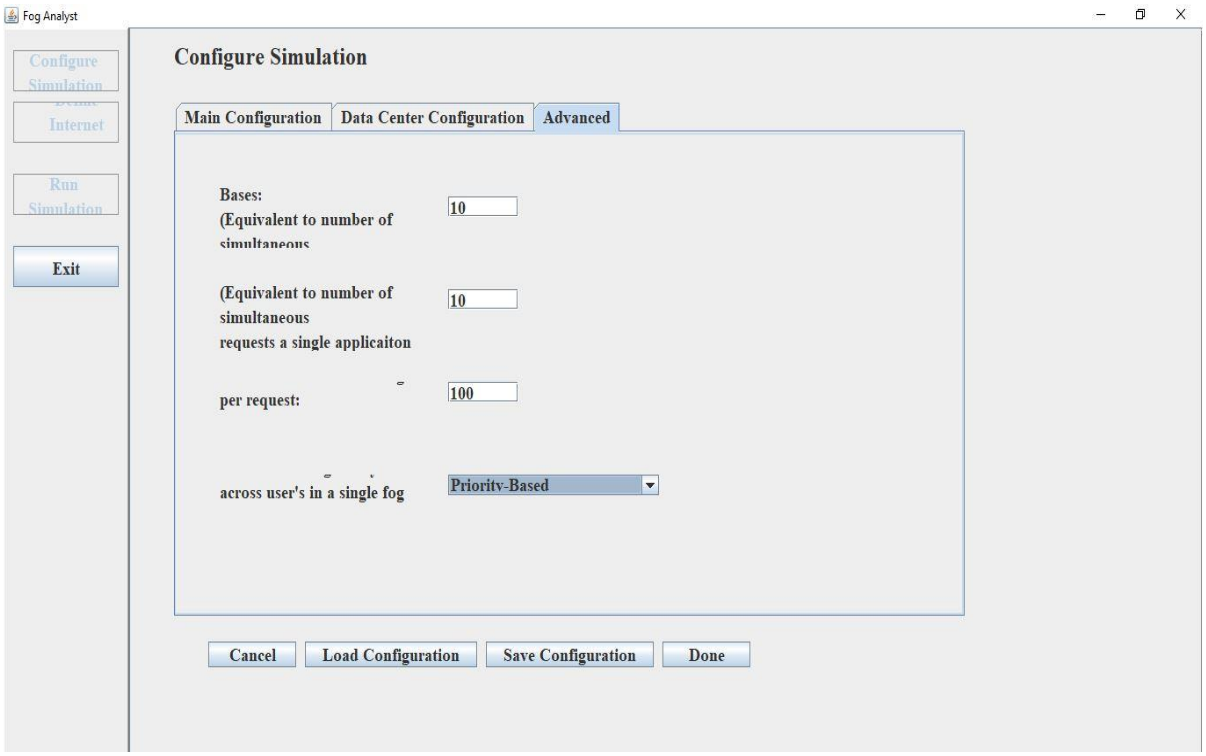


Figure 4.6: Advanced window of the configure simulation

Using this window, you can select either priority scheduling, round robin or active monitoring scheduling algorithm. You can also save your topology that you have made using the save button in the topology window and can load that topology anytime, in a similar way you can save your configuration for user’s base and data centers and can load whenever you want. After this you can click on the done button after that you can also configure the internet configuration that is you can set the transmission delay between the two regions and also set the bandwidth between the two regions in mbps in other words you can set the delay matrix and the bandwidth matrix as shown in the next figure. After this you can click on the run simulation button to run the simulation fog nodes and users will get displayed on the different regions of the map and the graphs get plotted.

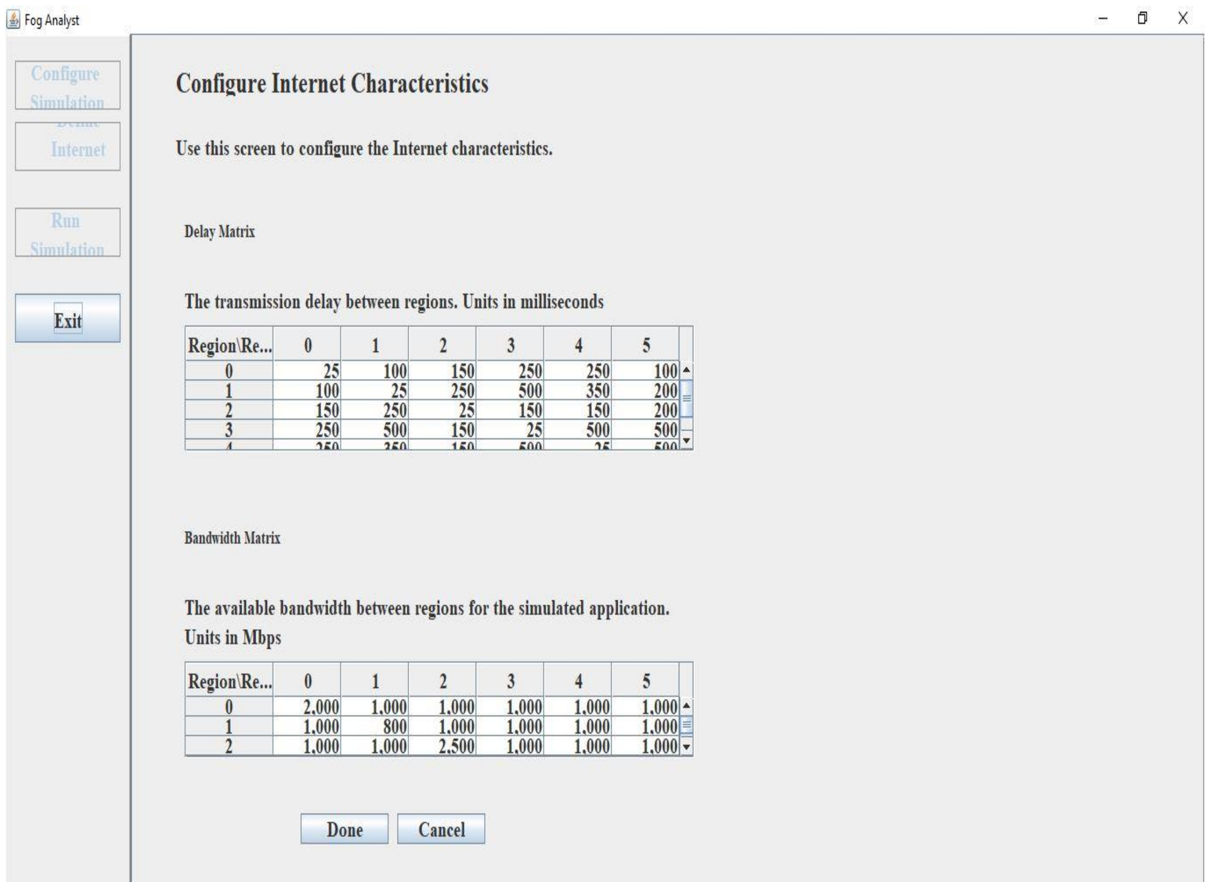


Figure4.7: Configure internet characteristics window

Let us begin our analysis by putting all the data centers and user into a single region and run the simulation for same topology with and without priority scheduling and total time for encryption and decryption that the application will take will also get shown in the results window. Firstly, we will run the application with priority scheduling and after that with round robin that is without priority and compare the results for both the cases.

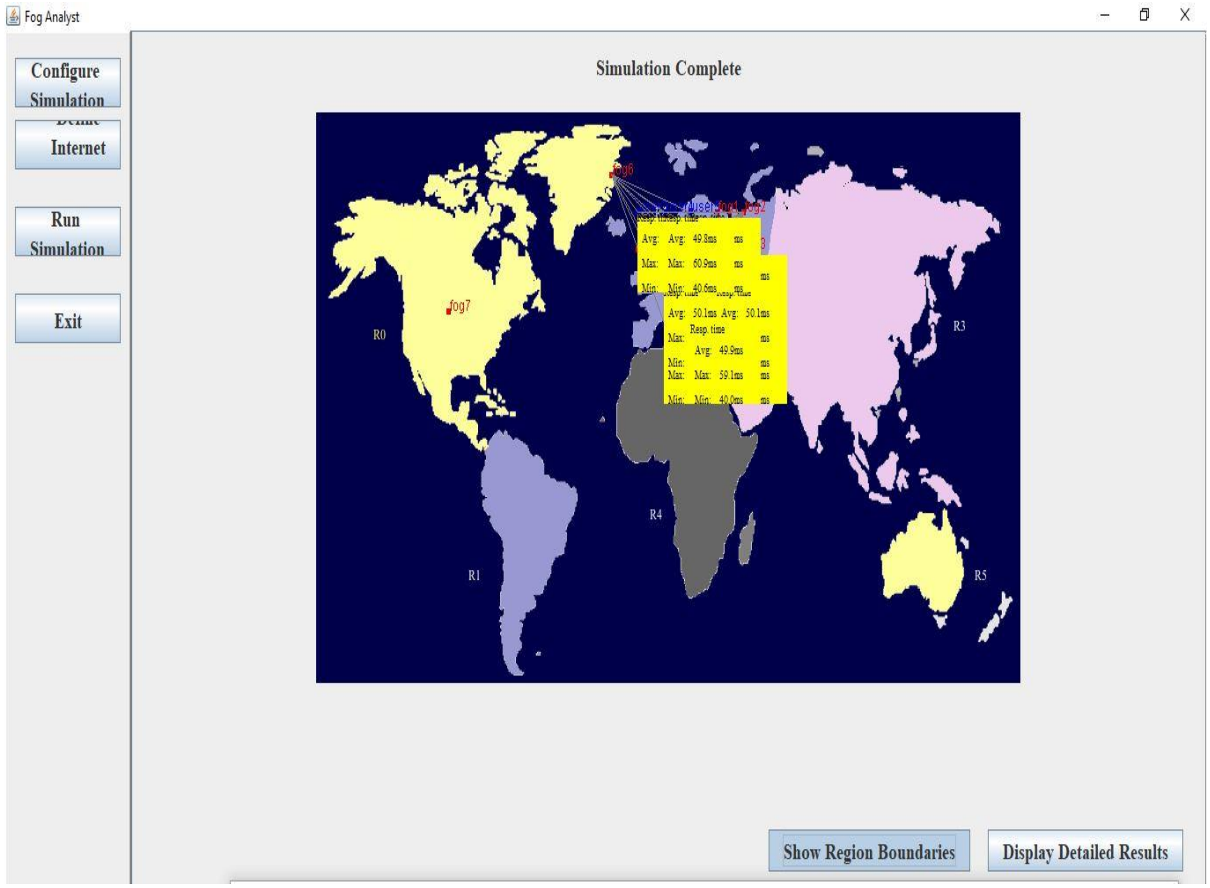


Figure 4.8: Region Plot for topology with 7 fog devices and 12 users

In the above plot you can see that all the users and fog devices are present in the region 2 There are total 6 regions but for our first case we have chosen region 2 to put our users and fog devices since all the fog devices should have to be present in the same region

And that you can view the simulation results in the simulation window the overall budget and summary is present in that window the overall cost and the encryption or decryption time is also shown in that window you can view the graph for each user and fog device. Use the export results window to save the results of the results window in a PDF file. After viewing the results click on the exit button to stop the simulation.

Simulation Results

Overall Budget and Time Summary

	Schedule Average (ms)	Schedule Minimum (ms)	Schedule Maximum (ms)
Overall Response Time:	50.04	37.63	62.88
Data Center Processing Time:	0.50	0.00	0.92

Export Results

Figure 4.9: overall Budget and time summary

In the above figure you can see the overall budget and time summary of the simulation for the given topology.

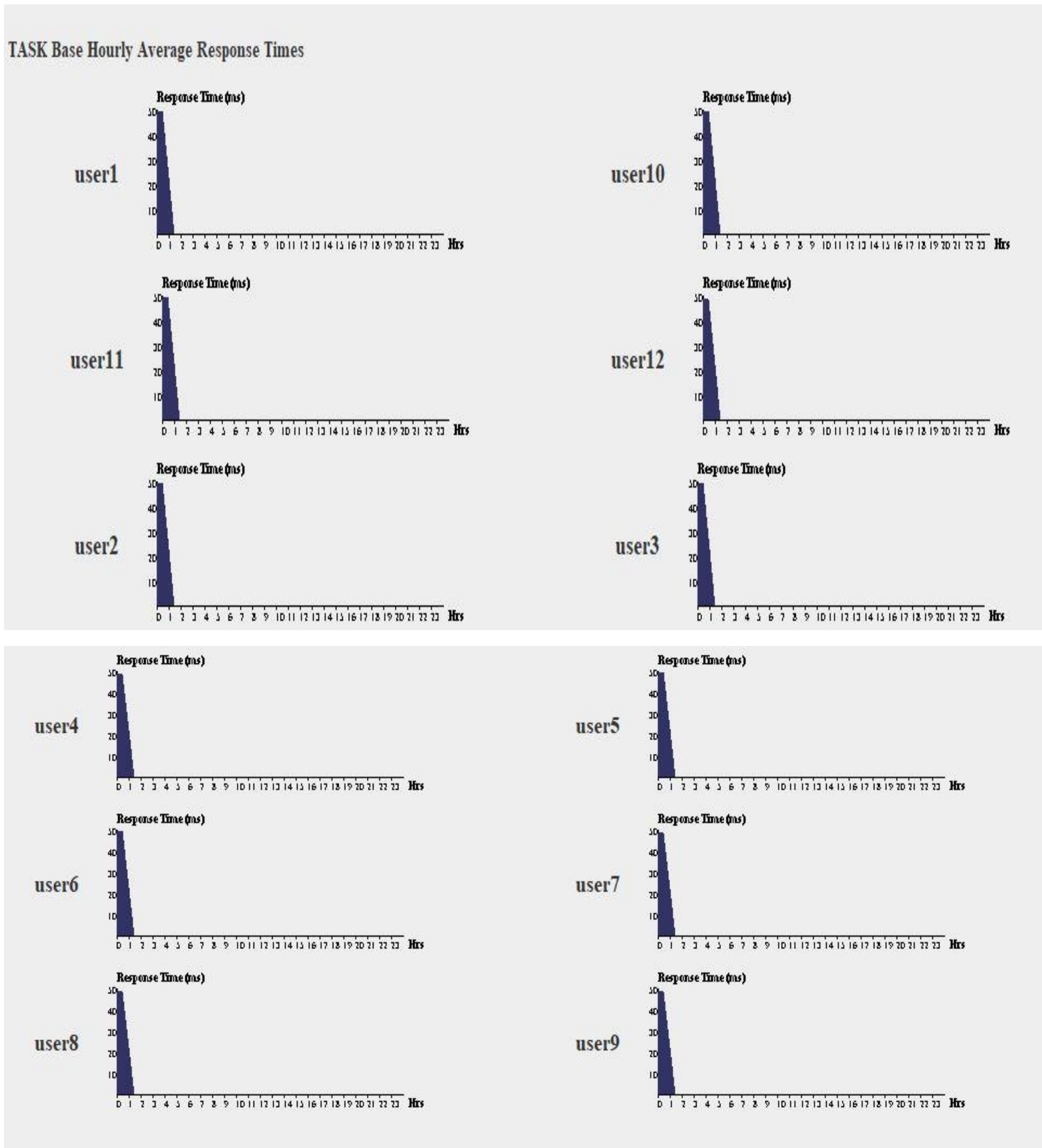
Overall response time is 50.04 ms, response time is the time that is elapsed between the enquiry enters the system and the time that is taken by system to respond that enquiry.
If we have two systems A and B system A has response time greater than system B then the system B is considered to be better than the system A, response time should be as low as possible. Above figure is also showing the time taken by the data center to do the processing and for our topology it is coming out to be 0.50.

RESPONSE TIME FOR NO OF TASK

Tasks	Avg (ms)	Min (ms)	Max (ms)
user10	50.125	38.14	60.882
user11	50.141	39.63	62.133
user12	49.9	39.879	60.634
user1	50.095	40.631	60.129
user2	50.219	40.132	61.631
user3	50.153	40.881	62.884
user4	49.803	40.631	60.885
user5	50.253	37.63	59.881
user6	50.082	40.544	61.379
user7	49.815	39.88	59.131
user8	49.906	40.032	59.135
user9	49.994	40.88	61.888

Figure 4.10: Response time for no of task

In the above figure you can see the max and min response time for each user and also the average response time for each single user is shown in the above figure.



Graph4.1: Task base average response time

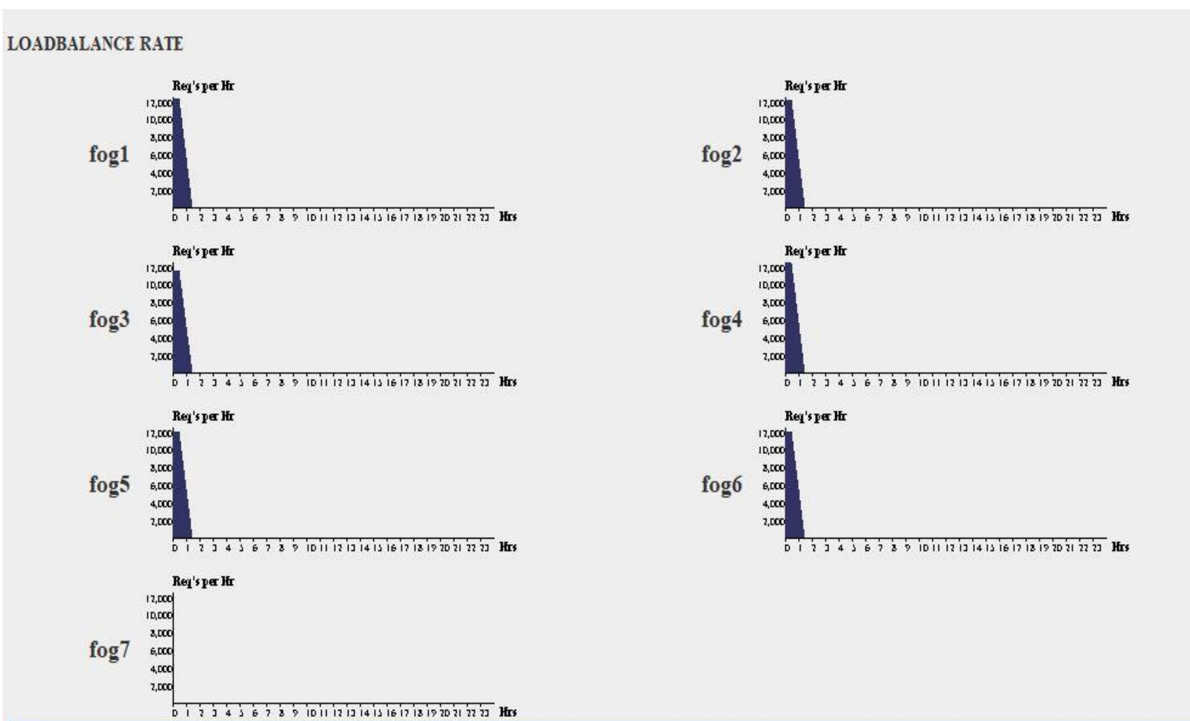
In the above figure you can see the task base average response time for 12 users. In the x-axis we have the hours and in the y-axis, we have the response time in milliseconds.

SCHEDULING TIME

Data Center	Avg (ms)	Min (ms)	Max (ms)
fog1	0.501	0.028	0.91
fog2	0.503	0.016	0.905
fog3	0.505	0.016	0.912
fog4	0.499	0.015	0.92
fog5	0.486	0.016	0.92
fog6	0.498	0.017	0.908
fog7	0	0	0

Figure 4.11: scheduling Time for case 1

In the above figure you can see the scheduling time for each of the fog nodes max and min, their average scheduling time in ms is shown in the above figure. Each user is sending the request to the fog nodes and it is the duty of the fog nodes to schedule each of the requests in a better way so that efficient working of the application can take place.



Graph4.2: Load Balancing Rate for each Fog node in case1

Above figure is showing the load balancing rate for each fog node in x-axis hours are given and in the y-axis we have the request's per hour. Load balancing means to effectively balance the rate in the network and properly balance or distribute the traffic in the network.

Cost

Total Virtual Machine Cost :	**$3.40**
Total Data Transfer Cost :	**$0.77**
Grand Total :	**$4.17**

Security Measure

Encryption excution time (milliseconds) :	**617**
Decryption excution time (milliseconds) :	**11775**

Figure 4.12: Cost and security measure for case 1

Above figure is showing the total virtual machine cost, total data transfer cost and the grand total for the case 1. Since we have implemented the elliptic curve cryptography for our topology the encryption and decryption are taking place in that algorithm the time taken in encryption is 617 ms and the time taken in decryption is 11775 ms.

Data Center	VM Cost	Data Transfer Cost	Total
fog1	0.5	0.132	0.632
fog2	0.5	0.13	0.63
fog3	0.5	0.123	0.623
fog4	0.5	0.132	0.632
fog5	0.5	0.128	0.628
fog6	0.5	0.128	0.628
fog7	0.4	0	0.4

Figure4.13: Total Data center cost for case 1

The above figure is showing the results for VM cost, data transfer cost for each fog node and the sum of these cost is the total cost of the data center.

After this we are going to run the simulation without priority scheduling and compare the result with above simulation.

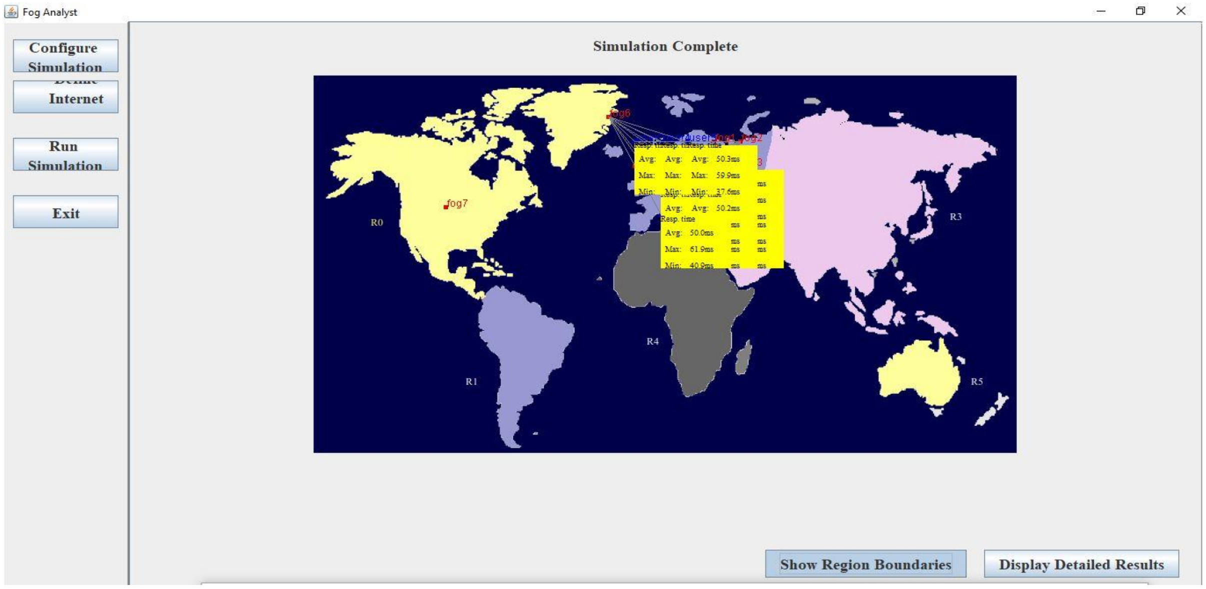


Figure 4.14: Region graph for case1 without priority

Overall Budget and Time Summary

	Schedule Average (ms)	Schedule Minimum (ms)	Schedule Maximum (ms)
Overall Response Time:	50.04	37.63	62.88
Data Center Processing Time:	0.50	0.00	0.92

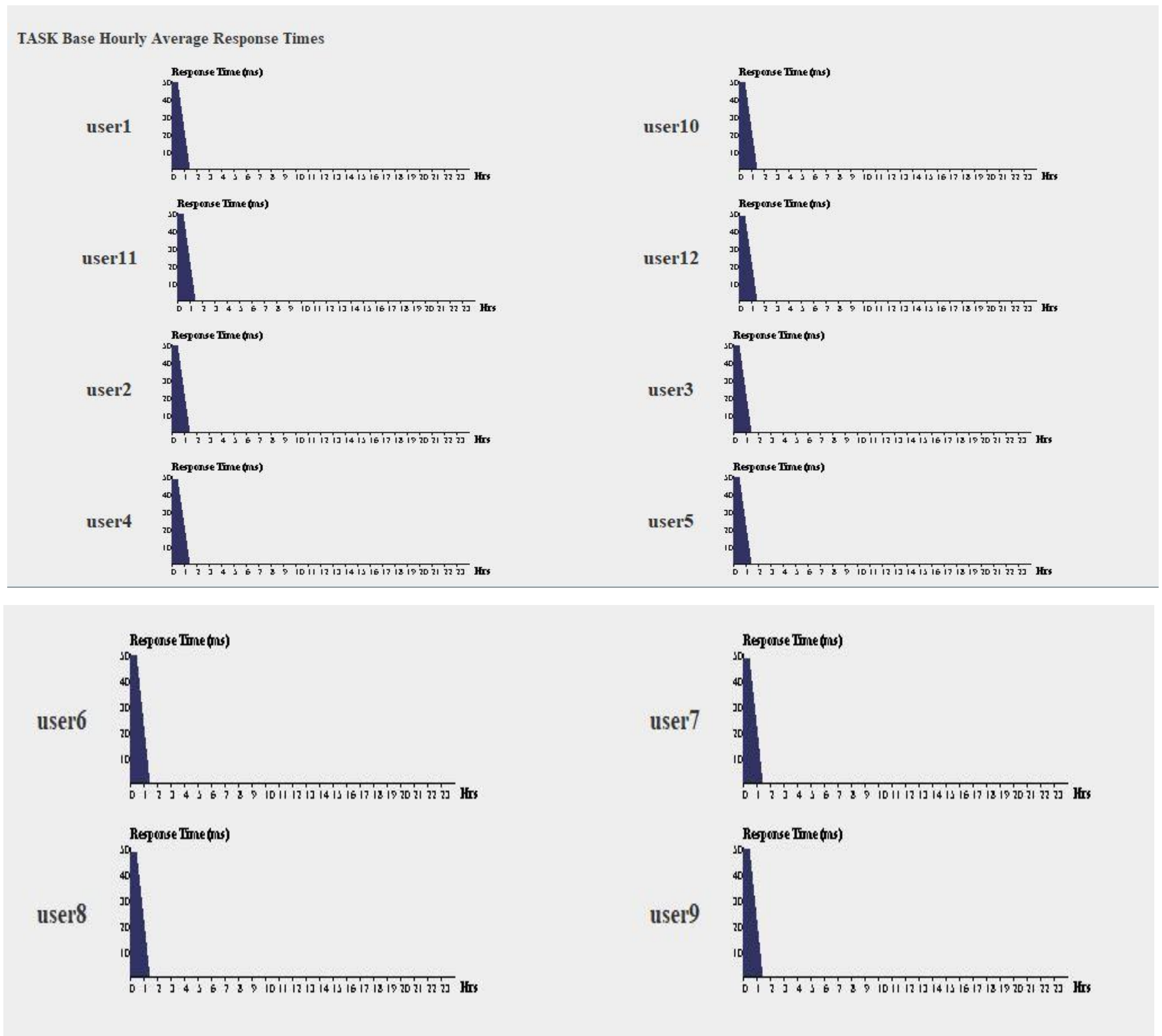
Export Results

RESPONSE TIME FOR NO OF TASK

Tasks	Avg (ms)	Min (ms)	Max (ms)
user10	50.181	40.141	60.882
user11	50.143	39.63	62.133
user12	49.914	39.879	60.634
user1	50.05	38.14	60.129
user2	50.223	40.132	61.631
user3	50.137	40.881	62.884
user4	49.814	40.631	60.133
user5	50.279	37.63	60.885
user6	50.065	40.544	61.379
user7	49.782	39.88	59.803
user8	49.885	40.032	58.885
user9	50.011	40.88	61.888

Figure 4.15: overall Budget and Time summary without priority

As you can see here that in the overall response time of the topology without priority not much difference is coming but you can see that the difference for a particular user is their, the difference is small but the difference is their but the end overall response time is almost same. But when you observe the difference in the cost for the system the cost in system with priority would be less than the cost in the system without priority. In the next figure you will see the task base hourly response for each user for the system without priority.



Graph4.3: Task base hourly average response for system without priority

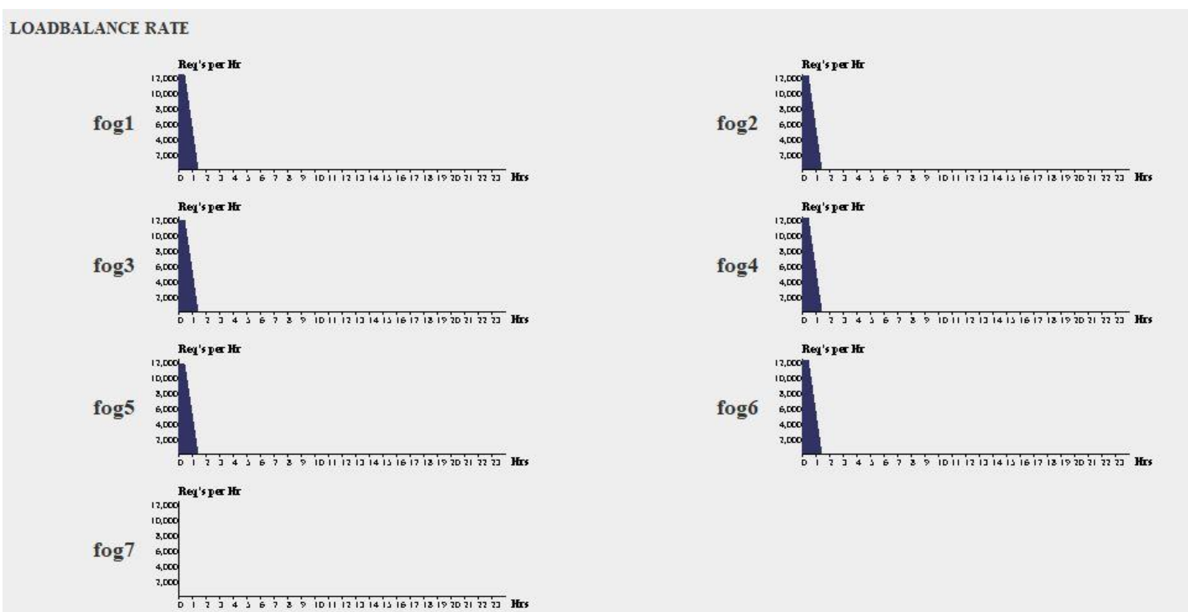
In the x-axis you can see the hours and in the y-axis is the response time in hours. There is a little bit of difference is coming for each user when compared with system with priority.

SCHEDULING TIME

Data Center	Avg (ms)	Min (ms)	Max (ms)
fog1	0.506	0.016	0.921
fog2	0.498	0.017	0.919
fog3	0.497	0.029	0.904
fog4	0.495	0.015	0.92
fog5	0.489	0.016	0.889
fog6	0.506	0.016	0.921
fog7	0	0	0

Figure 4.16: Scheduling time for system without priority

As you can note from the above result that there is some difference in scheduling time is coming for the fog nodes. The scheduling time is less in the system with priority than the system without priority the difference is small but the difference is still present their.



Graph4.4: Load balancing rate for the system without priority

Above figure shows the graphs for each fog nodes in the x-axis we have hours and in y-axis we have request's hour.

Cost

Total Virtual Machine Cost : **$3.40**

Total Data Transfer Cost : **$0.90**

Grand Total : **$4.30**

Security Measure

Encryption excution time (milliseconds) : **617**

Decryption excution time (milliseconds) : **11775**

Data Center	VM Cost	Data Transfer Cost	Total
fog1	0.5	0.131	0.631
fog2	0.5	0.13	0.63
fog3	0.5	0.127	0.627
fog4	0.5	0.13	0.63
fog5	0.5	0.124	0.624
fog6	0.5	0.261	0.761
fog7	0.4	0	0.4

Figure 4.17: cost and security measure for system without priority

From the above results you can see that the total cost in the system with priority scheduling is less as comparison to the system without priority. For each data center the total cost is little bit less as comparison to fog node in the system without priority. Hence from this we found that the total cost decreases when the priority scheduling algorithm is Applied in the system.

Now we are going to conduct the experiment for second case that is putting the fog nodes and the users in the different regions for the same topology with 7 fog nodes and 12users. Run the simulation with priority and without priority as we have done in the previous case and compare the results for both the cases whether the cost is decreasing for the system with priority or not, also compare the results for the system whether the overall response time is decreasing of not. The overall response and cost for the system with priority should come less than the system without priority. After running the simulation set the configuration settings as follows:

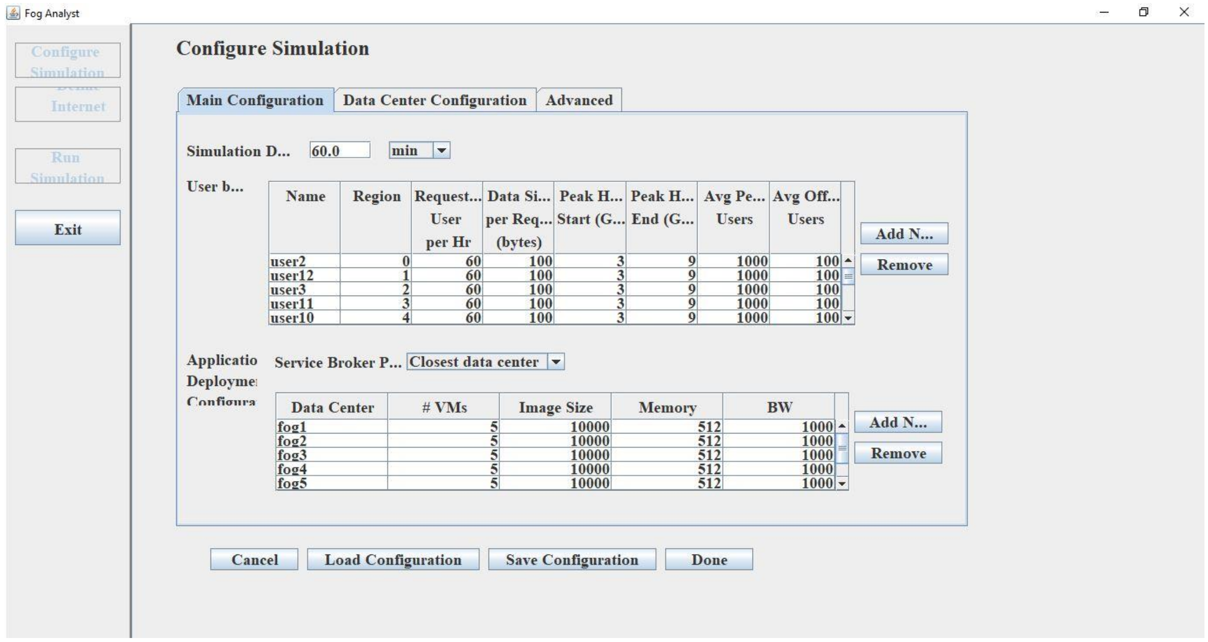


Figure 4.18: main configuration for case 2

As you can see in the above figure we have allotted the user bases in different regions.

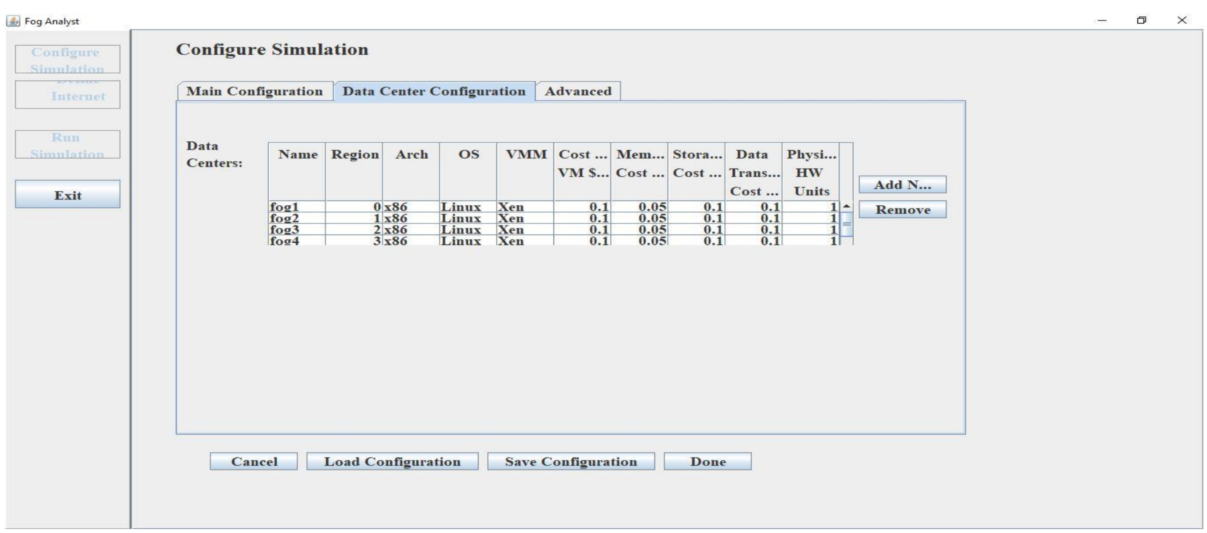


Figure 4.19: data center configuration for case 2

As shown in the above figure the fog devices are allocated in the different regions.

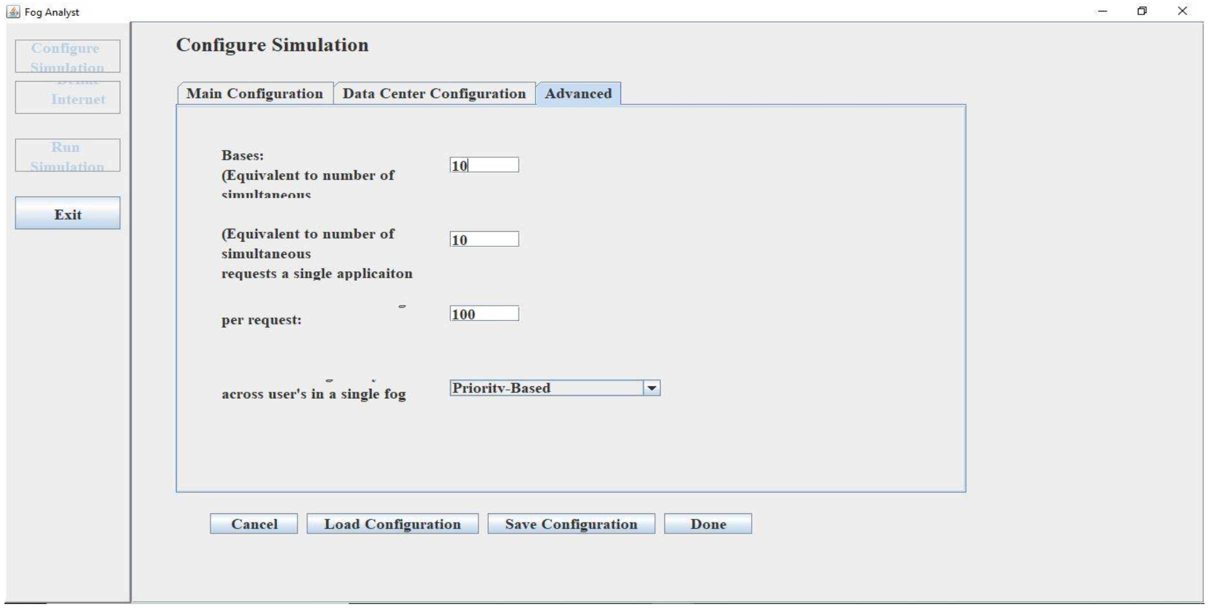


Figure 4.20: advanced window for case2

As shown in the above figure firstly we are going to run are application with priority scheduling so firstly the setting is for the priority based. Then after this run the simulation to see the results. For our second case let us take another topology.

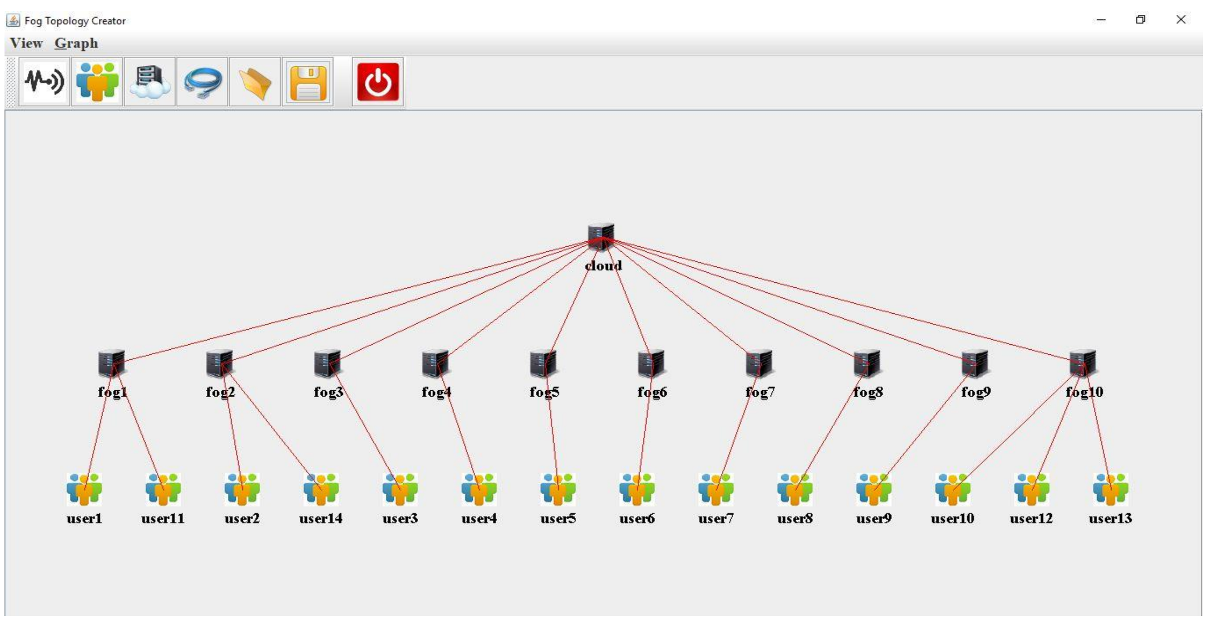


Figure 4.21: Topology for the second case

Above figure shows the topology with 10 fog devices and 14 users. Now we will place these fog devices and fog nodes in different regions.

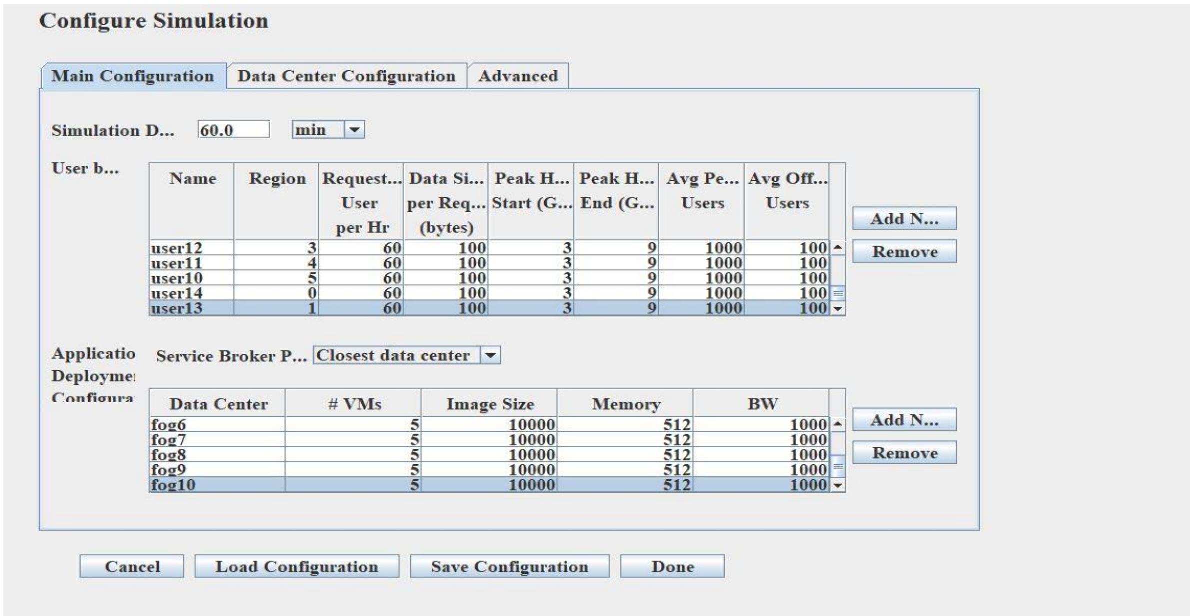


Figure 4.22: main configuration window for case2 with priority

Use the above window to allot the users in the different regions.

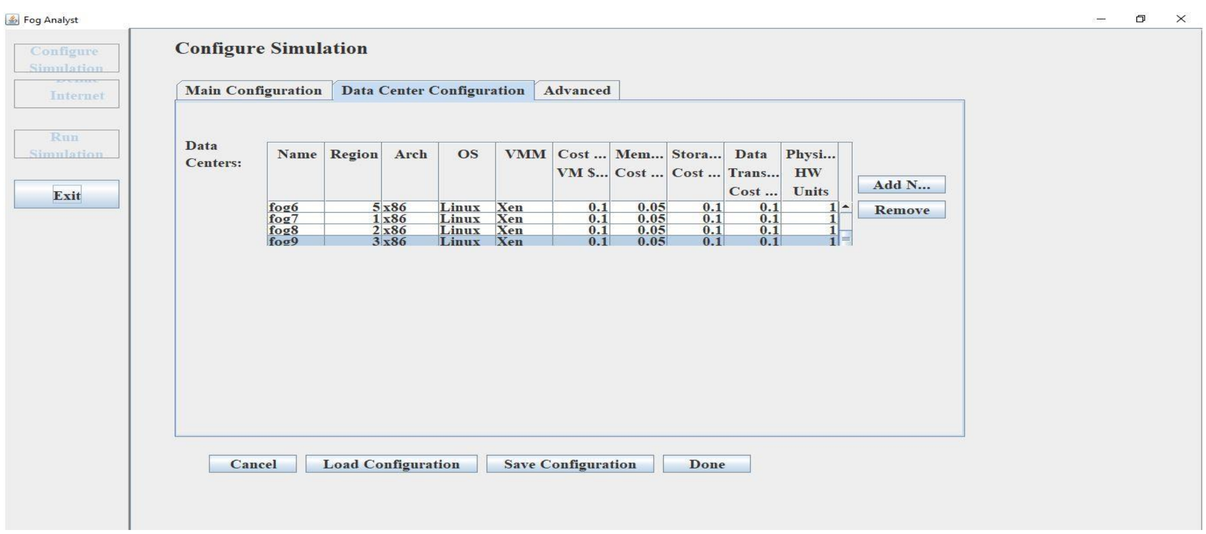


Figure 4.23: data center configuration window for case 2 with priority

Using the above window, you can set the data centers configuration as shown in above figure different regions are allotted to the data centers.

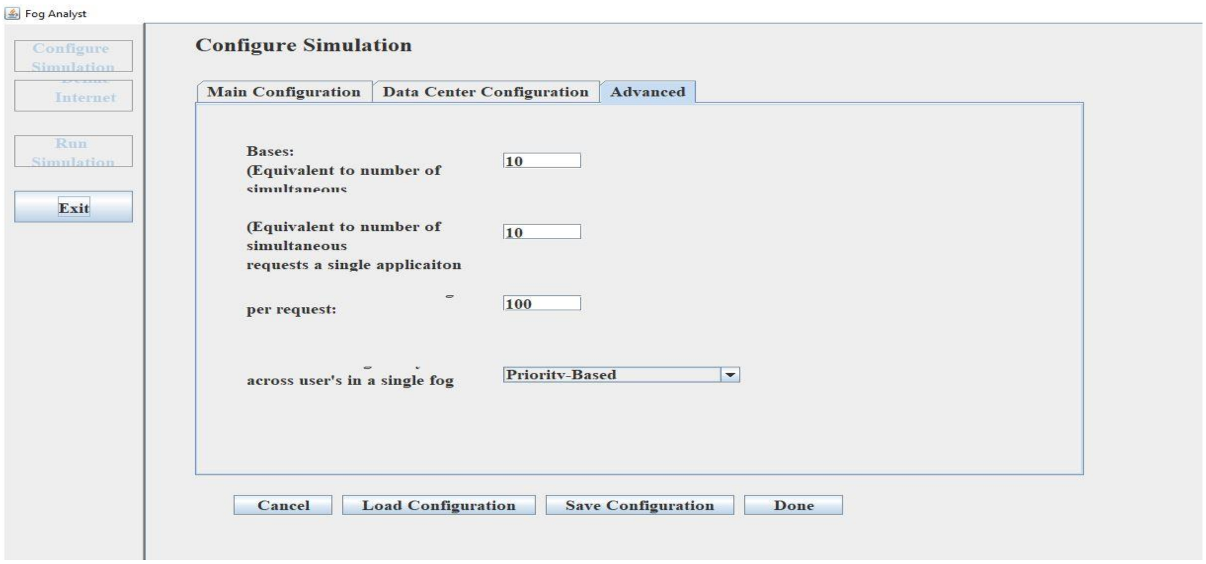


Figure 4.24: Advanced settings window for case 2 with priority

As shown in the above window that the setting is for the priority-based scheduling. After this run the simulation and analyze the results.

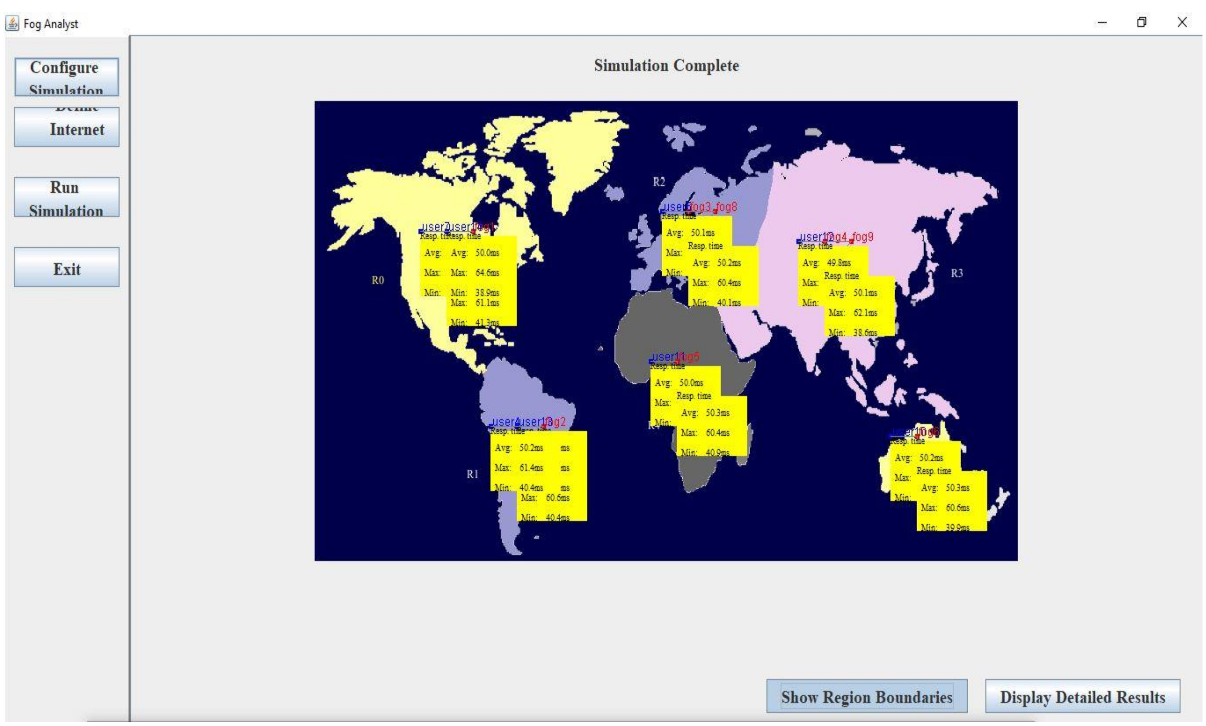


Figure 4.25: Region map for case 2

Simulation Results

Overall Budget and Time Summary

	Schedule Average (ms)	Schedule Minimum (ms)	Schedule Maximum (ms)
Overall Response Time:	50.14	38.63	64.64
Data Center Processing Time:	0.50	0.02	0.91

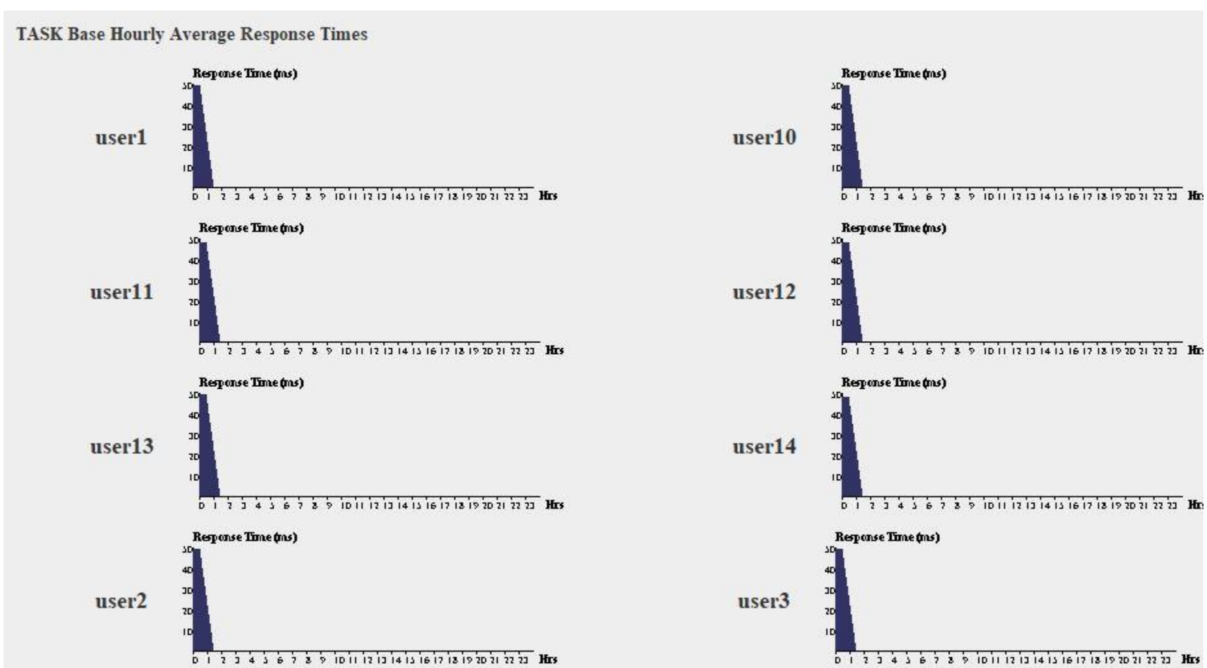
Export Results

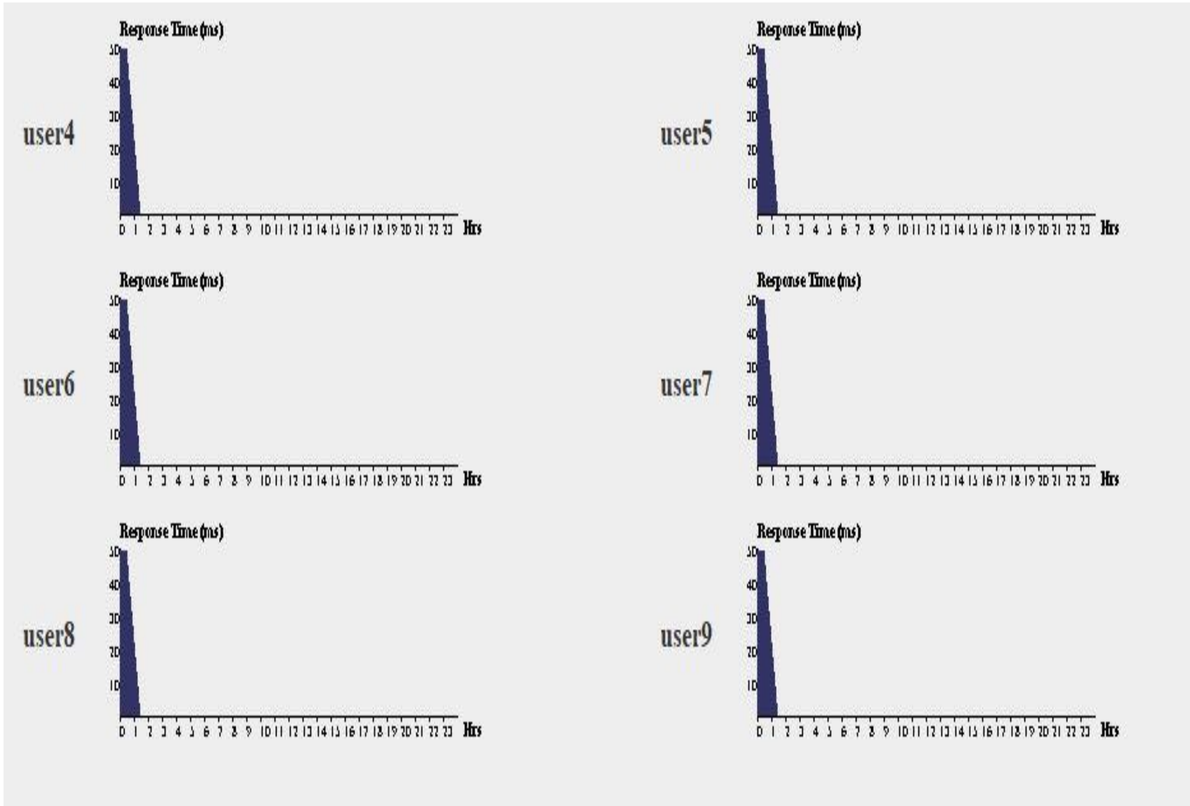
RESPONSE TIME FOR NO OF TASK

Tasks	Avg (ms)	Min (ms)	Max (ms)
user10	50.155	40.383	59.634
user11	49.964	39.158	61.404
user12	49.777	39.585	62.889
user13	50.256	39.649	61.889
user14	49.955	38.89	64.635
user1	50.179	40.129	60.384
user2	50.03	41.298	61.132
user3	50.271	40.389	60.643
user4	50.156	40.388	61.399
user5	50.132	38.63	59.631
user6	50.334	39.88	60.63
user7	50.302	40.633	60.884
user8	50.137	38.633	62.135
user9	50.282	40.902	60.421

Figure 4.26: Summary for case with priority

In the above figure you can see that the overall response time is 50.14 and the data center processing time is 0.50.And the response time for no of task is also shown in the above figure.





Graph4.5: task base hourly average response time for case 2 with priority

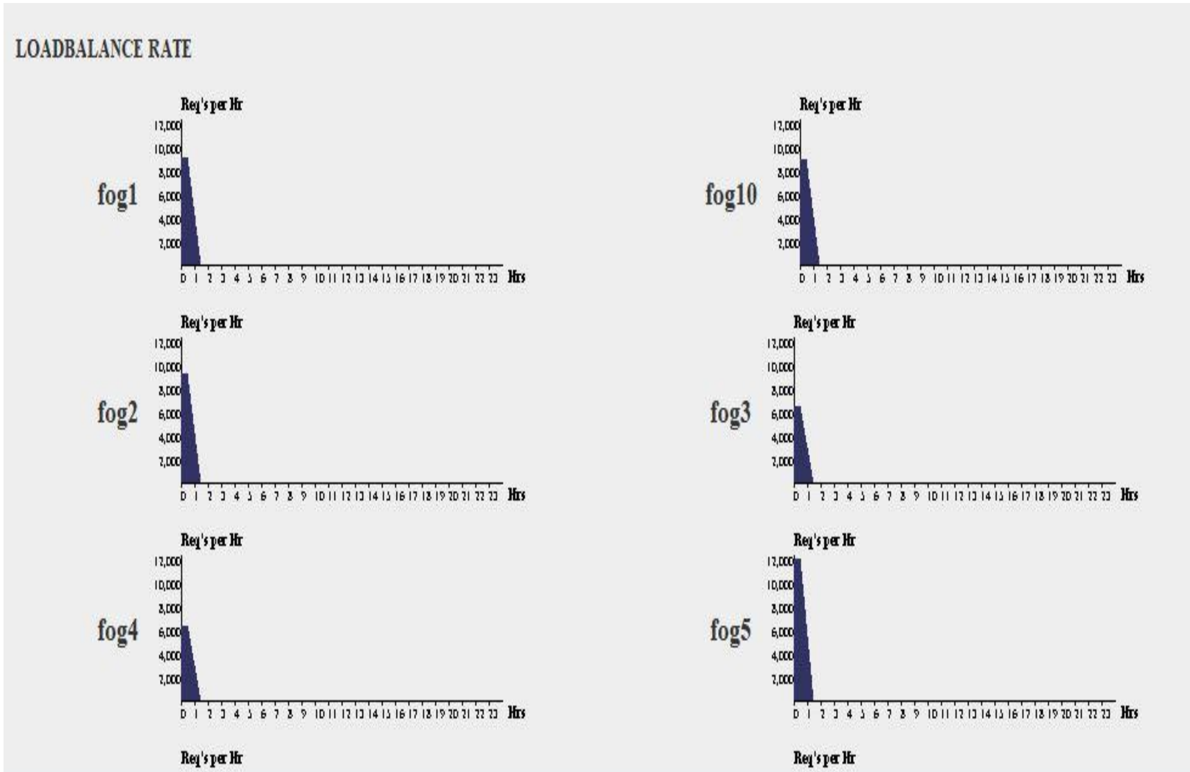
In the above figure you can see the average response time for 14 users.

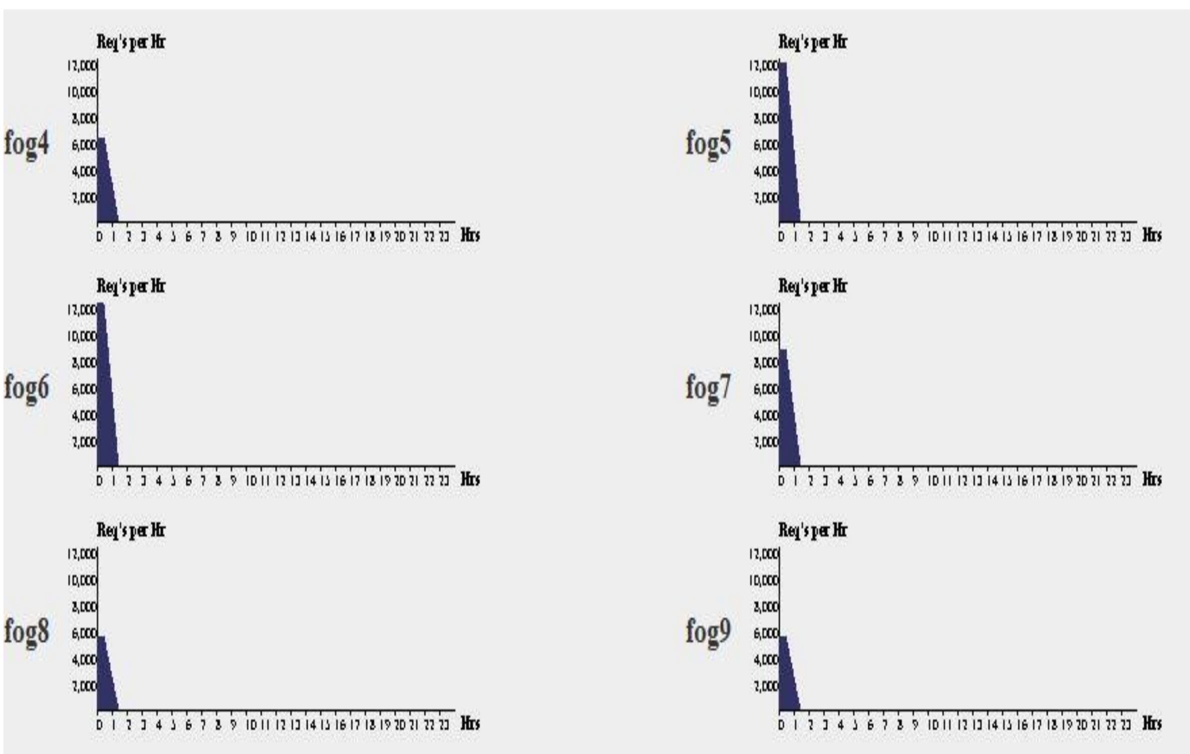
SCHEDULING TIME

Data Center	Avg (ms)	Min (ms)	Max (ms)
fog10	0.5	0.017	0.889
fog1	0.507	0.042	0.89
fog2	0.503	0.026	0.899
fog3	0.498	0.016	0.885
fog4	0.502	0.018	0.885
fog5	0.505	0.03	0.913
fog6	0.49	0.016	0.884
fog7	0.505	0.021	0.892
fog8	0.505	0.015	0.881
fog9	0.499	0.019	0.882

DELAY

Figure 4.27: Scheduling time for fog nodes case2 with priority





Graph4.6: load balancing graphs for fog nodes case 2 with priority

In the above the load balancing graphs for each fog nodes are shown.

Cost

Total Virtual Machine Cost :	**$5.00**
Total Data Transfer Cost :	**$0.90**
Grand Total :	**$5.90**

Security Measure

Encryption excution time (milliseconds) :	**617**
Decryption excution time (milliseconds) :	**11775**

Figure 4.28: Cost and security measure for case with priority

As you can see in the above figure that the total cost for the system is coming out to be 5.90.

Data Center	VM Cost	Data Transfer Cost	Total
fog1	0.5	0.097	0.597
fog2	0.5	0.098	0.598
fog3	0.5	0.069	0.569
fog4	0.5	0.068	0.568
fog5	0.5	0.128	0.628
fog6	0.5	0.13	0.63
fog7	0.5	0.094	0.594
fog8	0.5	0.059	0.559
fog9	0.5	0.06	0.56
fog10	0.5	0.095	0.595

Figure 4.29: data centers cost for case 2 with priority

Now we will run the simulation for case without priority scheduling change the scheduling method to round robin using the advanced window.

Overall Budget and Time Summary

	Schedule Average (ms)	Schedule Minimum (ms)	Schedule Maximum (ms)
Overall Response Time:	50.17	37.11	62.63
Data Center Processing Time:	0.50	0.02	0.92

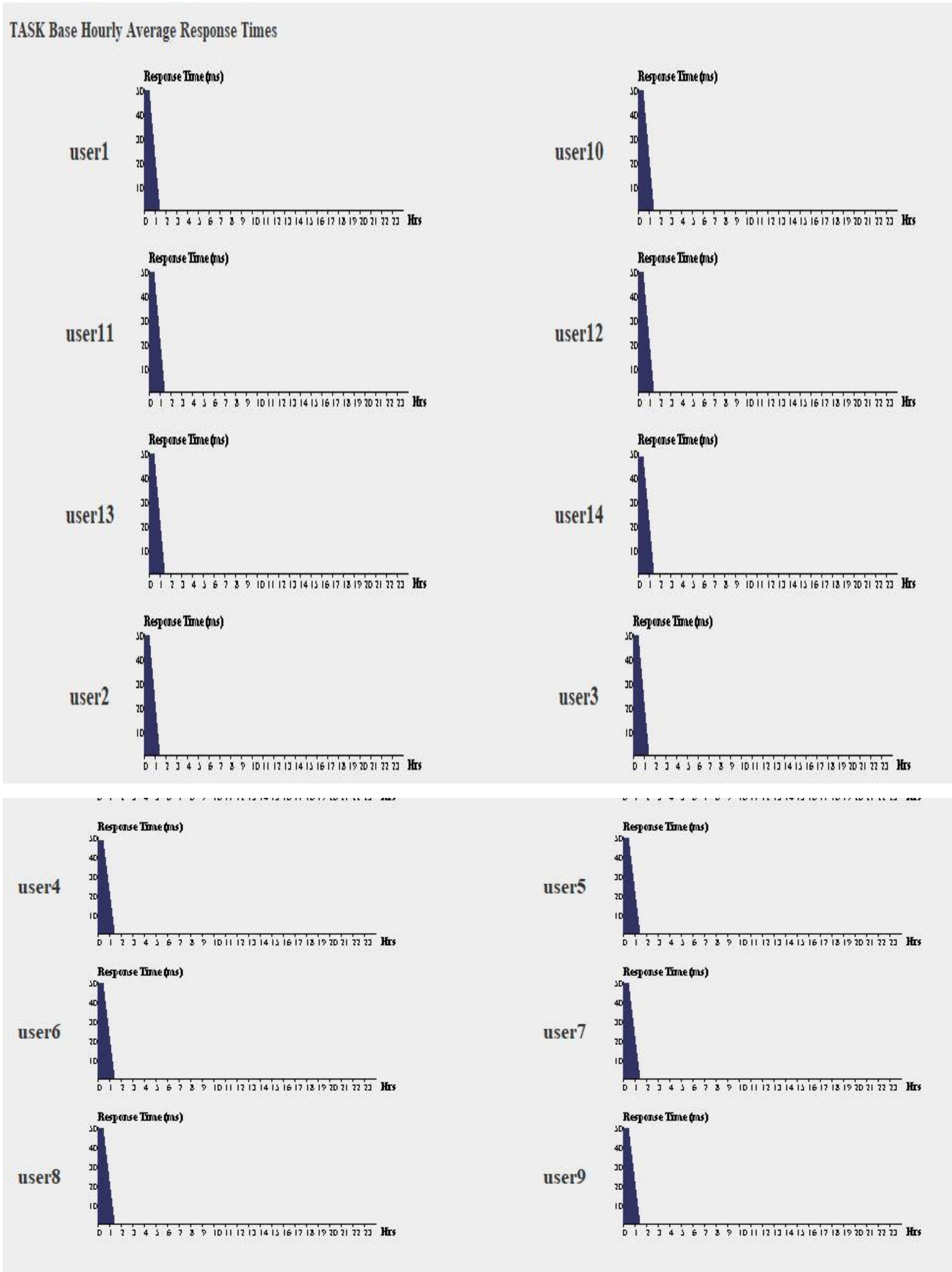
Export Results

RESPONSE TIME FOR NO OF TASK

Tasks	Avg (ms)	Min (ms)	Max (ms)
user10	50.15	40.136	61.384
user11	50.223	39.646	60.659
user12	50.331	40.313	60.884
user13	50.13	39.662	61.677
user14	49.994	39.636	61.631
user1	50.201	39.131	62.633
user2	50.031	37.135	61.382
user3	50.155	40.14	60.05
user4	49.896	39.144	61.393
user5	50.123	38.88	61.12
user6	50.323	40.88	61.131
user7	50.466	38.897	61.64
user8	50.102	37.11	61.132
user9	50.348	39.913	62.155

Figure 4.30: Overall Budget summary for case without priority

In the above figure you can see that the response time for the users is more as comparison to response time in application with priority.

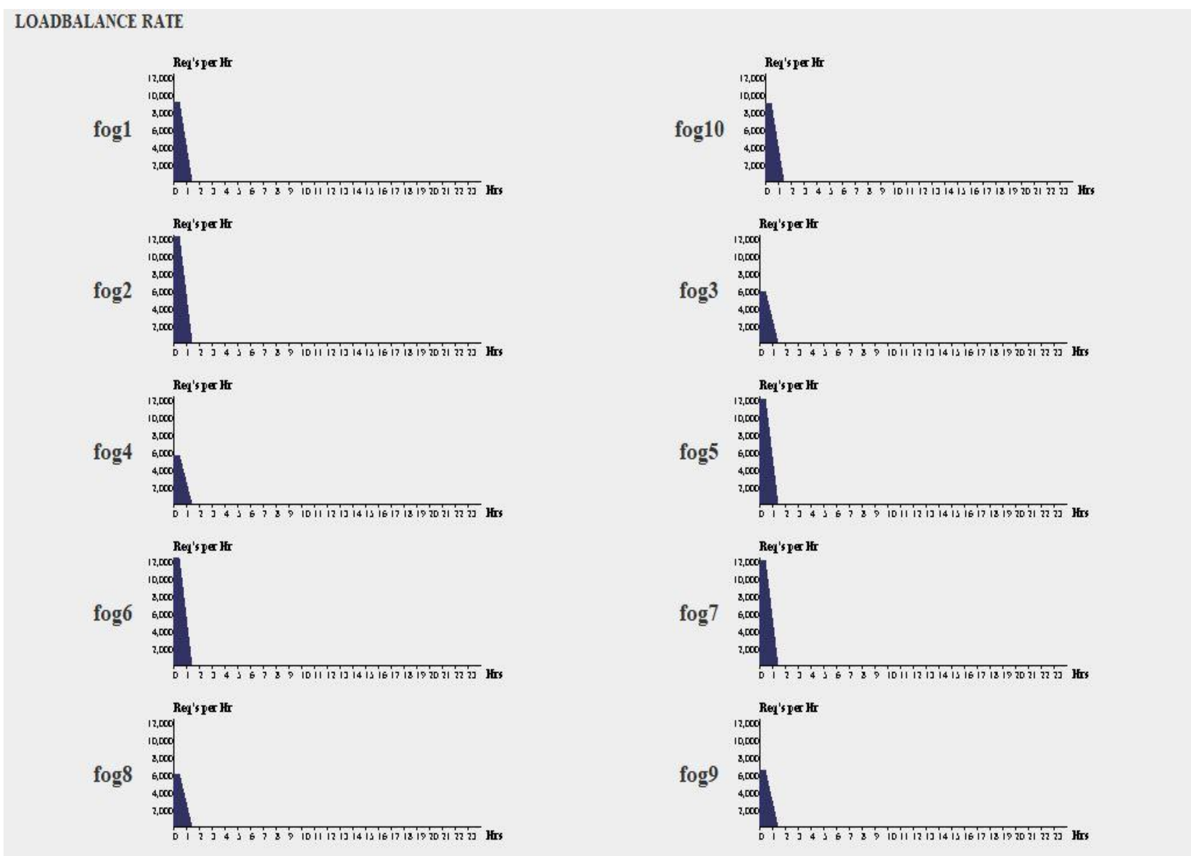


Graph4.7: Task base average response time for each user case2 without priority

SCHEDULING TIME

Data Center	Avg (ms)	Min (ms)	Max (ms)
fog10	0.49	0.018	0.885
fog1	0.499	0.017	0.886
fog2	0.504	0.021	0.912
fog3	0.496	0.016	0.88
fog4	0.504	0.018	0.888
fog5	0.501	0.027	0.91
fog6	0.51	0.016	0.883
fog7	0.493	0.025	0.923
fog8	0.495	0.016	0.879
fog9	0.513	0.018	0.891

Figure 4.31: Scheduling time for case2 without priority



Graph4.8: Load balancing rate case2 without priority

Above figure is showing the load balance rate for each fog device.

Cost

Total Virtual Machine Cost :	$9.00
Total Data Transfer Cost :	$0.96
Grand Total :	$9.96

Security Measure

Encryption excution time (milliseconds) : 617

Decryption excution time (milliseconds) : 11775

Data Center	VM Cost	Data Transfer Cost	Total
fog1	0.5	0.097	0.597
fog2	0.5	0.129	0.629
fog3	0.5	0.063	0.563
fog4	0.5	0.06	0.56
fog5	0.5	0.128	0.628
fog6	0.5	0.13	0.63
fog7	0.5	0.128	0.628
fog8	0.5	0.065	0.565
fog9	4.5	0.068	4.568
fog10	0.5	0.095	0.595

Figure 4.32: Cost and security measure case2 without priority

From the above results it is clear that total cost in priority scheduling is less as comparison to another scheduling algorithm.

CHAPTER-5

CONCLUSIONS

The fog computing is nowadays a emerging filed in IT sector because this is providing a very good compliment for the cloud computing. From the above discussions we have concluded that in the applications where latency is an issue that is where we can't tolerate can kind of delay in the application if any delay would come in the application then there would be harmful effects, in such situations fog computing is suitable way because latency or can kind of delay is going to decrease. We have also studied about the latest research in this field. Fog computing is actually a real new topic and very little research work is done in this field and many more things can be possible by using the fog computing technology. Next we have seen that for any network security is major issue but another difficulty also comes with the fog computing that this network is distributed on security of fog nodes is a major issue the main thing that we have to deal here is with the processing limitations of the fog nodes so the previously available algorithm like RSA that is used in the cloud computing would increase the processing overhead so we have chosen ECC in which packet size is size and there would not be any processing overhead on fog nodes. Next, we have tried to decrease the implementation cost of the system by applying a proper scheduling algorithm. By running the simulation with different scheduling algorithms, we have found that priority scheduling algorithm will decrease the overall response time of the system and also decrease the cost of the application. Some of the applications of the fog computing are the smart traffic lights, connected cars, self-maintaining train, smart building control. Let us discuss these applications one by one:

- Smart traffic lights: we can use the fog computing technology to create the smart traffic lights where the smart traffic light can act as fog devices which can sense the flashing lights of the ambulance and according to that it can change the traffic light signal so that the ambulance can easily pass through the lanes in the traffic.
- Connected cars: Nowadays self-driven cars are in trend by using fog computing we can build these auto-driven cars which can easily park themselves without any driver this can be possible with the fog computing because fog computing provides

a real time interaction between the cars and this can also help in reducing the number of cars accidents in the roads.

- Self-maintaining train: This is another application of the fog computing where the ball bearing sensor present in the train can sense the changes in the temperature and give an indication to the driver so that he can make appropriate changes, in this way we can easily avoid any kind of disaster.
- Smart building control: Smart buildings can be build using the fog computing technology in these buildings we can fit different kind of sensors to measure the change in room’s temp, moisture of the room or the various gases level in the room , these sensors can gather the information and can exchange with each other to generate the useful information according to which it can change the room temp and try to increase the fresh air in the room.

The future work that can be possible is that there is a possibility of improving the scheduling algorithm this can be achieved by applying the dynamic priority settings in response to the traffic in simple words the threshold values T1 and T2 that we have taken can be dynamically changed.

REFERENCES

[1] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the art and research challenges," J. Internet Services Appl., vol. 1, no. 1,pp. 7–18, May 2010.

[2] X. Zhu et al., "Improving video performance with edge servers in the fog computing architecture," Intel Technol. J., vol. 19, no. 1,pp. 202–224, Apr. 2015.

[3] S. Agarwal, S. Yadav, and A.Yadav, "An Efficient Architecture and Algorithm for Resource Provisioning in Fog Computing", in MCEP, 2016,doi: 10.5815/ijieeb.2016.01.06

[4] Chin Tsai and Melody Moh. 2017. Load Balancing in 5G Cloud Radio Access Networks Supporting IoT Communications for Smart Communities. Proceedings of 2017 IEEE Int. Symposium on Signal Processing and Information Technology (ISSPIT), Bilbao, Spain, Dec2017.

[5] Gurpreet Kaur and Melody Moh, "Cloud Computing Meets 5GNetworks: Efficient Cache Management Algorithms in Cloud Radio Access Networks," in Proc. of the ACM Annual Southeast Conference ACMSE), Richmond, KY, Mar 2018.

[6] M.Dakshayini and H. S. Guruprasad, "An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment", International Journal of Computer Applications (0975 – 8887), ISE, Bangalore, India, Volume 32– No.9, October 2011

[7] A.Ingole, S. Chavan, and U. Pawde, "An Optimized Algorithm for Task Scheduling based on Activity based Costing in Cloud Computing", IJCA Proceedings on 2nd National Conference on Information and Communication Technology NCICT(3):34-37, Nov., 2011

[8] J. Xu, B. Palanisamy, H. Ludwig, and Q. Wang, "Zenith: Utility-aware Resource Allocation for Edge Computing", IEEE International Conference on Edge Computing., 2107, 10.1109/IEEE.EDGE.2017.15

[9] R. Singh, S. Paul, and A. Kumar, “Task Scheduling in Cloud Computing: Review”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6), 2014, 7940-7944

[10] M. Verma, N. Bhardwaj, and A. Yadav, “Real Time Efficient Scheduling Algorithm for Load Balancing in Fog Computing Environment”, in MCEP, 2016, doi: 10.5815/ijitcs.2016.04.01

[11] C. Pawar and R.Wagh, “Priority Based Dynamic resource allocation in Cloud Computing”, Intelligent Systems and Signal Processing (ISSP),2013 International Conference, 10 June 2013

[12] E. Elghoneimy, O. Bouhali, and H. Alnuweiri, “Resource Allocation and scheduling in Cloud Computing”, Proc. Of the IEEE International Workshop on Computing, Networking and Communications, 2012, pp.309 – 314.

[13] A. Marphatia, A. Muhnot, T. Sachdeva, E. Shukla, and L. Kurup,“ Optimization of FCFS Based Resource Provisioning Algorithm for Cloud Computing”, IOSR Journal of Computer Engineering (IOSRJCE)e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 10, Issue 5 (Mar.- Apr. 2013), PP 01-05

[14]G.Ateniese, K. Fu, M. Green, and S. Hohenberger, ``Improved proxy re-encryption schemes with applications to secure distributed storage," ACM Trans. Inf. Syst. Secur., vol. 9, no. 1, pp. 130, 2006.

[15]P.Yuriy, R. Kurt, S. Gyana, and V. Vinod, ``Fast proxy re-encryption for publish/subscribe systems," ACM Trans. Privacy Secur., vol. 20, no. 4,p. 14, 2017.

[16]G.Ateniese, K. Benson, and S. Hohenberger, ``Key-private proxy re-encryption ,"in Topics in Cryptology CT-RSA (Lecture Notes in Computer Science), vol. 5473, M. Fischlin, Eds. Berlin, Germany: Springer ,Berlin, 2009

[17] M. Green and G. Ateniese, ``Identity-based proxy re-encryption,'' inApplied Cryptography and Network Security. Berlin, Germany: Springer,2007.

[18] Z. Wang, ``Leakage resilient ID-based proxy re-encryption scheme for access control in fog computing,'' Future Generat. Comput. Syst., to be published.

[19] Y.-J. Song and J.-M. Kim, ``Secure data sharing based on proxy re-encryption in fog computing environment,'' in Proc. Asia Pacic Appl.Sci. Eng. Better Hum. Life, 2016, pp. 5256.

[20]Carla,,mouradian , DialaNaboulsi, Sami, Yangui,RochH.Glitho,Monique J. Morrow,Paul A. Polakos, ”A comprehensive survey on fog computing ”,IEEE communications surveys and tutorials,Vol.20,No.1,First quarter 2018.

[21] Yunyoung Nam,” Analysis of lightweight encryption scheme for fog-to-things-communication”,10.1109/ACCESS.2018.2822822.

[22] https://doi.org/10.1145/3190645.3190699