Faculty of Science, Technology and Communication

Service Level Agreement Assurance in Cloud Computing Data Centers

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Information and Computer Sciences

Author:
Abdallah Ali Zainelabden Abdallah IBRAHIM

Supervisor:
Prof. Pascal Bouvry

Reviewer:
Prof. Ulrich Sorger

Advisor:
Dr. Dzmitry kliazovich

July 2015

Abstract

SaaS is providing cloud applications like all the normal classes of applications of normal computing like the web applications, file applications, email applications, real-time applications, highly interactive applications, massive data analysis applications, high performance computing applications and mobile cloud applications. Cloud computing uses the internet data centers to host the applications and data storage and also the processing power in the addition to the virtualization. Clouds are a huge stack of easily and usefully virtualized services and resources (Like software, platform and hardware). These resources are used by people all over the world and dynamically configured to accommodate more and more load and scale to a huge number of users. This stack of resources and services is delivered to the customers by a pay-per-use model which verifies that the services provided by the cloud providers are provided by means of service level agreements.

In this master thesis, we present an assurance to the service level agreement between the cloud users and the cloud services providers by trying to assess and evaluate the cloud computing data centers services and applications provided to the cloud users. Also we introduce mathematical test models for the web, file, real time and distributed applications. We can use these test models to see the behavior of the applications over a long time.

In this master thesis, we classify the cloud applications into 8 different classes of applications and then identify the failures and problems which affect these cloud applications. We classify also all failures types which may occur in cloud data centers like network failures, physical server failures, VM unavailability, individual failures racks and individual component failures. We got the important metrics for each class of applications and developed real scenarios for four classes. Then, we introduce different failures to these scenarios to see what is the impact in the most important metrics of applications. Then, we simulate these scenarios in Network Simulator 2. Finally, we solved and mitigated the failures in simulation by using two failure mitigation techniques. The first one is the virtual machine migration and redundancy. The second technique is the forward error correction.

Experimental results acquired over the course of this master's thesis give an assessment and verification of the cloud applications services provided by the cloud provider data centers. And also verify that the cloud data center provider gives their users the best services performance and high QoS, we use this verification to assure SLA response times metric and a good quality and performance of services as mentioned inside the SLA document. This verification will be given to the cloud customers.

Acknowledgment

"Today don't beg, don't ask, Just thank God in silence;
for all the blessings in your life. "

Paulo Coelho

The road to want something is always guided by unseen strong currents of support from people who believe that you will do it what you have set forward to do. Its only time that you will get back to the road from the path that was taken as a deviation.

The journey is the reward. It was a long trip to get to the point of writing a master thesis. But nevertheless, it was an experience I would not like to miss. I got to know many people and learned many things during that trip, and it was always a pleasure. The document you are about to read is the result of six months of work spent in SnT, at the University of Luxembourg, in the context of my Master Thesis.

My sincere acknowledgement to Prof. Pascal BOUVRY for allowing me to be under his guidance during this course of research work in my Master thesis; not only for being my supervisor but also who was always dedicated in all his lectures and gave me the possibility to do this master thesis with him, and to Prof. Ulrich SORGER for reviewing my thesis and who also was always dedicated in all his lectures. I also like to thank Dr. Dzmitry KLIAZOVICH my daily supervisor for understanding my strengths and weaknesses; and in advising me at every step with conducting experiments and analyzing results; Also for his tremendous support in helping me finish this project in time with all the logistics that was necessary.

A big thank you to all my Professors and Doctors i have met with them during this master period for being there when I needed them the most and to all my colleges, for all their support and understanding. Special acknowledgement to My Mother, Mr. Ali Zainelabden Abdallah, My Sister and Mr. Mohamed Ali for being my pillars of strength through thick and thin.

Finally thanks to my family and friends for all their support and understanding. I need to apologize to my parents for not having visited them as much as I would have liked during this period. We'll definitely catch up soon enough.

Working on this thesis was a wonderful adventure thanks to all of you !

Abdallah IBRAHIM
July 17, 2015

List of Acronyms

- **BER** Bit Error Rate
- **CBR** Constant Bit Rate
- **CC** Cloud Computing
- **CDC** Cloud Data Center
- **DA** Distributed Application
- **DC** Data Center
- **DRAM** Dynamic Random Access Memory
- **DT** Delay Time
- **FA** File Application
- **FEC** Forward Error Correction
- **FTP** File Transfer Protocol
- **HIA** Highly Interactive Application
- **HPC** High Performance Computing
- **HSA** High Sensitive Application
- **HTTP** Hyper Text Transfer Protocol
- **IaaS** Infrastructure as a Service
- **IP** Internet Protocol
- **IRTP** Interactive Real Time Protocol
- **IT** Information Technology
- **KIM** Keep It Moving
- **LSA** Low Sensitive Application
- **MCC** Mobile Cloud Computing
- **MDA** Massive Data Analysis
- **MOS** Mean Opinion Score
- **NAM** Network Animator
- **NIST** National Institute of Standards and Technology
- **NS** Network Simulator
- **OTCL** Object Tool Command Language
- **PaaS** Platform as a service
- **PC** Personal Computer
- **POP3** Post Office Protocol 3
- **QoE** Quality of Experience
- **QoS** Quality of Service
- **RT** Response Time
- **RTA** Real Time Application
- **RTCP** Real-Time Transport Control Protocol
- **RTP** Real-Time Transport Protocol
- **SaaS** Software as a Service
- **SCTP** Stream Control Transmission Protocol
- **SIP** Session Initiation Protocol
- **SLA** Service Level Agreement
- **SMTP** Simple Mail Transfer Protocol
- **TCL** Tool Command Language
- **TCP** Transport Control Protocol
- **UDP** User Datagrame Protocol
- **VBR** Variable Bit Rate
- **VINT** Virtual Intent Testbed
- **VM** Virtual Machine
- **VmM** Virtual Machine Migration
- **VoIP** Voice over Internet Protocol
- **WA** Web Application

Contents

List of Figures

List of Tables

Chapter I

Introduction

"Whatever the mind of man can conceive and believe, it can achieve"

W. Clement Stone

1.1 Motivation

Cloud [2] is the new paradigm of delivering both the applications as a services over the Internet and the platform software systems which all hosted in the data centers that provide those services to the cloud consumers. Where the services is the term software-as-a service, and the hardware and the software running on this hardware inside the data center is refer to the cloud.

The cloud applications hosted in the cloud data centers are used by all the cloud users all over the world. Example of those everyday use applications like Skype, Facebook, Dropbox, Amazon, Gmail, Google Docs, Google Maps,..etc. And inside the cloud data centers there are many types of failures like network failures, physical server failures, VM unavailability failures,...etc. Once one of the previous failure happen on the cloud data centers, it will affect the performance of the applications and may corrupt or stopped it and this return to the class or the type of the application. This corruption or performance degradation of the application is refer to the violation of the service level agreement which is between the cloud data center provider and the cloud user. Where in the cloud platforms, the cloud service providers have to obligate by the SLA with the cloud user. In [7], For each SLA violation the cloud service provider should pay to the cloud customer a pre-defined penalty.

The violation to the SLA may happen because of two general reasons:

1. Failures in the SaaS services it self, which mean that there is a problem or a bug in the application software it self. This what we will not care in this thesis. SLA violations in SaaS systems return back to the company which develop the software it self. In this master thesis, we assume that there is no any software failures in the cloud application hosted in the cloud data center.

2. Failures in the infrastructure of the cloud data centers which happen in the cloud data center not in the software like the network and the hardware failures. These failures is represented as SLA violations in IaaS and in the PaaS. In this master

thesis we tried to find those failures in the cloud data centers and looked forward solve or at least mitigate them.

Throughout this thesis, we identify and classify the failures may happen in the cloud data centers. Then classify the cloud applications and how the data center failures affect these classes of cloud applications. After seeing the impact of the failures on the applications, we tried to solve or mitigate the failures to let the applications still working during the failures are happening.

In this thesis, we present a way to simulate the failures and the mitigation or the solution of the failures in NS2. Also present a classification to the cloud applications and which network protocols used in these applications. Also, how to keep the application moving in case of failure happen.

In this thesis, we introduce a mathematical model for four different classes of applications, and from them we can see the behavior of the applications for a long period of time.

By the previous explanation, during this thesis also we provide an assessment to SLA by evaluating the failures and their impact on applications. We also provide an assurance to the SLA between the cloud provider and the cloud customer by solving and mitigating the failures appear in the cloud data centers and keep the applications alive.

1.2 Objective

The objective of this thesis is to provide an assurance to SLA in the cloud computing data centers.

To achieve our objective in this thesis, we used to assessment first the impact of failures in the cloud data centers on the cloud applications. Then introduce ways of mitigating failures to keep the application moving during failures are happening.

1.3 Contribution & Thesis Organization

In this section, we will explain exactly my contribution in this thesis and how was the organization of the thesis run.

The Thesis was divided into 5 different phases, as the following:

- ***Phase 1 Research Phase*** : During this phase of the project, we searched for the cloud applications and the cloud DCs failures. Also in the same phase, we classified the cloud applications into classes and also identify the failures into types. Finally, in this stage we searched for how we will solve the failures and mitigate them.

- ***Phase 2 Implementation*** : During this phase of the project, we developed and implement a real scenarios of the cloud applications. Then we introduce failures to those scenarios to see the impact of the failures in applications metrics and the performance of the different classes of applications. Also getting some graphs which show the behavior of the realistic applications during failures are happening.

- ***Phase 3 Simulation On NS2*** : During this phase of the project, we have moved from the reality implementation to the simulation implementation. Where we implement and simulate the classes of applications. Then introduce failures in the simulation and again see the impact of these failures on the different metrics for each class of applications. In this phase, the implementation was more dynamic and flexible than in the real application development, where in this stage we could implement more classes of applications than in the realistic development stage. Also getting some graphs which show the behavior of the simulated applications during failures are happening.

- ***Phase 4 Solving & Mitigation*** : In this phase of the project, we solved and mitigated the failures happen in the CDCs and which affect the classes of applications. Because of the poor of the real devices and the difficulty of the real development of the applications, failure mitigation is only simulated in NS2. Also getting some graphs which show the behavior of the simulated solutions and mitigation of failures on applications metrics.

- ***Phase 5 Analysis & Documentation*** : This is the last stage in the project which contained three steps. The first step was the analyzing of experimental results we have got in the previous stages. The second step was the writing of the scientific conference paper and try to publish it, whatever before defense or after. The third step was the writing of the master thesis project report and document every thing.

The last thing in this section, is the list of real contribution in this master thesis project. The followings are our contributions :

1. Identify the cloud computing applications and then classify them into different classes.
2. Identify the different types of failures in the CDCs.
3. Develop real applications to test real failures on it. And get some plots.
4. Provide a mathematical model for testing some classes of applications.
5. Simulate the classes of applications on NS2.
6. Introduce failures in the simulation and get some plots.
7. Solve and mitigate failures identified in the CDCs in the simulation.
8. Write a scientific conference paper to document our thesis project and publish it.

1.4 Structure

The remainder of the thesis is structured as follows :

We give a details overview and background about cloud computing, SLA, Cloud data centers and the NS-2 simulator in ***Chapter 2***. In ***Chapter 3***, we discuss not briefly the related work which relevant to SLAs, CDCs, cloud applications and the cloud simulators. The description of our problem and the problem statement in addition to the methodologies used in this problem are in ***Chapter 4***. The first contribution in this thesis work which is the classification of the cloud computing applications into categories is in ***Chapter 5***. The failures of the cloud data centers are identified and classified in ***Chapter 6***, also how we can solve the failures. In ***Chapter 7***, we present our real scenarios experiments for the cloud applications and the mathematical based models we got from the experiments. The NS-2 simulator experiments for the applications and the failure mitigation implementation are in ***Chapter 8***. ***Chapter 9*** shows our experimental results from the real scenarios experiments and the simulators experiments. We discuss the whole work in this master thesis in ***Chapter 10***. ***Chapter 11*** reviews most of the challenges we faced in this master thesis work and implementations. Finally, we conclude our work by summery of contribution, summery of finding and the future work in ***Chapter 12***.

Chapter II

Background

"It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts."

Sherlock Holmes -A Scandal in Bohemia

In this chapter, we introduce the background to fully understand this master thesis. We start by an introduction to the CC field in Section 2.1 as it is the field of interest of my work. In Section 2.2, we present a sub-field of CC, the SLA where we need to assure the services provided in it. Then we review the structure of the Cloud Computing DCs in Section 2.3. The CDCs is the place where the cloud providers services are hosted, so we need to review the DCs network and how they work.

We finish by giving an overview of the Network Simulator 2 and TCL as our work is heavily based on it. We used NS2 simulator to simulate our experiments where it is more flexible than the real experiments.

2.1 Cloud Computing

Because of the surpassing success of internet and the applications on the internet in the last years, resources of computing are now more ubiquitously available any where and any time. This computing resources has enabled the realization of a new style of computing which is called cloud computing. In the new style of computing, the resources (e.g, CPU and storage) are provided as general utilities that can be leased and released by users through the Internet in an on demand fashion. Inside cloud computing environment, the traditional service provider role is divided two providers. The first provider, is the infrastructure providers who manage cloud platforms and lease resources according to a usage based pricing model (according to usage). The second provider, is the service providers who rent computing resources from the infrastructure providers (one or more than one provider) to introduce services or to serve the end users. From the industry point of view, cloud computing has made a massive impact on the IT industry over the last few years. The proof of that is the philosophy of the big companies, like Amazon, Google and Microsoft. Those large companies strive add more powerful, reliable and minimized cost cloud platforms, and business projects search to reconstruct their business models to earn money and gain benefits from this new style (paradigm) of computing.

There are a lot of convincing features in the cloud computing which make it attractive and convenience to the business companies owners. Like:

1. Easily manageability and easy accessibly: All the services inside the cloud is based on the web and offered through the internet. So, those services are easily manageable and accessible on many kinds of devices through the internet connection. Those kinds of devices are not only the computers and laptop computers, but also cell phones and PDAs.

2. High scalability and high availability: The cloud computing infrastructure offer a huge amount of resources from data centers and make them available for using and accessing easily. The service provider could by easy way, extend and publish its services to a large scales in order to manage and handle high speed increasing of services demands (e.g, flash-crowd effect). This model is sometimes called surge computing [11].

3. Lower initial investment [3]: If you already have a computer and an Internet connection, you can very likely take advantage of most cloud offerings without investing in any new hardware, specialized software, or adding to staff. This is one cloud computing advantage that has universal appeal regardless of the industry you are in or type of business you run.

4. Pay as You Go: Cloud computing uses a pay-as-you-go pricing model [11]. Large upfront fees are not the norm when it comes to cloud services. Many software as a service applications and other types of cloud offerings are available on a month to month basis with no long term contracts.

5. Faster deployment: If you have a team work, you can be up and running faster with the services provided by the cloud than you can if you want to buy, plan and build in your own company. With many software as a service applications or other cloud offerings you and your team can start using the service within hours or days rather than weeks or months.

6. Independent location and independent devices: Cloud computing services provided over the internet, so you can use and access the services from any where and from any device at home or at work. And the cloud services not also developed to use it in a special browser or operating system. So it can be accessed via PC, Mac, on tablets like the iPad, and through mobile phones. Accessing from anywhere and from any type of devices is a very useful advantage for the people who travel a lot or moving a lot in the work when their work is spread out across multiple locations.. So this will let them able to work from home and use the services from any where.

7. Reliability and security: It is extremely important to your business that your services remain up at all times. Therefore, you need a reliable cloud service provider that can guarantee service level and up-time. Cloud computing ensure the reliability and the security of the service provided by the cloud.

Fig. 2.1 summarizes the cloud features and characteristics.

2.1.1 Definition

From the previous nutshell introduction about cloud computing, I can now define the term cloud computing like the following definition:

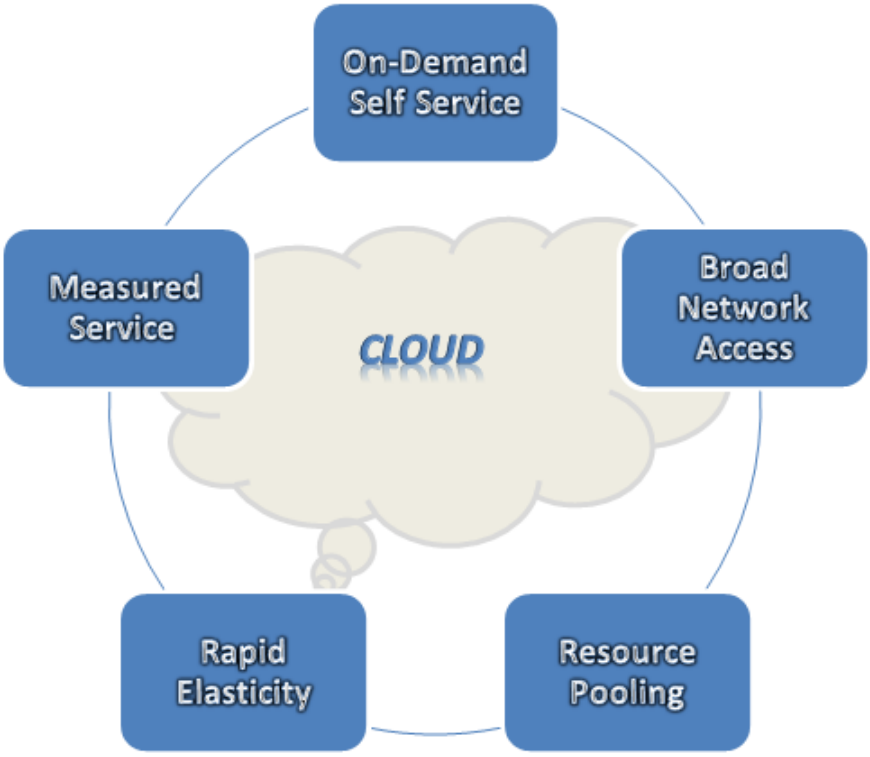


Figure 2.1: Cloud Features and Characteristics

"Cloud computing can be defined as a new style of computing in which dynamically scalable and often virtualized resources are provided as a services over the Internet." [1]

I can say that, Cloud Computing is a method of computing which depending on the Internet to use software or the other IT services on demand. With cloud computing, the resources and the costs are shared. Users can pay to only the services they need to use it at any time. As we mentioned before, cloud computing is like a business model. Providers of the cloud services provide all the services over the Internet, whatever what are the services. They are software, hardware or platform. The national institute of standards and technology [11]:
*"**NIST definition of cloud computing** Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."*

And the Final Version of NIST Cloud Computing Definition [3]is:
"cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g, networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

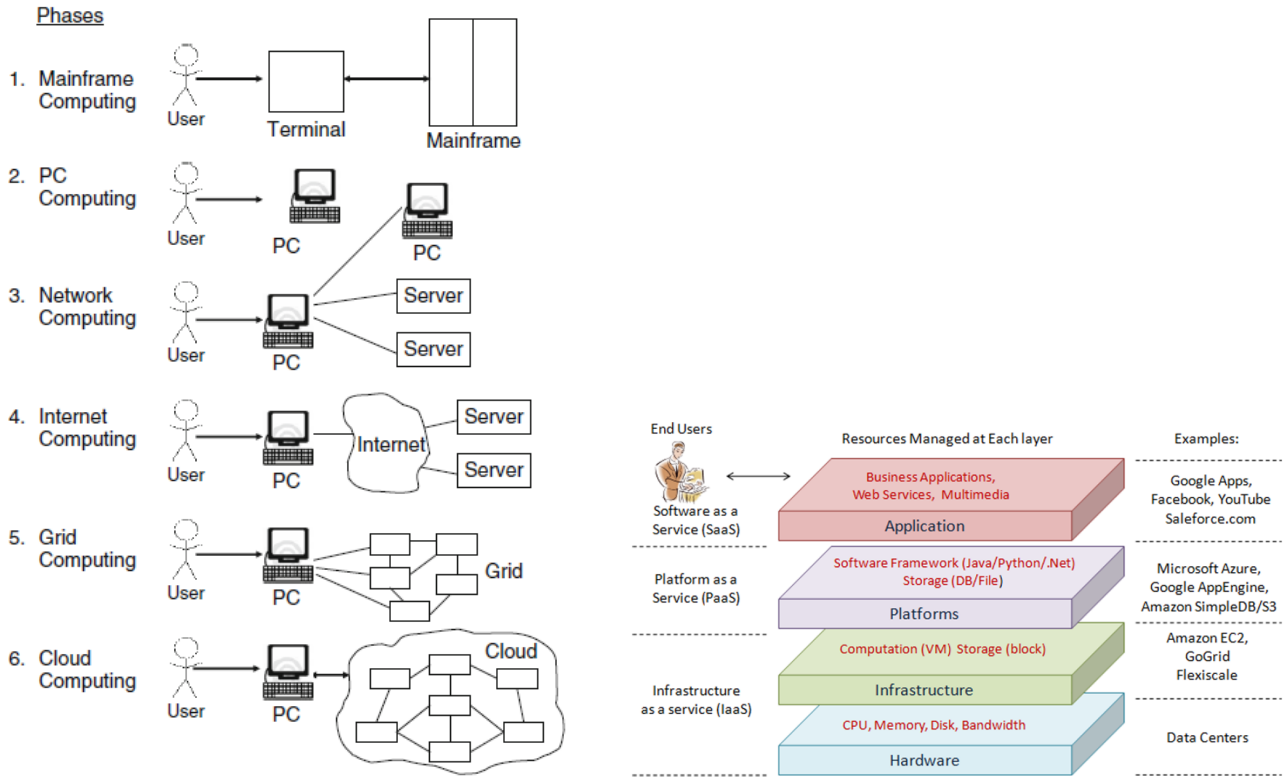
2.1.2 Computing Paradigms & Cloud Types

From [1], there are six phases of computing was evolved during the time. The six phases are from mainframe computing till cloud computing. And by the cloud computing evo-

lution, it helps the ubiquitous computing to achieve it's goals in many fields. Let's see the 6 phases of the computing paradigms as in Fig. 2.2a.

The first phase is the mainframes. In this phase many users shared powerful mainframes using terminals. The second phase is PC computing where stand-alone PCs became powerful enough to achieve the users' needs. The third phase is Network computing that PCs, laptops, and servers were connected together through local networks to share resources and increase performance and reliability. The fourth phase is Internet computing where local networks were connected to other local networks forming a global network such as the Internet to utilize remote applications and resources and connecting to the network from any where provide the Internet connection. The fifth phase is Grid computing where is grid computing provided shared computing power and storage through a distributed computing system. The sixth is Cloud computing that is like what i introduce before.

There are three types of cloud computing. The public cloud, the private cloud and the hybrid cloud. In the public cloud (or external cloud) computing resources are provided through the Internet via web service or web application from a third party provider. In this type, the applications of multiple different users are mixed together in the cloud's servers, storage systems, and networks. The Private cloud (or internal cloud) is like a private network but it is inside the cloud. It is used and managed for IT company or to a cloud provider. A hybrid cloud environment is the combination between the first two types of cloud computing. It combines a public and a private cloud together to distribute and provide services and applications across both a public and private cloud.



(a) Computing Paradigms (b) Architecture Layers

Figure 2.2: Cloud Computing [1]

2.1.3 Cloud Computing Architecture

The layered architecture of the cloud computing is like a stake of services provided by the cloud. Like SaaS, which is shown on top of the stack. SaaS is represent the application layer for the architecture of the cloud computing. SaaS allows the user to run and use software applications remotely from the cloud. This is like all nowadays philosophy of mobile and and we applications which allow the computing is ubiquitously. The second element in the stake is the Paas which represent the platform layer for the architecture of the cloud computing. PaaS is provide the operating systems and the platforms and the required services for a specific application. The last element in the stake is IaaS which represent the infrastructure layer (or the visualization layer) for the architecture of the cloud computing. IaaS provide a very big amount of storage and computing resources and it guarantees processing power and reserved bandwidth for storage and Internet access. This layer enable us to guarantee the accessing and the processing of the ubiquitous data be more rapid and consistent. And this due to the large amount of ubiquitous data we need to store and access it. Finally to store the ubiquitous data, The data-Storage-as-a-Service (dSaaS) provides a data centers to store the data by including bandwidth requirements for the storage. Fig. 2.2b from [11] summarize the previous layers of cloud computing.

2.1.4 Understanding Cloud Computing Applications

Simply cloud computing is the aggregation of software computing and offered services which available from cloud data centers. Cloud data centers consists of decentralized network of servers.

Nowadays, the term *"Cloud"* is used instead of the internet services that the people are using and enjoying with it [11]. While the users are using the popular services and websites, they don't know that it is a cloud based. Many applications like web-based email clients like Yahoo and Gmail, Wikipedia and YouTube, also peer-to-peer networks like Skype or Bit Torrent and Social networking sites are all applications that run in the cloud. All this just need a web browser and an Internet connection, also the applications have no centralized location or a company that organize them.

Cloud computing for the business is called usually the enterprise cloud computing. Inside the companies, instead of buying or building the physical infrastructure which needed for the software applications and programs, the cloud resources like SaaS is a available for the companies need to build systems and software or use it. Where, if they don't use the cloud, they will need hardware and an infrastructure to support it such as office space, networks, servers, storage, power, cooling, and bandwidth. All these in addition to the developer and experts who run and mange them. Also this in case the company require to use applications like Microsoft, SAP, or Oracle.

Cloud computing provide a simple solutions to all the previous challenges faced by the companies, organizations and individual users.

2.1.5 Cloud Computing Related Technologies

Cloud computing usage is replacing many kind of technologies like grid computing, virtualization, utility computing and autonomic computing [12].

In this subsection, i discuss the different kind of technologies which included under the cloud computing. As the following:

1. ***Virtualization*** : Virtualization technique is meaning that the total installation of one computer or machine with all it is applications is running in another machine or computer [13]. The result form this installation is that the entire system has all the software applications which running on the server. And all these software applications are running now on a virtual machine.

   Virtualization is represent the cloud computing technology foundation, where the cloud users can use and access the storage, processing power and the applications hosted on the cloud data centers virtually without having any knowledge about the storage or the processing power or the applications on the server.

2. ***Grid Computing*** : Grid Computing is the computing between a network of computers that connected and utilized together to earn a large super computing resources like high computing power and high storage. These network of computers are hosted in different locations. By grid computing, we can access the network of computers to perform large and very complex computing operations [11].

   Usually there is a confusion between the understanding of cloud computing and grid computing although they are not the same and quit different, where Grid computing gather all the resources of many computers in the network to work all together to solve a single problem in the same time [13].

3. ***Utility Computing*** : Utility Computing is a computing model when the user can use the computing services and pay for these services. The model used to pay for the services is called "pay-per-use" model. In the cloud computing, is approximately the same where the model used in the cloud computing for billing is the "Pay-as-you-go" model for the business model, and that is why we can say that cloud computing is including the utility computing technology.

4. ***Autonomic computing*** : Autonomic computing is a technology was invented by IBM company. The meaning of this computing technology is that the computers automatically can mange themselves without any intervention from the human. In Autonomic computing, the computers also can correct and mitigate errors and failures to themselves without any intervention from the human.

   For example, when one computer node in the network has a failure and this computer has some program and applications running on it. The applications running on this computer are automatically transferred to another computer in the network and this consider as a self correction. In cloud computing also the same process is done but on the virtual machines migration and also the mitigation of the failures as we did in this master thesis.

The understanding of cloud SLAs is an important topic to review during this thesis work, where we need to assure SLA services provided from cloud providers to cloud customers. In the next section, we review SLAs and examples about them.

2.2 Service Level Agreement

Clouds are a huge stack of easily and useful virtualized services and resources (Like software, platform and hardware). These resources are used by all the people all over the world and dynamically configured to accommodate more and more load and scale to a huge number of users. Also introduce the resource utilization. This stack of resources and services is offered to the customers by a pay-per-use model which verify that the services provided by the cloud providers are provided by the ways of SLAs.

2.2.1 SLAs Overview

SLA is represent the document contract that is between cloud customers and cloud services providers. SLA is the part of a contract which defines exactly what services a service provider will provide and the required level or standard for those services. This contract is needed to be accepted, negotiated and agreed between the two sides, the cloud customers and the cloud providers.

At the end of the negotiation operations between the cloud customer and the cloud provider, they commit and proof an agreement. This agreement is represented as a SLA. This SLA works as the contract which contains the items for the expected level of service between the cloud customer and cloud provider.

SLA is working paralleled as the blueprint and the guarantee for the cloud providers services. From the previous nutshell introduction about SLA we can now define the service level agreement. SLA is an agreement between a cloud service provider and a cloud service customers or consumers where this agreement is relevant to the service level. Service level is meaning the quality of the services provided from the provider to the consumer. This agreement can be setup by writing a legal and formal contract, then signing it by the two sides (consumer and provider). The items which should mentioned in the SLA contract should respect to security, priorities, responsibilities, guarantees, and billing modalities [14]. Additionally to the items in the SLA contract, there are some metrics which should be guaranteed like availability, throughput, response time, ... etc.

Generally, SLA should verify the quality of the service provided to the consumer and this verification should be written in the SLA document. From the business point of view, the quality level can be three levels like basic, silver, gold, platinum [14].

From [14], there are two essential phases for service level agreement as the following:

- Agreement on the quality of service.
- Service monitoring at run time.

Fig. 2.3 [2] describes the cloud computing SLA architecture and the SLA document.

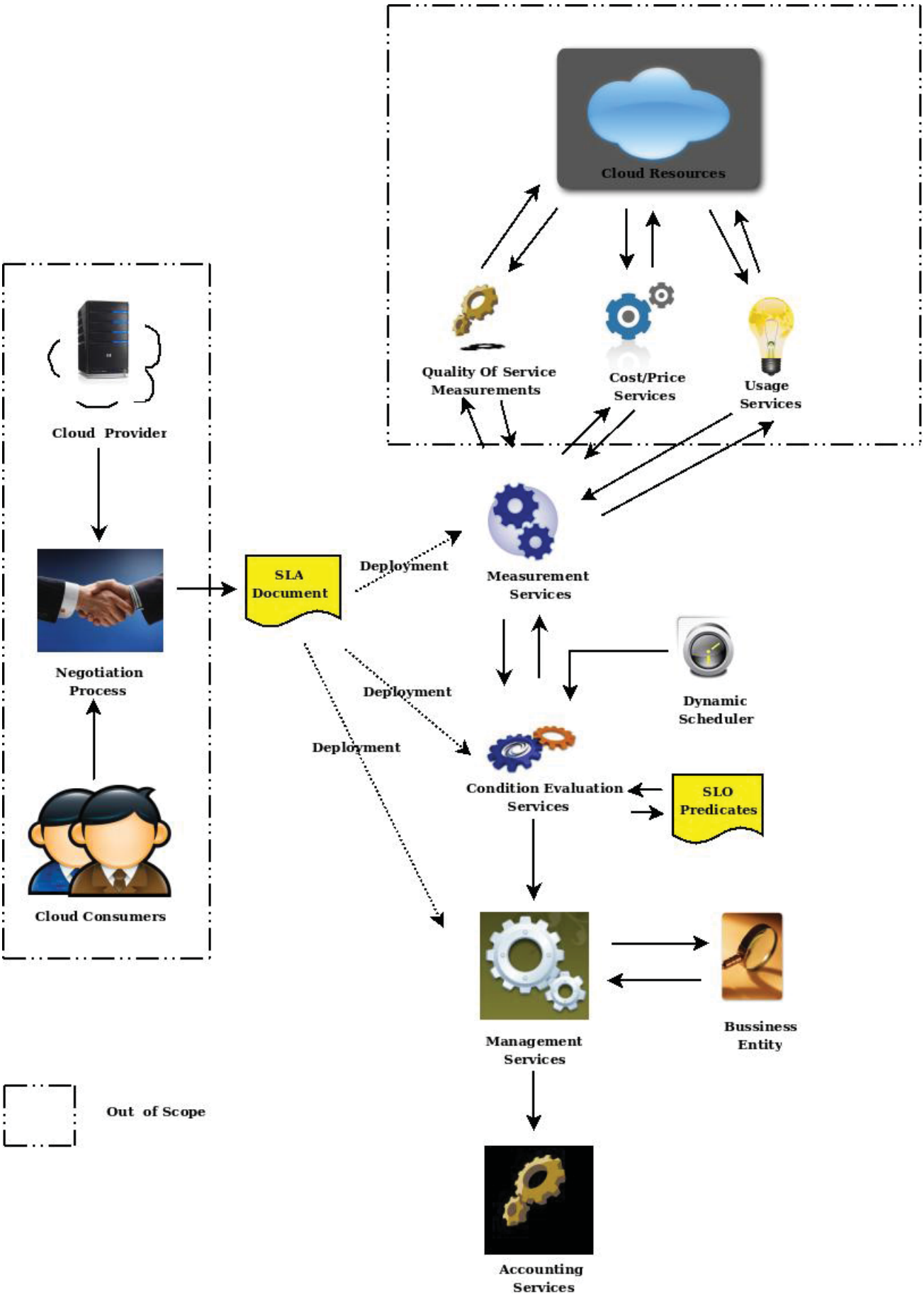


Figure 2.3: SLA Architecture [2]

2.2.2 Anatomy of a Typical Cloud SLA

A typical cloud SLA of a cloud provider contain the following components [15]:

1. ***Service guarantee :*** This component illustrates the different metrics while a provider seek to achieve in the service guarantee time period. Examples for service guarantees, like Availability (should be e.g, 99.9%), disaster recovery, response time (should be e.g, less than 50ms), and fault resolution time (e.g, within one hour of detection). If one of the previous metrics is failed, this will result in a service credit return back to the cloud customer.

2. ***Service guarantee time period :*** This component illustrate the time period mentioned in the SLA in which the service guarantee should be work and continue. This time can be month or at least a billing one hour.

3. ***Service guarantee granularity :*** This component describe the size of the resources which determined by the service provider and guarantee by the provider also. For example, the granularity can be on a per service, per instance, per transaction basis, or per data center..etc.

4. ***Service guarantee exclusions :*** This component is include the instances which are not included in the service guarantee metric calculations.

5. ***Service credit :*** This component describe the amount of credit will be given back to the customer if the service guarantee is not be done or not met.

6. ***Service violation measurement and reporting :*** This component describes the measures and reports that can tell ***how?*** is the violation to the service guarantee done and ***who?*** violate the service guarantee.

2.2.3 Why SLA ?

Number of cloud services customers is increasing world wide every day. In the other side cloud providers are trying to deploy and build data centers to let them introduce services to global distributed cloud users.

Data centers resources capacity is limited, so the providers try to distribute the load to the global data centers to provide a stable services to the customers. The providers also needs to guarantee the service quality level provided to the consumers. Providers can guarantee the service quality level to the customers by signing the SLA with the customers. SLA contains the level quality of the services provided to the customers and that is ***why*** all the community of the cloud computing need service level agreement. The providers also have to use the load balancing algorithms to prevent the SLA violations (e.g, the response time for the services).

SLA plays an important role in defining the relationship between cloud customer and his IT supplier. It guarantees him a certain level of service, giving him confidence that if something goes wrong, the supplier will respond quickly. Without a comprehensive SLA

in place, he is unlikely to have much comeback if his IT supplier fails to respond to his requests.

But the problem is that the service quality level is just written on the SLA document, but there is nothing verify on it. In this master thesis we tried to assessment quality of services provided to the customers, and this allow us to assure what the SLA document cover. An example for that is the Amazon EC2 SLA, where it guarantees about 99.95% of the availability of the services provided in a specific area over a 365 day period [13]. In our work, we focus on the performance and quality of the services provided from CDCs to cloud customers and the availability of those services is another topic.

2.2.4 What does a SLA Cover?

Example of SLA is a part of the contract between a business company and its IT supplier. The SLA sets out what levels of service are acceptable and crucially, explains what compensation you will receive if the IT supplier fails to meet these levels.

Typically, a SLA covers:

- **Uptime.** This applies to important software or services that the cloud customer company needs. Typically, uptime guarantees apply and access to servers, cloud services (like email or web hosting) or other parts of cloud provider IT system that are vital to cloud customer company business.

  For example, cloud provider (IT supplier) might guarantee 99.9% uptime for customer's cloud backup system. Uptime is the availability of services provided from cloud provider. These services should be available usually, we didn't assure this in this thesis.

- **Response times.** These measure how long it takes cloud provider (IT supplier) to respond when cloud customer raise a request for support. This represent the quality of services and the performance of the services provided by cloud provider.

  For example, cloud provider might promise to respond to critical problems within 15 minutes. Also, it might promise the high performance and quality of services even some failures happen in the cloud DCs. In this work, we need to assure the response time in the SLA and this by assuring the high QoS and high performance for the services provided.

The services provided by cloud providers are hosted on CDCs, so we need to review the architecture design of the DC to see what are the problems we may face the services provides and will harm the response time which should covered by the SLA. In the next section, we review DC network architecture where there are many failures may occur inside it. Also, we review the virtualization where it used inside the DCs. We need the reviewing of the CDCs because DCs represent the second side in the SLA contract which is where the cloud providers host their services. And to assure the SLA response time of the services, we need to know every thing about the place those services hosted in.

2.3 Cloud Computing Data Centers

Because of the highly increasing of the number of cloud applications and services users in the world wide every day, the cloud providers used to deploy and build data centers (or datacenters) to satisfy the increasing of cloud users and introduce services to the global distributed cloud users in all over the world.

Data center represents the brain of the big company and it is the place used by the company to run the most critical processes and to store the company data. In case the company provide some cloud services, the users for these services can access the data center from any where in all over the world.

2.3.1 Data Centers History

For long time ago, the large-scale computers systems have been used and were familiar in the big companies. In the 1940s, the large scale computers were very big, where they were needed to host in large rooms [16]. All these previous description is refereed to the mainframes computers. But nowadays, the individual PCs being much powerful than the mainframe systems from these days.

Even if after the appear of individual PCs, the very large-scale operations still need a complex IT infrastructure, or the combination of a group of computers systems together to do some complex operations. These complex IT infrastructure has a good large amount of hardware and processing power and they also still housed in a special room sizes. Theses rooms size let us say two names for these computer systems where if the rooms is not so large, we can say that it is just ***"server rooms"***. If the rooms are very big and huge amount of computer rooms it is called ***"Data Centers"***.

Data Centers are built by large companies or government agencies. Although, they are used increasingly to satisfy the rapid growing cloud services and solutions for business and private applications.

2.3.2 Data Centers Architecture Design

Data centers are used to a large computer systems and the related components to those systems like storage systems, the network and the telecommunications systems. Data centers also contain power supplies and the redundancy backup for them, also the redundant data communications links and connections. They have also air conditioning and fire suppression as environmental controls [3]. All this in addition to security devices like firewalls and security protocols on routers and switches of the network.

During the designing of the data centers, we have to respect the following key aspects [3] :

1. ***Resilience & Flexibility:*** This is refer to the scalability of the data center which ensure maximum usage without any degradation on the performance.
2. ***Availability:*** The Data Center continuity of working all the time is very important specially for the business.
3. ***Performance:*** Data center should be fast working and response by expected way. Where the more rapid is the best.

4. ***Security:*** Data center design should protect the data stored on it and the operations (on all data center resources) from any kind of theft and hacking. Also let the all the access under control.

5. ***Effective architecture for data separation:*** Data center infrastructure should provide network based backup, storage backup and perfect back-end network access.

6. ***Predictable fail over:*** Data center design should be ensuring service availability as maximum as possible.

Fig. 2.4 [3] describes the cloud computing Data Center architecture design.

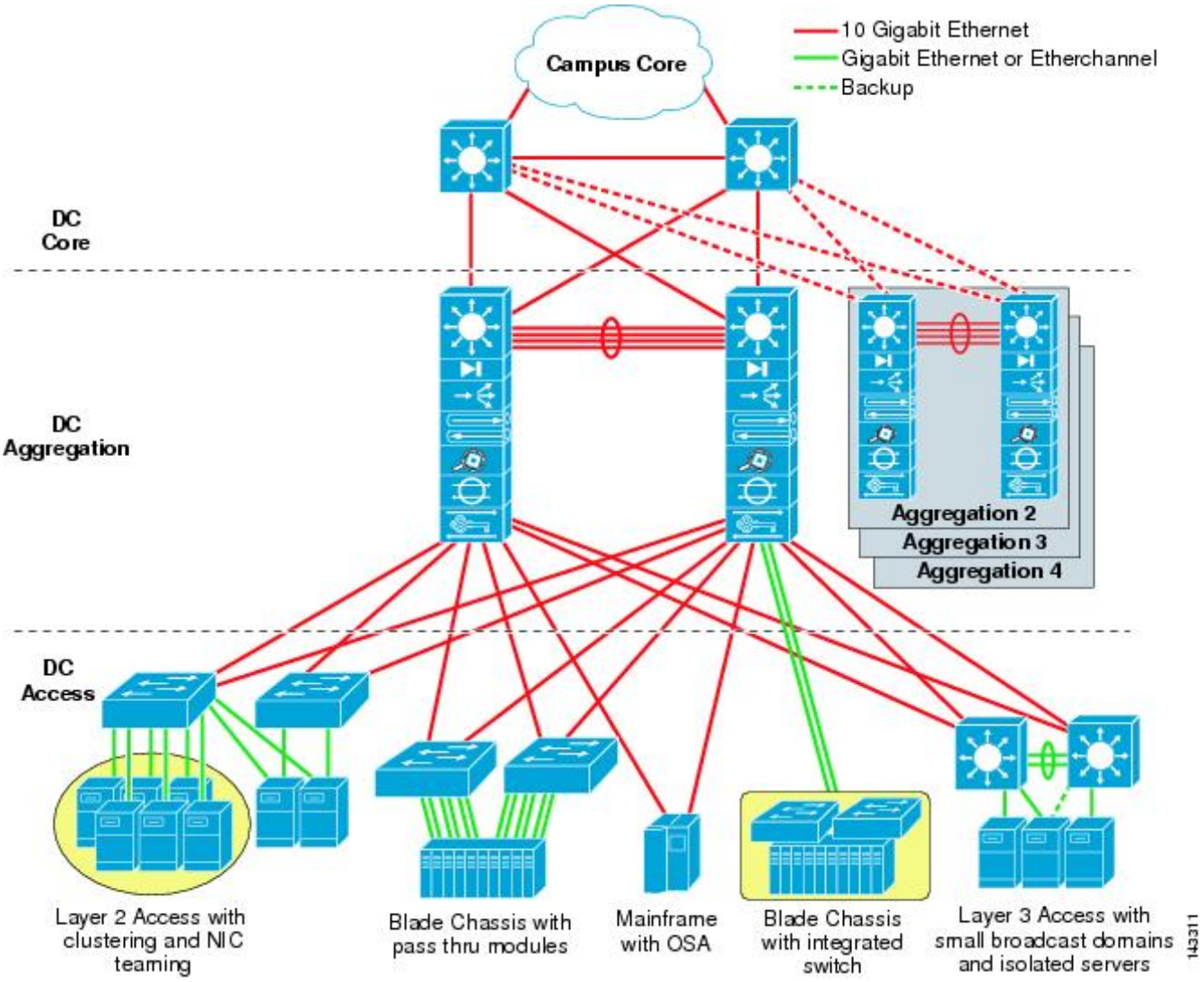


Figure 2.4: Data Center Architecture Design [3]

2.3.3 Data Center Network Architecture

Fig. 2.5 [4] describes the data center network architecture. In the network of the data centers, there are three layers. The first layer is the Internet layer. In this layer, the data center bridge to the world wide web network. The second layer in the data center contains the switches and the load balancers like the Aggregation switches and the Top

of Rack switches. The third layer in the data centers contains the Core and the Access Routers. Table (2.1 [4]), summarize all the devices abbreviations in the data center networks.

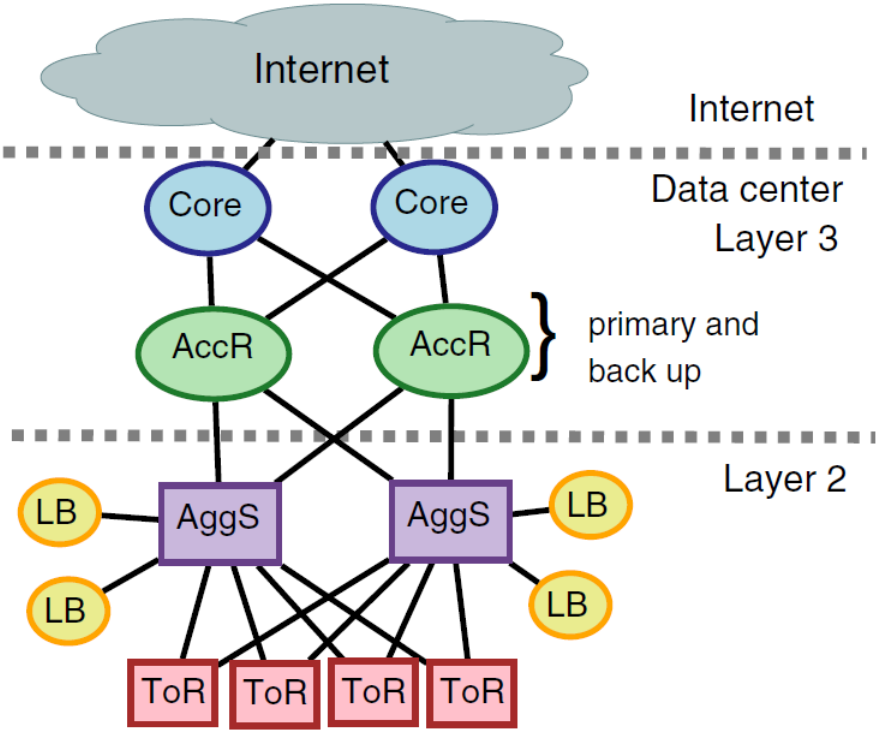


Figure 2.5: Data Center Network Architecture [4]

Table 2.1: Summary of device abbreviations

Type	Devices	Description
AggS	AggS-1, AggS-2	Aggregation switches
LB	LB-1, LB-2, LB-3	Load balancers
ToR	ToR-1, ToR-2, ToR-3	Top of Rack switches
AccR	-	Access routers
Core	-	Core routers

In the network, the rack of servers are connected to the Top of Rack (TOR) switches usually by 1 Gbps link. The Top of Rack (TOR) switches are connected to the Aggregation switches (AggS) and the backup ones for redundancy. The Aggregation switches are connected together and then connected to the upper layer with the Access Routers (AccR). The Access Routers (AccR) collect the traffic from thousands of severs and route it to the Core Routers which connected to the top layer of Internet. The Core Routers connect the whole Data Center network with the Internet.

All the links in the data center networks are using Ethernet protocol and the physical connections are the combination of fiber and copper cables. To limit the overheads and to isolate different applications hosted in the data center network, the virtual LANs (VLANs) used to partition the rack of servers. The redundancy used in each layer of the

data center network is established to mitigate the failures may happen in the data center.

Not only the switches and routers in the data center network, the load balancer and the firewalls is addition. The Load Balancers (LBs) connected to each Aggregation switches. Table (2.2 [4]) summarizes the link types used in the data center network.

Table 2.2: Summary of link types

Type	Description
TRUNK	connect ToRs to AggS and AggS to AccR
LB	connect load balancers to AggS
MGMT	management interfaces
CORE	connect routers (AccR, Core) in the network core
ISC	connect primary and back up switches/routers
IX	connect data centers (wide area network)

2.3.4 Virtualization in Cloud Data Centers

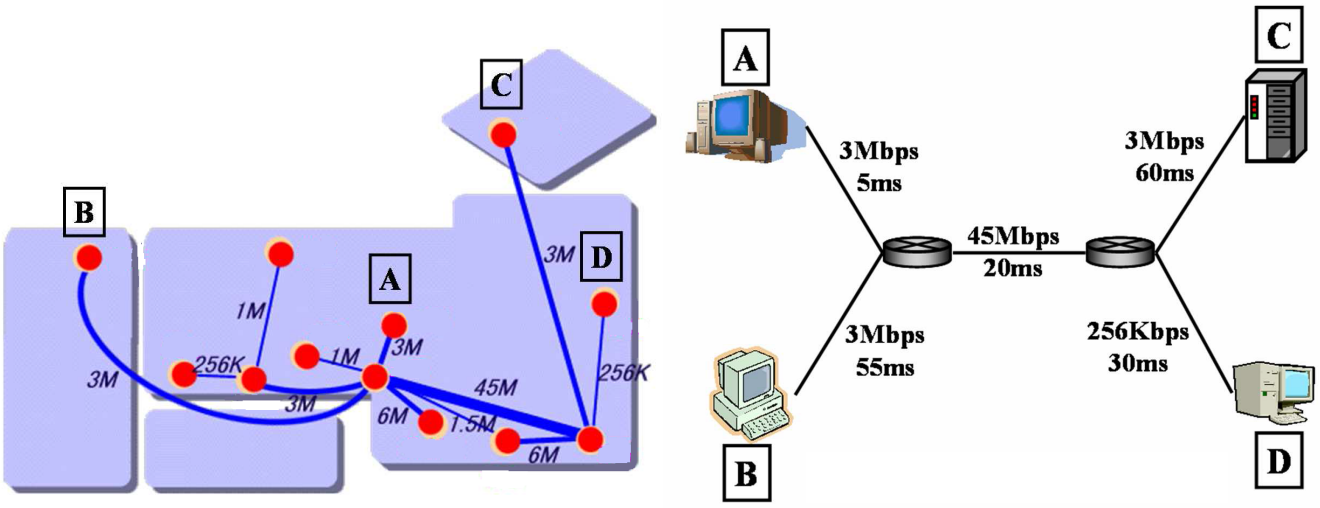
Virtualization is the process of recapitulation of the main or the primary physical structure of many of technologies like operating systems, a storage devices, hardware platform or the other network resources [17].

The main goal of virtualization is properly share the hardware between multiple applications. One of the benefits of the virtualization is that it ensure the high resource utilization and this return back with high saving in hardware, cooling and power consumption [17].

In this thesis, we need to assure SLA response time and this by testing the services hosted in cloud provider DCs. To do this in a real DC network is difficult because of the rare of real network devices, so we need to find another method to test the services. We used the simulation as an alternative to the real DCs environment for testing the services. To simulate the cloud services applications, there are many simulators used for that(e.g, CloudSim, GreenCloud,..etc). We use NS-2 to simulate our experiments. So, we need to review the NS-2 simulator and how does it work and which languages and tools it based on. We review NS-2 in the next section.

2.4 Network Simulator 2

To set up a real network topology for the purpose of doing some real experiments where it is the best way to study communications and networking even in Internet or in local network [5]. But unfortunately, setting up a real network topology is not easy and is expensive where it needs a lot of expensive devices and media. Because of the high cost and the difficulty of building real network topology to study communication and networking, the virtual network introduced by the Network Simulator is used for doing experiment of communication and networking in only one machine. Fig. 2.6a [5] shows an example of real network topology and Fig. 2.6b [5] shows an example of the same real network but in the simulation to do experiments.



(a) The Reality Network (b) The Network for Simulation

Figure 2.6: Difference between Real and Simulation Network [5]

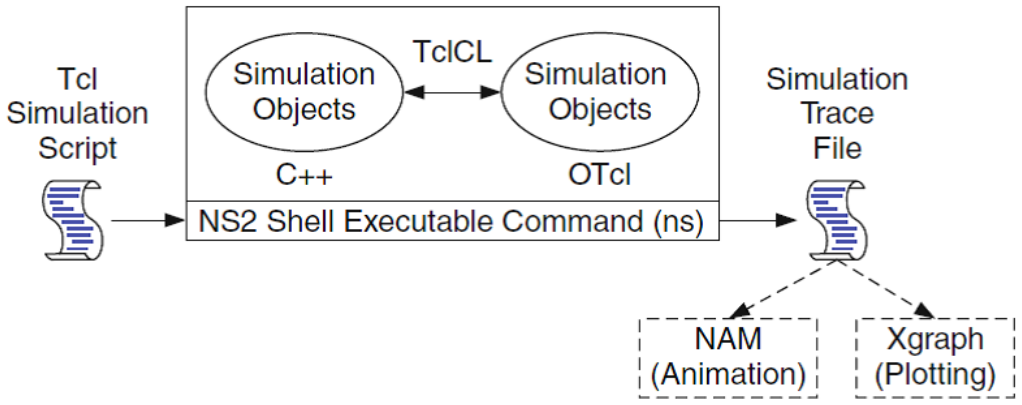


Figure 2.7: NS-2 Architecture [6]

2.4.1 NS-2 Overview

NS-2 is an event network simulator, the event is based on the packet level. NS-2 is developed under the VINT poject in the institute of UC Berkeley [6]. There are three version of NS were developed. NS-1 was developed in 1995 and NS-2 released in 1996. In 2008, NS-3 released.

We choose NS-2 to do our experiments and test the cloud applications and this because we need to move our experiments to a DCs topology in the GreenCloud[1] Simulator and the underlying platform for GreenCloud is the NS-2.

NS-2 used a scripting language called Object Oriented TCL, and it is an open source software available for Windows and Linux [6].

[1] Developed in University of Luxembourg [18]

```
-----------------------------------------------
+ 1.825127 2 3 tcp 1040 ---A--- 1 1.0 3.1 30 303
- 1.825367 2 3 cbr 210 ------- 0 0.0 3.0 203 274
r 1.82558 3 2 ack 40 ------- 1 3.1 1.0 29 302
-----------------------------------------------
```

Figure 2.8: Trace (.tr) File Example

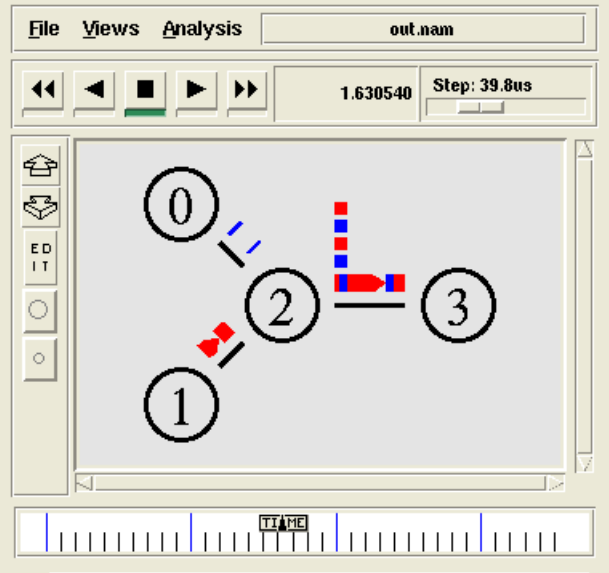


Figure 2.9: Animation Software NAM

NS-2 contains a huge number of network protocols, algorithms, applications, devised for wireless, wired and mobile networks. It can simulate the real network topology structure and the protocols used in the network structure. Also it have a packet delay or packet drop on the link characteristics.

2.4.2 NS-2: How does it Work?

Network Simulator version 2 is an easy open source and free simulator for using and it is one of the famous simulators all over the world. It uses C++ to build the network protocols classes and to link them together. C++ is like the shell for the NS-2 Simulator. It also uses TCL language for deploying scenarios for simulation. Fig. 2.7 [6] which indicate that NS-2 is an object oriented simulator, written in C++, with an OTcl interpreter as a front end. Inside the scenario file ***(.tcl)***: network topology, transmission time, protocols used for communications,...etc are defined on it. Once the scenario file is executed, the results of the simulation will be written to the trace file ***(.tr)*** and the animation file ***(.nam)***. Inside the trace file all the information regards the communication is written in the file like in Fig. 2.8. The NAM file contains the data for displaying the animation of the experiment result as in Fig. 2.9 which shows the animation during the display.

To write a scenario in NS-2, we need to understand what happen in the real network and what happen in the network inside the NS-2. Without any explanations the following two figures explain the different between real network and network in NS-2. Fig. 2.10 [5] shows the realistic network where the node have 4 layers using TCP/IP model. The 4

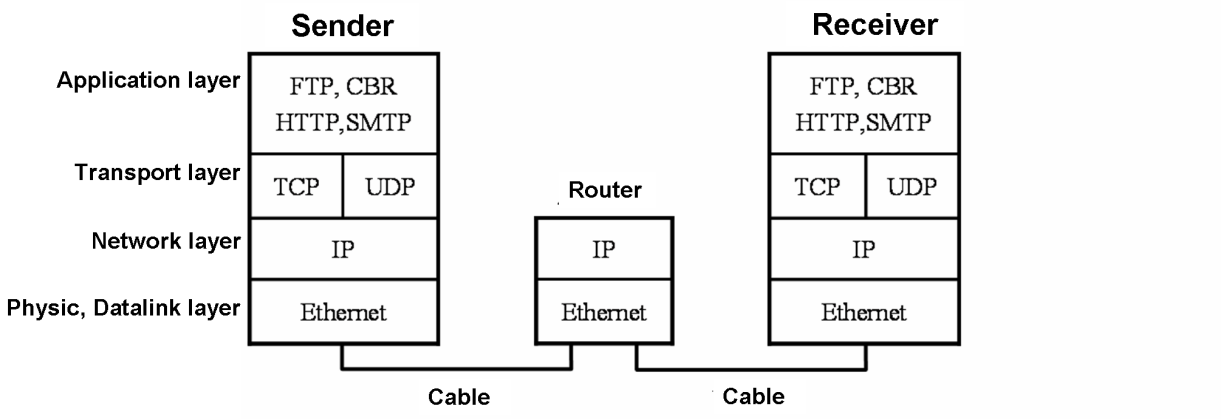


Figure 2.10: Realistic Network [5]

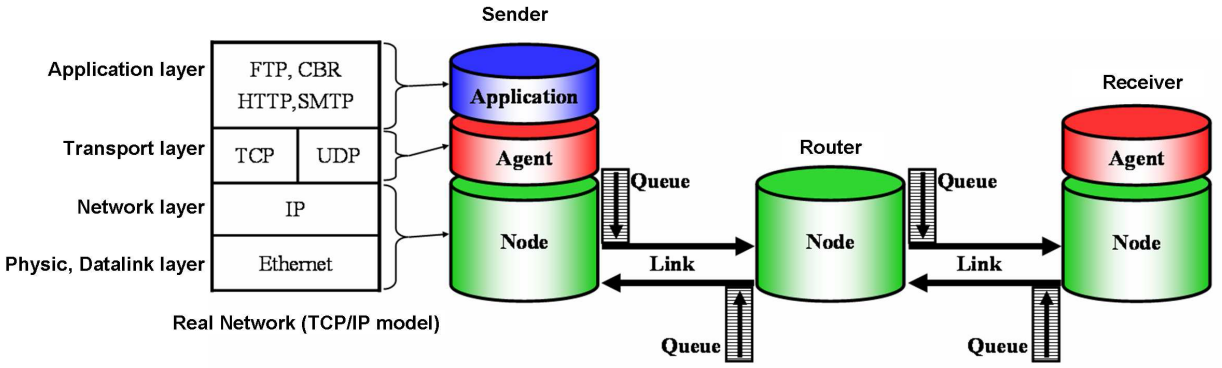


Figure 2.11: Network Inside NS-2 [5]

layers of TCP/IP of the node in NS-2 is shown in Fig. 2.11 [5] .

After producing simulation results, we can analysis the simulation results in the trace file by using Perl[1] or AWK[2] scripting languages to get some important analysis values like response time, total time for the whole message, end to end delay, throughput, latency, jitter, ...etc. Each value from the previous values is depend on what we are simulate in the TCL file. We can see these results also in the NAM tool as in Fig. 2.9.

In This master thesis, we used the TCL scripts to build my experiment topology. In case we want to change the protocols classes or add new agents, we used the C++ language. To analysis the trace file, we used the Perl script language for that. Then we plot the results in R language.

[1]The Perl Programming Language: www.perl.org
[2]The GNU Awk Programming Language : https://www.gnu.org/software/gawk/manual/gawk.html

2.4.3 Tool Command Language

TCL is a powerful scripting language which easy to understand, it is a dynamic programming language [19]. It is very good for wide range of uses like web and desktop applications, networking, administration, testing and many more.

TCL is an open source language. It is used in the NS-2 simulation to build the topology and the scenarios wanted to simulated and tested. Also used to build agents and applications on the nodes in the network topology.

In addition to the previous overview about CC, SLA, CDCs and NS-2 where they are the main topics this thesis work based on, we will review the related work relevant to those topics in the next chapter. We need to review the previous related work to see how the previous work going on and see the similar work in SLA assurance.

Chapter III

Related Work

"If you know your enemies and know yourself, you will not be imperiled in a hundred battles... if you do not know your enemies nor yourself, you will be imperiled in every single battle."

Sun Tzu - The Art of War

In this chapter, we provide in Section 3.1 an overview of some of the works aiming at preventing The Service Level Agreement violations. Then we provide an overview of the related work about the cloud data centers in Section 3.2. And in Section 3.3, we present a research review about the cloud computing applications. Finally, reviewing the research work related to the cloud simulation and simulators are in Section 3.4.

3.1 Service Level Agreements

In this section, we will review the research has been done in SLAs. There is a significant research in SLAs which preformed with a good effort. We overview the existing research work in managing and establishing SLAs in the cloud computing society.

The authors in [2] propose a mechanism for managing SLAs in the cloud computing environment by using the web service level agreement framework. They developed it to monitor and enforcement SLA where all these in a SOA[1].

SLA Violations

Arpan Roy, Rajeshwari Ganesan and Santonu Sarkar in [7] introduce the KIM software framework as a cloud controller which aids in minimizing service failures which happen because of the SLA violations. SLA violations are like violation of utilization, availability and response time in the SaaS cloud data centers. They use the migration as the primary mitigation technique and also they tried to mitigate the SLA violations without using any migration. The paper [7], seems to be the closest work to the current work in this master thesis paper. In [7], they developed a new system as in Fig. 3.1 and we just simulate the failures and solve it in the simulation and also simulate the failures in a real scenarios but we couldn't solve it in the real scenarios because of the lack of resources and this is the difference between us and the KIM system. Wood *et al.* [20] introduced the Sandpiper system for automated mitigation of increasing response time of the host

[1]Service Oriented Architecture: is a technique supports the combination between loosely coupled services where each service is independent functionality.

Physical Machines and utilization in a virtualized data center due to workloads. The migration in Sandpiper system is only depend on the increasing of utilization threshold and not depend on SLA failures or violations. Shen *et al.* [21] introduced the CloudScale system that make a proactive forecasting for the upcoming SLA violation, they also did a dynamic resource allocation in addition to the migration of the work load. CloudScale system uses the Markov Chain based state space approach to do the prediction process. In [22–24], they suggest an approach to allocate VMs[1] as possible to the given server without any violations to the VMs SLAs. Last but not least, Salman in [15] provide a study which states that there is no any cloud provider offers any guarantees for the performance of computer services and leave the customer detect the SLA violation. Salman also provide a guidelines on how should be the SLAs defined for the future cloud services.

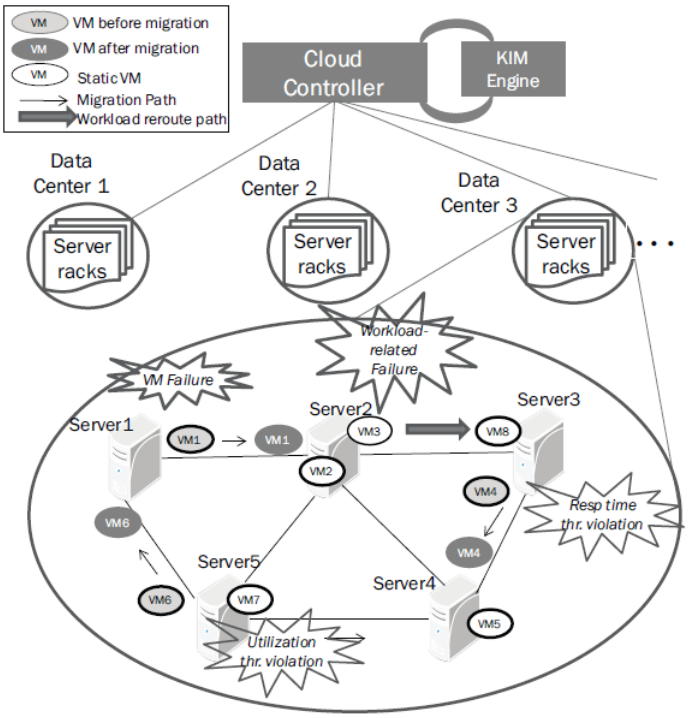


Figure 3.1: SaaS workload dynamics with KIM software framework [7]

3.2 Cloud Data Centers

In this section, we will review the significant level of research in CDCs. There is a significant research in CDCs which preformed with a good effort. But the research work have been done related to the cloud data center is a hug amount and has numerous branches. Like Load Balancing, Storage Migration, VM Migration, Job Scheduling, Power Management, Energy Saving, Traffic Management, Network Architecture, Switching& Routing, Fault Tolerance, Map Reduce, Resource Management, HPC, ...etc. We will just focus on the related topics with our work like Data center Failures Network Architecture of data centers in Subsection 3.2.1 , Failure Mitigation techniques in Subsection 3.2.2 and VM Migration in Subsection 3.2.3.

[1] Virtual Machines

3.2.1 Data Center Failures

Kashif Bilal, Saif Ur Rehman Malik, and Samee U. Khan in [17] discuss the challenge and trends in the cloud data centers. This after review the cloud data centers architecture and the structure of the cloud data center network. Finally, they review the virtualization in the cloud data centers. Ranjithprabhu *et al.* in [9] proposed a technique to eliminate the single point of failure in the cloud data centers by elaborating the redundancy techniques. They also did data replication by using data mapping to prevent data losing in cloud computing. Phillipa Gill *et al.* in [4] answered some questions to understand the network failures in cloud data centers by measuring the impact of the failures then analyzing the measurements and finally discussed the implications of the failures. They also introduced a brief discussion about the cloud data centers network structure. Our work in this thesis used the network failures mentioned in the paper [4] and we going to discuss them in the incoming chapters where the network failures is one category from the cloud data centers failures.

When we talk about the network failures, there is two level of failures. The first level is the application failures and the second is the network connectivity failures. The previous research work related to the application level failures are in [25,26] . The previous research work related to the network connectivity level failures are in [27–32]. For the application failures, Padmanabhan *et al.* in [26] consider the failures faced by the web client. They explain the failures which happen during the TCP[1] end to end connectivity session. Also related research work to the cloud data centers failures is the cloud computing failures previous research work. Because of the increasing of interest in cloud computing, the understanding of failures and how to mitigate it became an interesting and important trend of research. The previous research work was focused on the DRAM[2] as in [33], the storage as in [34,35] and server nodes failures as in [36]. Ford *et al.* in [34] discussed the availability of distributed storage and explain the failures of storage nodes which localized in one single rack. Kashi *et al.* in [32] characterized the physical server failures in the big cloud data centers. They classify the physical server failures to percentages classes in a study of 100 servers failures.

Greenberg *et al.* in [8] propose a joint optimization of data center resources and network. They also propose a new mechanizes and systems for geo-distributing state. They also discuss all the failures faced in the cloud data centers like the physical server, network, power draw and infrastructure as well the cost for each one respectively. Cloud data centers also have another category of failure that is not focused too much on it in this thesis which is the network failures inside the data centers. Sriram Sankar and Sudhanva Gurumurthi in [10] discussed the soft failures in the large data centers. They verify that the availability of the services is affected by the soft failures where data center failures are important trend in large data centers. This because software failures is not required any hardware replacement however it produce in service down time and disrupt the normal service operations. Finally, Rahul *et al.* in [37] study the network failures in the data centers and their impact on the cloud services.

[1] Transmission Control Protocol: is one of the reliable transport layer protocol in the OSI and TCP/IP models.

[2] Dynamic Random Access Memory: is a type of Random Access Memory which store the data bit in a separate capacitor inside the integrated circuits.

3.2.2 Mitigation Techniques

After the cloud data center failures happen in the data center, we need to solve the failures or at least mitigate the the failure because some situations we could not solve the failure directly and the mitigation is needed in those situations. The mitigation techniques are many and they depend on the failure type. The famous technique of mitigation is the redundancy in the data center.

The mitigation of failure is divided into two stages, the first is the failure detection and diagnosis. The second one is the failure recovery. There are not too much related research papers in the failures mitigation techniques. But there some recent related work in the detection and diagnosis stage of the failure mitigation. Banerjee *et al.* in [38] said that some systems detect and identify the failures in the IP network by using the active probes. But Kandula *et al.* in [39] and Kompella *et al.* in [40] detect diagnosis the failure by taking measurement data from the network and the end hosts then they built a probabilistic models to identify the component location which cause failure. Recent work is interested in deploying systems that help to get any failure cause any to the performance of the applications whatever the failure is hardware based or software based [41, 42]. There also a big research work related to the distributed systems failure diagnosis in [43–45]. In our work of this thesis we introduce the failure to test the cloud applications, so we know what exactly the failure and it's diagnosis.

The second level of the failure mitigation is the failure recovery. The failure recovery in data center is not new and have many related research work. Isard in [46] propose a management system for automated servers. The system name is Autopilot. Autopilot system is based on the concept of recovery oriented computing. When The Autopilot system detect a failure, it react with one possible recovery action from three. The three recovery actions are restart, reimage, or RMA[1] . Wang *et al.* in [47] propose a rapid recovery service that can quickly recover and mitigate link failures by pre-computing all the possible updates link for each link in case of failure and this computing is in the forward table. The name of this rapid recovery service is R3. Kiran *et al.* in [48] propose a System Support for Automated Availability Management which called Total Recall. Total Recall is distributed system for storage that has a convenience amount of redundancy to indemnity for host availability changes. Finally, Xin *et al.* in [49] proposed an automated system called NetPilot. NetPilot goal is to mitigate or solve the data center network failures rapidly.

3.2.3 Virtual Machine Migration

VM migration is one of the famous failure mitigation techniques where it represent the redundancy in the cloud data center. It is not only used as failure mitigation but also used to improve and increase the utilization of the physical machines inside the cloud data centers. VM migration is also giving the chance to the data center to make re-balancing the workloads across the physical machines, and this is the promise of maximizing the utilization of the physical machines. We will review the related research work about the virtual machine migrations in the cloud data centers. Most of the previous work is in the following papers [20, 50–56].

[1]Return Merchandise Authorization or return authorization (RA) or return goods authorization (RGA): is the part of a returning process of a product (device,..etc) for repairing.

Clark *et al.* in [50] said that the VM live migration is used to improve server side performance from the side of the physical resources and power consumption [20]. The work in [51] uses the dynamic VM consolidation to decrease the number of working physical machines in data centers. Verma *et al.* in [52] discussed the important issues between the physical resources (e.g. physical machine, switches, routers, ...etc) utilization and the power consumption in the cloud data center. They analyzed the work load of the applications and made consolidation for power saving. But they didn't take traffic and link load into account during doing the VM consolidation and migration in data centers.

In the following proposals [53, 55] for VM migration by considering network traffic among virtual machines, but they only consider the total sent data between virtual machine and the distance between physical machines during doing migration. Jun *et al.* in [54] propose a new automatic VM migration system in the data center that can detect hotspots (e.g. physical host over-loaded and network congestion) and dynamically remap VMs for the purpose of improving the network performance. The name of the new automatic VM migration system is MWLAN (Migration With Link And Node load consideration). The VM migration approach proposed in MWLAN can perfectly balance the load of the network link and reduce the local data center network congestion. MWLAN migration system considers the traffic among VMs and also the link traffic load of data centers.

Fung *et al.* in [55] introduced a new scheme called AppAware. AppAware is a noval computational scheme for enrollment inter-VM dependencies and the basic network topology into VM migration decisions. They used the simulation to prove that their proposed computational scheme decreases network traffic by up to 81 % when compare it with a well known alternative VM migration method where the alternative VM migration method is not application-aware. Finally, Fung *et al.* in [56] designed and implemented a scalable live VM migration scheme called S-CORE. S-CORE is designed to dynamically reallocate VMs to servers during minimizing the overall communication effect of the flows of the active traffic. They compared S-CORE with diverse aggregate load and coordination policies. They got a result that S-CORE can achieve about 87% in the communication cost reduction with fixed number of migration rounds.

3.3 Cloud Applications

CLoud applications are the services provided by the cloud providers and hosted in DCs. We need to review the related work with cloud applications because we will test them in our experiments to assure SLA. In this section, we review these related work.

Cloud applications became a big trend in research work due to the daily increasing of the users who use it world wide. Cloud applications represents the SaaS[1] layer in the cloud computing. There are many types of cloud applications, we will classify them into classes in the next chapters. In the real Experiment chapter, we will also discuss mathematical models for testing the cloud applications we have designed it. In this section we just review the related research work about cloud applications in Subsection 3.3.1 and then the research work related to the previous models proposed for testing different cloud applications in Subsection 3.3.2.

[1]Software as a Service: One of the main cloud computing services provided to the cloud customers

3.3.1 Cloud Computing Applications

There are many research work related to the cloud computing applications. Ali *et al.* in [57] gave an overview of the cloud computing applications and discussed the characteristics, traits and issues of the cloud computing applications. Puja in [58] discussed the cloud computing and the software as a service applications. He explained the concept, Services provided by cloud computing and different service providers.

Because of diversity of the cloud applications, we will review the related work for each type of the cloud applications. In the web applications, Eljona *et al.* in [59] discussed the general concepts, tools and practices used for testing the performance of the web applications. In the massive data analysis application, Weiyi *et al.* in [60] proposed an approach for explaining differences between pseudo and large-scale cloud deployments and this to assist developers of BDA[1] Apps for cloud deployments.

In the real time applications, Arthur *et al.* in [61] proposed a framework for deploying distributed real time systems and applications in cloud. Their conceptual framework enable the developers of the distributed real time applications to forecast and handle the issues early during the designing phase of the systems. The gave a good practice for evaluating the distributed real time systems. Qiang *et al.* in [62] developed a system interface called PROTEUS. PROTEUS is developed to gather information about the network performance like throughput, loss, and one-way delay and then forecast the future performance of the network by using regression tree. They used it to do network performance forecast for Real-Time, Interactive Mobile Applications. David *et al.* in [63] supported the real time applications in the integrated services packet network (ISPN). They reviewed the characteristics of the real time applications and then the requirements of the real time applications. Finally, they proposed an ISPN architecture that support two different real time services.

In the mobile cloud computing applications, Hoang *et al.* in [64] gave a detailed survey about mobile cloud computing applications. The survey help the general readers to get some information about the mobile cloud computing like the definition, applications and architecture. They also introduced the issues related to the mobile cloud applications in addition to the existing solutions and approaches as well as the future research direction related to the mobile cloud computing applications [65]. Fu *et al.* in [66] proposed a policy for efficient energy consumption in mobile cloud computing. They proposed a framework based on the properties of the mobile cloud application to minimize the energy consumption.

In the high performance cloud computing applications, Abhishek *et al.* in [67] answered some questions related to the high performance computing in the cloud like why and who should select (or not select) cloud for HPC[2], for what applications, and how should cloud be used for HPC? They performed an evaluation of the performance for the HPC applications hosted in different platforms from supercomputers to commodity clusters, even if in-house and in the cloud. Also, Keith *et al.* provided a performance analysis for the high performance computing application but now in the Amazone[3] web services cloud [68].

[1] Big Data Analytics Applications
[2] High Performance Computing, like UL HPC Cluster
[3] http://aws.amazon.com/

In the highly interactive applications, Sumeer *et al.* in [69] propose an approach to design a work load model for the high interactive applications. Mohamed *et al.* in [70] propose a novel SLA-Aware for rearranging I/O[1] requests of database transactions and for optimizing I/O-intensive transactions in the highly interactive applications.

In the distributed applications on the cloud, Muhammad *et al.* in [71] reviewed the distributed applications processing frameworks in the smart mobile devices and this under the trend of mobile cloud computing. Antonis *et al.* in [72] presented an architecture for evaluating, collecting, modeling and evaluating of distributed application deployment in the multi-clouds. Jan *et al.* in [73] presented four different strategies for deploying on the infrastructure of the cloud computing. Stanisław *et al.* in [74] presented the method of constructing the real-time applications for the IaaS model of cloud computing. Xavier *et al.* in [75] proposed an application model for introducing any type of distributed application which consisted of a set of interconnected virtual machines. The deployment of the distributed application in the cloud is included a protocol for self-configuring the virtual application machines. The verification of the self-configuration protocol is done by Gwen *et al.* in [76].

3.3.2 Testing Models

The previous research work related to the testing models for the cloud applications are in the following research papers [69, 77–82]. Generally, W.K. Chan *et al.* in [77] overview the modeling and testing for the cloud computing applications. They present a set of procedures to setup cloud computation and model-based testing to do testing for the cloud computing applications. Jerry *et al.* in [78] provided a clear tutorial on cloud testing and cloud-based application testing. They provided clear concepts, discusses the special objectives, features, requirements, and needs in cloud testing. They also provided a clear comparison between web-based software testing and cloud-based application testing. This comparison discussed the important issues, needs and challenges in testing cloud-based software applications.

The previous models-based for testing web applications in the following papers [79, 83–86]. Hassan *et al* in [79] discussed a model based software testing way for testing the web applications by using the StateCharts. Sebastian *et al.* in [83] improved the method of testing the web applications by using the user session data. Nazish *et al.* [84] introduced a model based testing to find errors in web applications. Juhan *et al.* in [85] introduce a model-based testing of web applications by using NModel[2]. They used web-based positioning system called WorkForce Management as a case study in their work. Anneliese *et al.* in [86] propose a model for testing web application with using Finite State Machine[3]. In our work we propose a mathematical model based for testing the web applications, it based on a real experiments. We will discuss it in the incoming chapters.

The previous model-based for testing real time applications, highly interactive applications and mobile cloud applications in the following papers [69, 80, 81]. Saddek *et*

[1] Input and Output.

[2] NModel: *"is a lightweight toolkit based on C# that can be used across different platforms for establishing such a domain-specific testing application [85]"*. Available from http://nmodel.codeplex.com

[3] *"Finite state machines provide a convenient way to model software behavior in a way that avoids issues associated with the implementation. [86]"*

al. in [80] proposed a novel method for testing of real-time applications in general and robotic applications in particular. Chuanqi *et al.* in [81] provided an approach to modeling mobile applications for testing environments based on a Mobile Test Environment Se-mantic Tree (MTEST). Rémi *et al.* in [87] proposed a model based tool for interactive prototyping of highly interactive applications. Also in [69] they proposed a workload model for highly interactive applications. In our work we didn't get a testing model for testing the mobile applications and the highly interactive applications, but for the real time we introduce a model based on the real experiments.

The previous model-based for testing distributed applications in the following papers [82, 88, 89]. Diwaker *et al.* in [88] propose an approach called DieCast. DieCast used to test the distributed systems by using an accurate scale model. Huey *et al.* in [82] presented a statistics based integrated test environment called SITE. SITE is developed for testing distributed applications. SITE provided the support automatically, like test development, test execution, test failure analysis, test management, test measurement and test planning. Giovanni *et al.* in [89] propose a method of testing the performance of the distributed software applications in an early stage in the development. They derived the testing from architecture designs so that the distributed application performance can be tested by using the middle-ware software at early stage of the development process. In our work we propose a mathematical model based for testing the distributed applications, it based on a real experiments. We will discuss it in the incoming chapters.

3.4 Cloud Simulation

Cloud computing is became a very big trend of scientific and business IT research. Because of that there are a lot of simulation tools and platforms deployed to accommodate the research experiments in the cloud computing and data centers of the cloud. In addition to the booming of research in cloud computing, the expensive physical resources used to do real life experiments in cloud computing let the companies, universities and research centers to develop the simulation platforms for generally network experiments and specially for the cloud computing where it is belong to the network and communication research. But now they are developing simulation tools especially for cloud computing. All the cloud computing simulators details are collected in the survey introduced by Ahmed *et al.* in [90]. In addition to the detailed survey about the cloud computing simulators, they also provide the future directions for the cloud computing simulators.

There are many of cloud computing simulators mentioned in [18, 91–101]. Calheiros *et al.* in [91] introduced the CloudSim simulator for cloud computing. CloudSim is the famous simulator tool available for the environment of cloud computing, it is an event driven simulator. Kliazovich *et al.* in [18] introduced the GreenCloud simulator which based on Network Simulator 2. GreenCloud is a packet level simulator, it was developed to estimate the energy consumption in the cloud data centers, and try to reduce the energy consumption inside the cloud data centers. Núñez *et al.* in [92] provided the iCanCloud simulator. iCanCloud was developed by taking into account the drawbacks of the previous cloud computing simulators like CloudSim,GreenCloud and MDCsim [94]. Wickremasinghe *et al.* in [93] developed a novel cloud simulator called CloudAnalyst. CloudAnalyst is based on the CloudSim simulator. CloudAnalyst is developed for performance evaluating and estimating the cost of large-scale geographically distributed cloud system which having a huge number of user workload based on different

parameters. There are other cloud simulators developed for cloud computing simulation like MDCSim [94], NetworkCloudSim [95], EMUSIM [96], GroudSim [97], DCSim [98], MR-CloudSim [99], SmartSim [100] and SimIC [101]. But each one of those previous simulators was developed for a specific purpose which is needed to be simulated in the cloud computing.

Table (3.1) which from [90] summarizes all the previous cloud computing simulators and a very small description for each one like the underlying platform, cost modeling, graphical user interface,...etc. In this thesis work, we used the underlying platform of the GreenCloud simulator which is the NS-2 simulator [6]. GreenCloud simulator is developed in University of Luxembourg [18]. We used just NS-2 platform in GreenCloud because GreenCloud simulator is too big for our work in this master thesis and we didn't focus on the energy consumption like in the GreenCloud. We just focus on the network of the data center and the physical devices. GreenCloud have big DC topology, we will move our work to it in the future work.

Table 3.1: Comparison between Cloud Simulators

Simulator	Underlying Platform	Open Source?	Programming Language	Cost Modeling	GUI	Communication Model	Simulation Time	Energy Model	Federation Policy
CloudSim	SimJava	Yes	Java	yes	No	Limited	second	yes	yes
Cloud Analyst	CloudSim	Yes	Java	yes	yes	Limited	second	yes	yes
Green Cloud	NS-2	Yes	C++, otcel	No	Limited	Full	Minute	yes	no
MDCsim	CSIM	Commercial	JAVA/C++	No	No	Limited	second	Rough	no
iCanCloud	SIMCAN	Yes	C++	yes	yes	Full	second	No	no
Network CloudSim	CloudSim	Yes	Java	yes	No	Full	second	yes	yes
EMUSIM	CloudSim, AEF	Yes	Java	yes	No	Limited	second	yes	no
GroudSim	-	Yes	Java	yes	Limited	No	second	No	no
MRCloud Sim	CloudSim	No	Java	yes	No	Limited	-	yes	yes
DCSim	-	Yes	Java	yes	No	No	Minute	No	no
SimIC	SimJava	No	Java	yes	No	Limited	second	Rough	yes

We will describe this thesis problem in more details and also provide the methodologies we use to do our experiments and test cloud applications in the next chapter.

Chapter IV

Problem Description

"Any man who must say, I am the king, is no true king "

Tywin Lannister - Game of Thrones

In this chapter, we provide a clear definition of the problem we tackle in this thesis. We illustrate this problem in Section 4.1 to give an intuition of the problem. We then precisely define the problem in Section 4.2. We close this chapter by a discussion on the methodology we are going to use to achieve our experimental work in Section 4.3.

4.1 Intuitive Approach

We are interested in this thesis, to provide an assurance to the Service Level Agreement in the cloud computing DCs. This can be done by provide an assessment to the cloud applications hosted in the data centers. The assessment of the cloud applications will be done in situations of failure to see what if application performance is affected by the failures or it can continue work during the failure. The performance of the application is affected by the failure, but it is still working with low performance. In the other hand, there are some applications which can't work with low performance. After seeing the failures, we solved the failures or at least mitigated them. The mitigation of the failures let the cloud application still running in case of failure even if with performance degradation. By solving the failures in the cloud data centers, we can now take a big assessment picture to the performance of the cloud applications hosted in the cloud data centers. This big assessment picture let us in this master thesis assure the SLA response time (QoS) between the cloud customers and the cloud applications hosted in the cloud data centers providers. Also let us see if there is any violations to SLAs in the cloud data centers.

4.2 Problem Statement

SLA contract between cloud providers and cloud customers covers two metrics, the uptime which is the availability of the services provided by cloud provider. The second metric is the response times for the services provided and if any critical failure happen, this metric represent the performance of the services and the QoS to the services provided to the cloud customers. In this thesis we need to assure SLA response times, in other words, we need to assure the performance of the services and guarantee a high QoS all the time as the cloud providers mention in the SLA contract. The cloud customers

can not verify on SLA metrics, so we need to evaluate these agreement between cloud providers and cloud customers to provide solutions to cloud providers if failure exist and provide an assurance of SLA QoS to cloud customer if services performance really working good all the time even if there are failures in DCs where the cloud services are hosted.

To assure SLA services performance, we need to do performance evaluation to the cloud services which are hosted on CDCs. The performance evaluation should be done in the environment of failures to see the behavior of services and then we solve failures and evaluate the performance again. With solving and mitigating failures we can ensure the SLA QoS, services performance and the fast response in case of any critical failures may occur. Cloud services are the cloud applications (SaaS). To proceed in our experiments and testing, we need to identify and classify cloud applications and failures may occur inside DCs and harm the performance of applications in addition to the solutions of the failures. Then we can test the applications performance with failures and solve the failures to assure SLA services performance.

The methodologies we follow them in this thesis work are in the next section and what is the main components of our problems such as the input , output and processing. Also the procedure to assure SLA from violations in QoS.

4.3 Methodology

Our problem takes as input the important metrics for each class of application and the different failure of the cloud data center and check what is the effect of the failures on those metrics. In the processing stage we try to test the cloud application classes in the real applications scenarios. We could do the real experiments four classes of cloud applications. We move our experiments to the simulation of applications scenarios in NS-2 to increase the number of classes we tested in this thesis work. Finally, the output of our problem is the assessment of the performance of the cloud applications by testing the metrics for each class after introducing the failures. Then the solution for the failures and the mitigation techniques for the failure are the final output of our work. Finally, by the two previous outputs we can assessment SLA (QoS) and then assure that there is no violations to the SLA response time and services performance between the cloud customers and cloud Data centers. Fig. 4.1 describes the problem input, processing and output, in addition to the methodology we follow to proceed in our work.

The implementation in thesis was in two methodologies. The first one is the real applications scenarios. The second is the NS-2.

4.3.1 Real Scenario

In the real scenario experiments, we developed different types of applications with PHP[1] and setup them locally in one machine, then we introduce failures by using a proxy tool called Charles[2]. For example, for the web we design a web application and setup it on a local web server, and the same scenario for file application, distributed application and real time application. Through the Charles tool and after introducing errors, we can get some metrics from the connection such as the delay time, response speed and

[1]WEB Programming Language
[2]Charles Version 3.9.3; WEB DESIGNING PROXY; charlesproxy.com

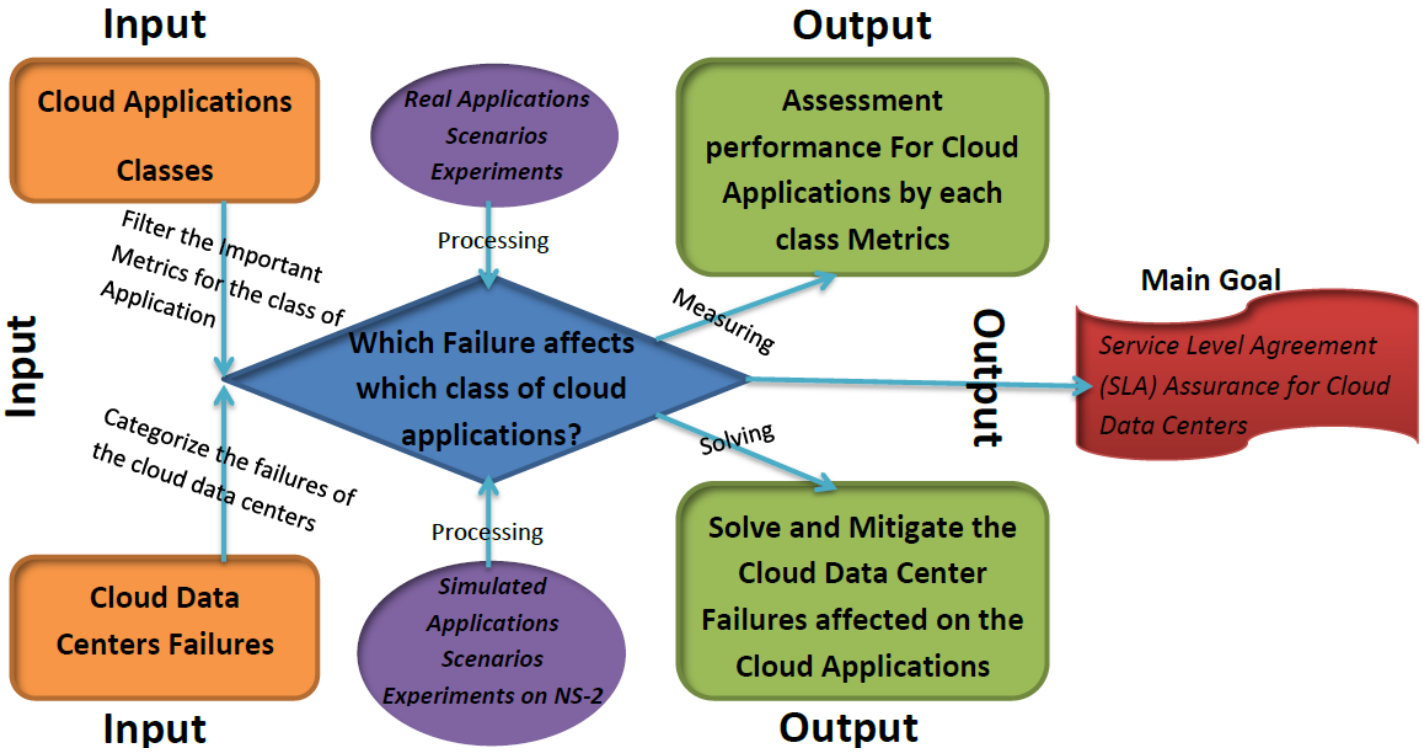


Figure 4.1: Problem Statement & Methodologies of this Thesis

time, latency,..etc. Fig. 4.2 shows how we introduce failures in the network, such as the degradation of the network bandwidth. Also Fig. 4.3 shows the overview results we can get from Charles after doing the experiment, like the response speed,...etc.

4.3.2 Simulator

In the Simulator experiments, we used the NS-2 to simulate our experiments. In the simulation experiments we can simulate most of the classes of the cloud applications. We can introduce failures and errors in the topology of the experiment. We developed a TCL file for each class of applications and run it in NS-2. From the output file we calculate the values of the metrics by using Perl and also introduce different types of failures. By using the NAM tool we can see the animation of the experiment and what happen exactly in the scenario. In the simulation methodology we simulate more classes of applications than in the real experiments. Finally after simulating and getting results with and without failures, we solve the failures in the simulation files by TCL also and some C++. We simulate the mitigation techniques in NS-2 to see the performance of the application classes and see how the failures mitigation increase the SLA QoS again after solving failures.

We will start with classifying CC applications in the next chapter. We will review the metrics for each class of applications which we will use them during testing and simulations.

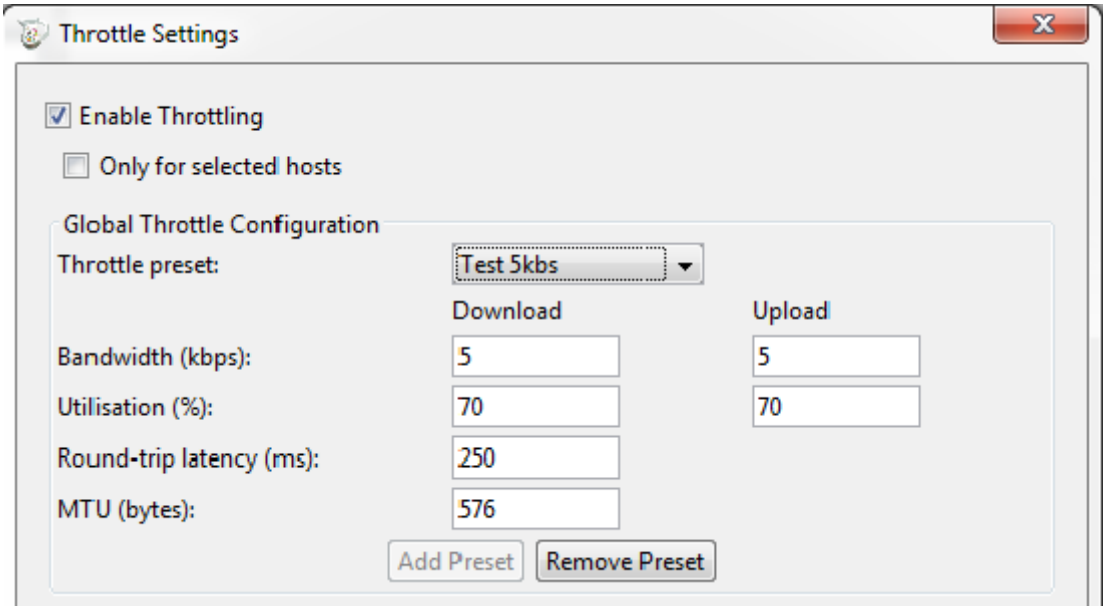


Figure 4.2: Introducing Failures in Bandwidth by Charles

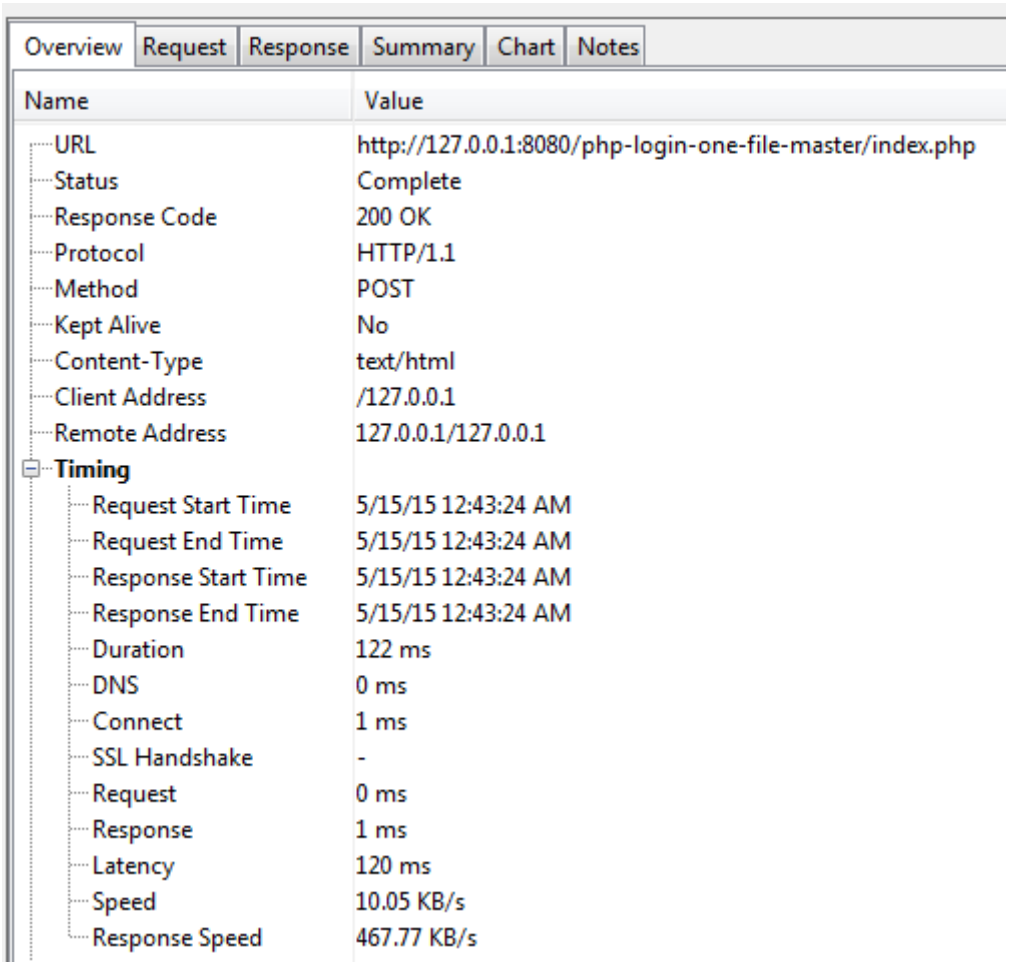


Figure 4.3: Results of Real Scenario Experiment from Charles

Chapter V

Cloud Applications Classification

"Make sure you visualize what you really want, not what someone else wants for you."

Jerry Gillies

In this chapter, the first contribution in this master thesis which is the classification of the cloud computing applications. We will discuss the different cloud applications and divide them into different classes in Section 5.1, also we will discuss the important metrics for each class of the cloud applications in the same section. In Section 5.2, we will do a new classification for the classes of the cloud applications that is based on the sensitivity of the failures.

5.1 Classes of Cloud Computing Applications

After looking for the applications hosted in the cloud data centers, we found a lot of cloud applications used by the cloud users. In this section, we will review these applications after dividing them into classes. Then we review the important metrics of the performance for each class of the applications.

We classify approximately all the cloud computing applications into eight main classes. Each class of the eight classes of the cloud applications has different properties, different way of working and different metrics related to the performance of the application. There are some metrics are common in some classes of the cloud applications. We will discuss those eight classes of cloud applications by defining it and give some real used cloud applications in this class in addition to the the important metrics related to the performance of the class of cloud application. Also the network and transport protocols used for each class of applications. The eight classes of cloud applications are as the following: class one is the Highly Interactive cloud Applications and it is discussed in the Subsection 5.1.1. In Subsection 5.1.2, we discuss the second class of applications which called the Web Applications hosted on the cloud. The third class is the File Applications hosted in the cloud data center servers (storage) and it is discussed in the Subsection 5.1.3. In Subsection 5.1.4, we discuss the fourth class of applications which called the Real Time cloud Applications. The fifth class is the High Performance cloud Computing Applications and it is discussed in the Subsection 5.1.5. In Subsection 5.1.6, we discuss the sixth class of applications which called the Mobile Cloud Computing Applications. The seventh class is the Massive Data Analysis Applications hosted in the cloud and it is discussed in the Subsection 5.1.7. Finally, the eighth class is the Distributed applications

which is in the last Subsection 5.1.8.

Table (5.1), summaries the classification and the identification of each class of applications and also examples of applications. In addition to the protocols used for testing and simulation, it contains the tools are used in the real testing applications.

5.1.1 Highly Interactive Applications

The class of HIA is a class of such applications that is a synchronous group ware such as distributed virtual environments, multi-player games and collaborative design. It allows and permits the distributed users to interact remotely through the cloud with each other users at the same time. This class of application allows the human cloud users to deal and interact with the applications directly by using the manipulation input devices such as mice, joy sticks, body position sensors,..etc [69]. This class of application such as gaming, interactive visualization and multimedia are becoming a great and important application installed on the cloud customers desktop and mobile devices. These applications have a limited window of time which is the frame and this to usually update the simulates universal state in addition to update the visual execution that based on the user inputs received immediately, and all these are because of the interactive nature of those applications [102].

Virtual Reality[1] [69] is a similar but different class, which is working with single or multi-user. In virtual reality application, the high output rendering and high computations are needed. Also the hardware resources management like the processors and memories and some of which could be distributed.

The transport protocols used in HIA are the Interactive Real-time protocol and Real time protocol. We will use the IRTP to simulate the HIA in the NS-2 simulator. RTP and IRTP protocols are working as an upper layer for UDP protocol. Also IRTP take the advantages of TCP protocol.

Examples of the HIA in the cloud such as gaming applications whatever on the mobile devices or in the personal computers, virtual reality applications, query response applications and office productivity applications like Google Docs online[2] and online office[3]. The important metrics for the HIA class of applications will be discussed in the next sub-subsection.

Important Metrics for HIA

HIAs need the rapid response time although the scarcity of resources. The resources are the data shared where it needs to be managed and this to provide a good responsiveness. Responsiveness is represent the first important metric for HIA. It means both the time used to receive a response from an action on the data by the same user (This is called the Response Time (RT)) and the effect of that action can be seen by a different user (This is called the User to User Time (UUT)). The second important metric is the scaling of the shared data, where in this class of applications the users could be widely distributed and may be there are a large number of users. This scaling needs the more memory (RAM[4])

[1] VR: is a computer simulated for the real world life. See the link: http://www.vrs.org.uk/
[2] https://www.google.com/docs/about/
[3] https://office.live.com/start/default.aspx
[4] Random Access Memory

space to be high available and scalability. All these metrics in addition to latency and throughput values.

In the highly interactive database applications [70], optimizing the input and output transactions is important and became an important metric specially in the database management systems. HIAs like web applications based on a database or a data warehouse have to meet the user's expectations like the high performance because this application support a huge number of data access. The users feedback and satisfactions or their experience determine the extent of success of HIAs and this keep the competitor of that applications away.

The important metrics for the HIAs and which we will focus on some of them in this thesis work are summarized as the following:

1. Response Time (RT).
2. User to User Time (UUT).
3. Scaling of the main memory size RAM and the processing power.
4. Input and Output intensive transactions speed.

If the previous metrics are violated during HIA is hosted in the cloud data center, the performance of the HIA will be degraded. Also the QoS and response times mentioned in SLA will be harmed even the service performance. In our work, after introducing the data center failures we will check on the previous metrics to see what is the affect of the failures on the HIAs performance and the service performance. Then we solve failures to assure the high QoS as in SLA.

5.1.2 Web Applications

WA is the most used cloud application ever. Most of the cloud users use every day web application to buy something, rent a car, reserve hotel or sale some products. Web application is like any web page we can browse on the Internet. The construction of the web application is constructed by the web server which is one of the servers inside the cloud data centers and the web client who is the cloud user. Cloud user used the HTTP to send a GET request to browse a page hosted on the web server inside the cloud data center. Then the web server replay with POST or PUT response to the cloud user request. We will simulate the HTTP protocol in the NS-2 simulator to test the web application.

Examples of the web applications are any web site hosted on the cloud data center like Amazon[1], e-bay[2],..etc.

Important Metrics for Web Applications

One of the important metrics in WA regards the performance is the time for the Get request packet from the web client till the response come from the web server and this represent the response time for the web application. The second important metric is the response time also but for the POST response packet from the web server till the

[1] www.amazon.de
[2] www.ebay.be

acknowledgment come from the web client. In general, the response time in the web application is the time for downloading pages and performing main transactions on the user side and of the back-end system side. All these in addition to the latency and throughput values.

The important metrics for WA and which we will focus on some of them in this thesis work are summarized as the following:

1. Response Time for GET Request.
2. Response Time for POST or PUT Response.

If the previous metrics are violated during WA is hosted in the cloud data center, the performance of the WAs will be degraded. Also the QoS and response times mentioned in SLA will be harmed even the service performance. In our work, after introducing the data center failures we will check on some of the previous metrics to see what is the affect of the failures on the WAs performance and the service performance. Then we solve failures to assure the high QoS as in SLA.

5.1.3 File Applications

File application is also one of the familiar and famous cloud applications. Most of the cloud users used it approximately every day where the cloud users used to upload and download files for studying, reading books, working,..etc. File application is the application of exchanging files between source and destination through the cloud. File application construction is consisting of the file server which hosted in the cloud data center servers and the file client. File application users use the FTP protocol to exchange (pull& push) files with the FTP server hosted in the data center. The user ask for upload (push) or download (pull) file from the server by using the connections established by the FTP application protocol. The file server (FTP server) reply with acknowledgment to the FTP client in case of uploading files and reply with the file requested to the FTP client in case of downloading. The FTP application protocol is working in the upper layer with the TCP protocol in the lower communication layer. We will simulate TCP under FTP protocol in NS-2 Simulator to test File applications.

Examples of File applications are like any cloud storage and exchange files such as Google Drive[1], Dropbox[2], One Drive[3] or Amazon Drive[4],...etc.

Important Metrics for File Applications

One of the important metrics in File application regards the performance is also the time used for pulling and pushing the files from and to the FTP server. This time also called the response time which is the time taken to pull or push a file from the user to/from the server in the data centers. Another important metric is the size of the transferred file. There is a trad off, if the file is too large and there is a failure has happen so the response time will duplicated. And if the file is small and there is a failure has happen so the response time will just increase a bit. Finally, the procedure used for uploading or downloading is a metric also where the speed of download is faster than the speed of

[1] https://drive.google.com
[2] https://www.dropbox.com/
[3] https://onedrive.live.com
[4] https://www.amazon.com/clouddrive/home

the upload. All these in addition to the latency and throughput values.

The important metrics for File application and which we will focus on some of them in this thesis work are summarized as the following:

1. Response Time for Pulling.
2. Response Time for Pushing.
3. Size of the transferred file.
4. Procedure used for Pulling and Pushing.

If the previous metrics are violated during FA is hosted in the cloud data center, the performance of the FAs will be degraded. Also the QoS and response times mentioned in SLA will be harmed even the service performance. In our work, after introducing the data center failures we will check on some of the previous metrics to see what is the affect of the failures on the FAs performance and the service performance. Then we solve failures to assure the high QoS as in SLA.

5.1.4 Real Time Applications

RTA is a class of cloud applications that works in a time frame methodology. The time frame methodology is meaning that the user senses as instantly or currently. RTA depends on the Worst-case Execution Time (WCET). WCET is the maximum period of time that used to accomplish a defined task or a set of defined tasks on a hardware platform. RTA is belong to the the Real Time Computing (RTC) or some times called reactive computing. RTC systems works with a real time constraints, like the response of the RTAs should be specified within a period of time constraints some times called the deadline. Also RTC sometimes refer to the Real Time Communications [62]. Real Time Communications applications can be hosted on the mobile devices or the PCs.

Real Time Applications in [63] has another name called play-back applications. The play-back applications took the signal of the network and packetizes the signals and then transmits the signal over the network. Not all the real time applications are play-back applications, there are the visualization applications which works as follow, visualization application displays the image which encoded in each packet whenever it arrives. For RTAs there are some services called the Real-Time Services (RTS). RTS is needed to meet the critical time parameters like the important metrics we will review later in this subsection. RTS is typically a two-way communication experience. The users in the RTS can instantly tell about the service quality if it slow or not working properly.

Examples of RTAs are many familiar applications, like the multimedia real-time communication including VoIP/video conferencing applications such as Skype[1], Face-Time[2], and Google+ Hangout[3]; interactive multi-player gaming applications such as Draw Something, Modern Combat 3[4], and Call of Duty[5]; and application sharing, desktop sharing, and virtual desktop interface (VDI). Real-Time Collaboration over Web (RTCWeb) is a

[1] http://www.skype.com
[2] http://www.apple.com/ios/facetime/
[3] https://plus.google.com/hangouts
[4] https://play.google.com
[5] https://www.callofduty.com/

new trend of research to enable RTC applications to run inside browsers without plug-ins.

The protocols used in this class of application are many and this because of the more different applications belong to this class of application. The network used for the RTAs is the best effort network and UDP protocol is used under the video and voice protocols. For the video streaming RTP and RTCP protocols are used. For the voice streaming VoIP is used. So in our work we will simulate RTP, RTCP and VoIP protocols in NS-2 simulator to test RTAs.

Important Metrics for RTAs

The class of the Real Time Applications have many important metrics rather than the other classes of the applications and this due to that RTAs are based on the Real Time Communications. The biggest challenge for delivering a conversation through the Internet is to provide a service with a quality that convenient to that and let the service quality like the Public Switched Telephone Network (PSTN). Conversation over the Internet services have to respect the timing requirements where the conversation media that arrives late is of little or no use to the receiver.

For the video call applications, it required a massive amount of data to be handled and loads on the servers so the processes used to deliver the video can be extremely high. So the high number of high specifications work stations, storage and servers are required for video streaming. For audio call applications the delay to receive the voice can not exceed 150 millisecond, otherwise the users will hear just echos and become irritated. That is why VOIP is a classic real time service.

From the previous requirements of the RTAs, we know that the network used for the RTAs is the best effort network, so the bandwidth should be shared with many other classes of applications and congestion that affects the quality of the experience. From that the first important metric for the RTAs is the quality of the service in the network have to be high for the RTAs. In our work we assume that the network have a high quality of the service for the RTAs. After the quality of the service, the real time service should meet the critical time parameters like minimal end-to-end delay and jitter. In addition to end-to-end delay and jitter, the number of packet lost is considered to be important metric also. The final important metric is the latency and throughput, they should be defined to be less than a defined value. All these in addition to the very high response time. Finally, the users can instantly tell about the service quality if it slow or not working properly after using and this represent the Mean Opinion Score (MOS).

The important metrics for the Real Time Applications and which we will focus on some of them in this thesis work are summarized as the following:

1. High Bandwidth (Represented in the QoS).
2. Time requirements, End-to-End Delay & Jitter.
3. The Value of the Packet Loss.
4. Latency & Throughput.
5. Mean Opinion Score (MOS).
6. High Response Time.

7. Number of workstations servers & storage.

If the previous metrics are violated during RTA is hosted in the cloud data center, the performance of the Real Time Applications will be degraded. Also the QoS and response times mentioned in SLA will be harmed even the service performance. In our work, after introducing the data center failures we will check on some of the previous metrics to see what is the affect of the failures on the RTAs performance and the service performance. Then we solve failures to assure the high QoS as in SLA.

5.1.5 High Performance Computing Applications

High Performance Application is a wide term that it is shell represents the computation of intensive applications which need very high acceleration. HPC gives the chance to the scientists and engineers to solve the complex engineering, science and business problems and also allows them to do experiments. All these by using a specified applications that require high bandwidth, enhanced networking and very high computing power capabilities. By moving HPC applications to the cloud, there are more advantages added to HPC applications such as elasticity. Elasticity is the flexibility of the data model and the clustering. Elasticity [67] is a way to decrease the risks caused by the under provisioning, it also decrease the lack of use cased by over provisioning. Another advantage of HPC applications when goes to the cloud is the built in virtualization founded in the cloud where it supports another way for doing flexibility, customization, migration, security and controlling of the resources to the HPC community.

HPC clouds have a fast growth in users of applications and also the platforms to run the HPC work load. In the beginning was the in-house dedicated super computers. HPC is no more limited to those who own supercomputers. Then the commodity clusters was appeared. The clusters was appeared with using HPC and also without using HPC. Then after the cluster, the virtualized resources is appeared with different degrees of virtualizations such as full, CPU only or none. Finally, the hybrid configuration which discharge a part from the work to the cloud.

HPC on the cloud has some benefits such as the following. The cloud allows scientists to build their own virtual machines and configure them to suit needs and performance. Clouds are convenient for embarrassingly parallel applications which don't communicate too much between partitions and which can be scaling for a common interconnects items to clouds. There are many network protocols used for HPC applications.

Examples of the HPC applications on the cloud are such as Medical Imaging, Financial (Trading), Oil& Gas Bio science, Data warehousing and other markets like Military, Data Compression, Coder/Decoder, Scientific Research[1], Security,...etc.

Important Metrics for HPC Applications

HPC applications on the cloud and cloud users for those applications face many challenges. In [67], HPC users and cloud providers faced some challenges like selecting the good platform while having a limited knowledge about the application characteristics, platform capabilities and the metrics like QoS and the cost.

[1]Like HPC Cluster in University of Luxembourg: https://hpc.uni.lu/

The challenges faced by HPC cloud users are representing the important metrics for this class of applications. The first challenge is the absence of high speed interconnects and the noise free operating systems to allow the strongly coupled HPC applications to scale. Bringing data from in and out of the cloud is another challenge. The submission model regards to the job queuing and the reservation of the VM deployment is also another challenge. So the important metrics are scoped in the processing power and the number of processors founded in the top of the node, the memory space used, the network performance and the Operating system used.

The important metrics for the HPC application and which we will focus on some of them in this thesis work are summarized as the following:

1. High processing (computing)power like - processors & cores. -RAMs.
2. Noise free operating Systems -VM deployment.
3. High Bandwidth on the Network
4. Speed of bringing data in and out the cloud (I/O processes).

If the previous metrics are violated during HPC application is hosted in the cloud data center, the performance of the HPC applications will be degraded. Also the QoS and response times mentioned in SLA will be harmed even the service performance.

5.1.6 Mobile Cloud Applications

MCC applications became the most used technology all over the world. Nobody nowadays does not have a mobile cell phone and hosted on it many mobile applications and mobile cloud applications and using it not every day but approximately every ten minutes. The new electronic devices such as tablets, smart phones, mobile note and cloud computing resources all are under a new fast growing field called the mobile cloud computing. MCC [65] is the extension of the cloud computing with adding an ad-hoc network which based and consisted of a group of mobile devices. The advantage of the new MCC technology of computing is that the applications of the MCC are not limited to a certain type of mobile devices or mobile operating system.

Simply defined MCC [64], it is the infrastructure when gathering both the data storage operations and the data processing operations and do them out side of the mobile device. MCC applications carry the computing power and the data storage out side from the mobile phones and doing them inside the cloud. MCC applications offloading is became a solution to increase the capabilities of the mobile devices like the computation power, storage space and the power consumption, and this done by offloading the computation of the applications to the servers in the cloud data centers [66].

The architecture of MCC application is consisting of two layers, The data centers layers which is the place where MCC application is hosted. The second layer is the IaaS, PaaS or SaaS. The advantages of MCC applications : (1) Extending battery lifetime. (2) Improving the data storage capacity and processing power. (3) Improving reliability. Also MCC inherits some advantages from clouds to mobile applications and services as following: (1) Scalability. (2) Multitenancy. (3) Ease of integration. (4) Dynamic Provisioning.

There are many examples of MCC applications, we can say that all the applications on the cloud computing installed on the PCs, there are similar to them in the mobile cloud computing but they installed on the mobile devices. MCC applications generally like Mobile Commerce[1], Mobile Learning[2], Mobile Health Care[3] and Mobile Gaming[4]. There are many network protocols used for MCC applications.

Important Metrics for MCC Applications

There are many challenges faced by the MCC applications and from them we can get the important metrics for MCC application class. The first challenge is the low bandwidth where in MCC, the radio resources for wireless networks is much scare as compared with the traditional wired networks. The second challenge is the availability, this because of the traffic congestion. The third challenge is the computing offloading, as we mentioned before offloading is one of the main features of MCC to improve the battery lifetime for the mobile devices and also to increase the performance of the applications and that is why it is a challenge to implement. The final challenge is how to enhance the efficiency of data access.

The important metrics for the MCC application and which we will focus on some of them in this thesis work are summarized as the following:

1. Bandwidth of the WiFi network.
2. Traffic congestion on the wireless network.
3. Computing & application (Computing power processing) offloading.
4. Data access from the cloud data centers.

If the previous metrics are violated during MCC application is hosted in the cloud data center, the performance of the MCC applications will be degraded. Also the QoS and response times mentioned in SLA will be harmed even the service performance.

5.1.7 Massive Data Analysis

Massive Data Analysis applications are like the search index and computing the web also the Big Data Analytic applications.

Big Data Analytic applications are a novel type of software applications that clout a large scale data, which is very big to fit in memory or even on one hard disk drive, to uncover actionable knowledge by using the large scale parallel processing infrastructure [60]. There are many network protocols used for the MDA applications.

There are many examples of the MDA applications like the Hadoop[5] cloud which is one of the familiar massive data analysis clouds. Also the Facebook[6], Instagram[7] and

[1] Zillow, Nordstorm, MizPee and Target
[2] Snapguide, Infinite Thinking Machine and Knowledge Guru
[3] CATRA and Directly Observed Therapy
[4] Candy Crash
[5] https://hadoop.apache.org/
[6] www.facebook.com
[7] www.instagram.com

Twitter[1] web sites are represent the massive data analysis applications, where in the background they use the huge amount of information related to the users and they try to analysis it. Finally, the search engines like Google[2], Yahoo[3],..etc also represent the massive data analysis applications.

Important Metrics for MDA Applications

There are technical requirements and challenges for the big data in the cloud. The first technical requirement is the scalability and the acceleration, where the big memory space and high processing power CPU and cores are needed. The second is the agility and the elasticity. The last one is the performance metrics like response time, latency, throughput,..etc all these values should not goes over a defined limit.

One of the important metrics is the response time that is the time used to retrieve the data after requesting in the search engine for example. The second important metric is the computing power like the CPU and cores used to do the processing retrieve and analysis the data also the RAM capacity should be respectful. the last metric is the precision and recall, precision is the fraction of the retrieved data that are relevant. Recall is the fraction of relevant data that are retrieved.

The important metrics for the MDA application and which we will focus on some of them in this thesis work are summarized as the following:

1. Response Time.
2. Computing Power.
3. Precision & Recall.

If the previous metrics are violated during MDA application is hosted in the cloud data center, the performance of the MDA applications will be degraded. Also the QoS and response times mentioned in SLA will be harmed even the service performance.

5.1.8 Distributed Cloud Applications

Distributed cloud applications are consisting of a group of VMs which running a set of connected software components [76]. Although the building of the distributed applications on the cloud is not easy work. Each VM in distributed cloud application has many software configuration parameters like the local aspects such as pool size, authentication data,..etc and the other depend on the the definition of the interconnections between the remote elements.

The distributed applications became the style of communications between the companies crew and also the normal people, where it represents the e-mail servers which distributed in the cloud. In the normal distributed systems there is a collection of computers that working together and introduced as one large computer. In the cloud distributed applications the collection of computers is hosted inside the cloud data center. The two distributed system even on the cloud or not they depend on the distributed computing. Distributed systems have some advantages like distribution, high Performance, resource

[1] www.twitter.com
[2] www.google.com
[3] https://search.yahoo.com/

sharing, reliability, availability, incremental growth,..etc some of them inherited from the cloud computing. The disadvantages of the distributed systems like complexity, software development difficulties and networking problems are solved by moving the distributed applications to the cloud.

Examples of distributed applications on the cloud are any e-mail server putted on the cloud and any application that distributed in many virtual machines in the same time. The e-mails examples such as Gmail[1], Yahoo mail[2], Microsoft mail[3] and the iCloud mail[4]. The network protocols used between e-mail server (Mail delivery or transfer agent) and email client (Mail user agent) are SMTP and POP3 protocols. So in our work we will simulate the SMTP protocol in NS-2 simulator to test the Distributed Cloud Applications.

Important Metrics for Distributed Cloud Applications

There are many important metrics in the distributed cloud applications. The first one is as usual the response time which here is the time for sending email for example and get the acknowledgment of receiving . The second metric is the availability of VMs. The high bandwidth is required to build distributed cloud like the network capacity and the delay time should be minimized. The delay time is the response time from the network. Finally the high computation power of the server nodes is the last metric. All these is addition to the latency and throughput values.

The important metrics for the distributed cloud application and which we will focus on some of them in this thesis work are summarized as the following:

1. Response Time.
2. The availability of the VMs.
3. The bandwidth of the network (capacity & delay time).
4. The computation power of server nodes.

If the previous metrics are violated during DA is hosted in the cloud data center, the performance of the distributed cloud applications will be degraded. Also the QoS and response times mentioned in SLA will be harmed even the service performance. In our work, after introducing the data center failures we will check on some of the previous metrics to see what is the affect of the failures on the DAs performance and the service performance. Then we solve failures to assure the high QoS as in SLA.

In the next section, we classified the previous eight classes of applications into two categories of failure sensitivity.

[1] https://mail.google.com
[2] https://mail.yahoo.com/
[3] https://login.live.com/
[4] https://www.icloud.com/

Table 5.1: Summary of the Classification of Cloud Applications

Application Classes	**Brief Definition of the Class of Application**	**Example of the Application**	**Most Important Metrics**	**Protocols Used to simulate the class**	**How tested in Real App?**
Highly Interactive Applications HIA	This class of application allow the human cloud users to deal and interact with the applications directly by using the manipulation input devices such as mice, joy sticks, body position sensors,..etc	Gaming applications,the virtual reality applications, query response applications and office productivity applications like the Google Docs online and the online office	1. Response Time (RT). 2. User to User Time (UUT). 3. Scaling of the main memory size RAM and the processing power. 4. Input and Output intensive transactions speed.	IRTP, RTP upper to the UDP	Not Tested
Web Application	It is like any web page we can browse on the Internet. The construction of the web applications is constructed by the web server which is one of the servers inside the cloud data centers and the web client who is the cloud user.	Any web site hosted on the cloud data center like Amazon, e-bay,..etc.	1. Response Time for GET Request. 2. Response Time for POST or PUT Response.	HTTP upper to TCP	Real local web Serve with a web application designed by PHP and using Charles.
File Application	It is the application of exchanging files between source and destination through the cloud. File applications construction is consisting of the file server which hosted in the cloud data center servers and the file client.	Like any cloud storage and exchange files such as Google Drive, Dropbox, One Drive or Amazon Drive,...etc.	1. Response Time for Pulling. 2. Response Time for Pushing. 3. Size of the transferred file. 4. Procedure used for Pulling and Pushing.	FTP upper to TCP	Real local file Server with a file application designed by PHP and using Charles.
Real Time Applications RTA	It is a class of cloud applications that works in a time frame methodology which means that the user senses as instantly or currently. It is called a conversational services.	VoIP and Video applications likeSkype, FaceTime, Hangout. gaming applications and application sharing desktop sharing, and virtual Reality	1. High Bandwidth (Represented in the QoS). 2. Time requirements, End-to-End Delay & Jitter. 3. The Value of the Packet Loss. 4. Latency & Throughput. 5. Mean Opinion Score (MOS). 6. High Response Time. 7. Number of workstations servers & storage.	RTP/RTCP, VoIP and the UDP is in the lower layer.	Real local chatting Server with a chat application and using Charles.
Massive Data Analysis Applications MDA	It is a new category of software applications that levarge large scale data, which is typically too large to fit in memory or even on one hard-disk drive, to uncover actionable knowledge.	Like Search index such as Google and Yahoo Search, also the Hadoop cloud. In addition to theFacebook, Twitter and Instagram. Also the Big Data Analysis applications.	1. Response Time. 2. Computing Power. 3. Precision & Recall.	Not Simulated	Not Tested
High Performance Computing Applications HPC	It is a wide term that it is shell represents the computation of intensive applications which need very high acceleration. It gives the chance to the scientists and engineers to solve the complex engineering, science and business problems and also allow them to do experiments.	Medical Imaging, Financial (Trading), Oil& Gas Bio science, Data warehousing and other markets like Military, Data Compression, Coder/Decoder, Scientific Research, Security,...etc.	1. High processing (computing)power like - processors & cores. -RAMs. 2. Noise free operating Systems -VM deployment. 3. High Bandwidth on the Network 4. Speed of bringing data in and out the cloud (I/O processes).	Not Simulated	Not Tested
Mobil Cloud Computing Applications MCC	It is the infrastructure when gathering both the data storage operations and the data processing operations and do them out side of the mobile device. So the data storage and the processing are done inside the cloud.	Mobile Commerce, Mobile Learning, Mobile Health Care and Mobile Gaming.	1. Bandwidth of the WiFi network. 2. Traffic congestion on the wireless network. 3. Computing & application (Computing power processing)offloading. 4. Data access from the cloud data centers.	Not simulated	Not Tested
Distributed Cloud Applicaions	The Cloud application functionality divides among multiple application components that can be scaled out independently. It consisting of a group of VMs which running a set of connected software components	Any e-mail server putted on the cloud like Gmail, Hotmail, Yahoo mail and icloud mail.	1. Response Time. 2. The availability of the VMs. 3. The bandwidth of the network (capacity & delay time). 4. The computation power of server nodes.	SMTP with the UDP in the lower layer	Real local mail Server with a email application and using Charles.

5.2 Classification of Failure Sensitivity

In this section, we classified the previous eight classes of the applications by another classification base that is the sensitivity of the failure respect to the application class. The classification in this section is based on two main classes. The classes are the HSA and the LSA. In the high sensitive applications class, the applications belong to this class have a high impact from the data center failures and they probably can't continue running (working) during the failure happen. In the low sensitive applications class, the applications belong to this class have a low impact from the data center failures and they probably can continue running (working) during the failure happen.

For FAs, it belongs to the LSA class where it can continue work while failure occurring like if we download or upload a file and there is a failure happen the file will continue download or upload but just will take more time than as usual. For WAs, it also belongs to the LSA class where browsing a web page while failure occurring just need more time than as usual. FAs and WAs have impact from the network and physical server failures. For RTAs, it belongs to the HSA class where it can't continue working during failures happen. For the Mobile Cloud applications, it belongs to the LSA class where it can continue work while the failure happen. For HPC applications, it belongs to the HSA class where it can't continue working during failures happen and also it need a high computing power and bandwidth. For MDA applications, it belongs to the HSA class where it can't continue working during failures happen and also it need a high computing power and bandwidth. For HIAs, it belongs to the HSA class where it can't continue working during failures happen and also it need a high response time and input/output transactions with high speed. HIA, MDA and HPC applications performance have a bad impact from physical server failures. Finally, for the DAs, it belongs to the LSA class where it can continue work while the failure happen like if we waiting for email or sending email we don't care about when will receive the email or when will the email be received. DA, RTA and MCC applications performance have a bad impact from network failures.

Table (5.2) summarizes the classification of the sensitivity for the cloud application classes. It shows also which failure have most effect on each class of application.

After classifying cloud applications, we need to classify and identify DC failures which may harm and face SLA services and QoS. In the next chapter, we review CDCs failures and see how we can solve or mitigate them to assure the high QoS and high applications performance as mentioned in SLA where if we solve the failures we can assure that no violations in SLA response times and QoS.

Table 5.2: Classifying Cloud Applications based on the Failure Sensitivity

Class #	**Application Class**	**Failure Sensitivity Class HSA/LSA**	**Which Failure?**
1	File Cloud	LSA	Network
2	Web Cloud	LSA	Network
3	Real Time	HSA	Network
4	High Performance Computing	HSA	Physical Server
5	Mobile Cloud Compuing	LSA	Network
6	Massive Data Analysis	HSA	Physical Server
7	Highly Iteractive	HSA	Physical Server
8	Distributed Cloud	LSA	Network

Chapter VI

Data Centers Failures

"We can not solve our problems with the same level of thinking that created them."

Albert Einstein

In this chapter, we review and classify the cloud data center failures. There are many types of failures such as the data center network failure, the physical server failures, the unavailability of VMs,... The failure we will focus on it in this work is the the network failures where the structure of the cloud data center failure is consisting of many devices and links, so the occurring probability of network failures is high. Network failures is discussed in the first Section 6.1. The second important failure is the physical server failures, where the physical servers are where the cloud applications are hosted and that it is why this failures also important. In the second Section 6.2, we review the physical server failures. There are many other failures types in CDC we discuss them in Section 6.3. After discussing the failures on the data centers, we explain how those failures can be solved or at least can be mitigated in Section 6.4 and that is how we solve the failures in our work. Finally in this chapter we will discuss which cloud data center failures have the most effect on each class of cloud applications in the last Section 6.5.

6.1 Network Failures

In Chapter 2, we explained the architecture of the data center network. The architecture of the data center network is in Fig. 2.5 [4]. Also Fig. 6.1 [49] shows the network topology of the cloud data center. It is not so complicated, but have many devices and links. The devices are like the Core and Access Routers, the Aggregation and Top Of Rack switches, Rack of servers and the load balancers and firewalls which connected to the Aggregate switches. All these in addition to the links between those devices. From the study has been done in [8], the data center network architecture approximately represents 15 % of the whole cost of the data center architecture. Failures in the network of cloud data center represent about 60% of the whole cloud data center failures. There are some features are required to be in the network of the data centers. Networks need to be scalable, efficient, fault tolerant and easy to manage. Also it should have the reliability which means how to analysis and deal with the errors in the network.

The network data center failures is categorized into different types such as the software failures in the network, hardware failures in the network devices, configuration failures in the network devices and unknown failures in the network. Each type from the previous

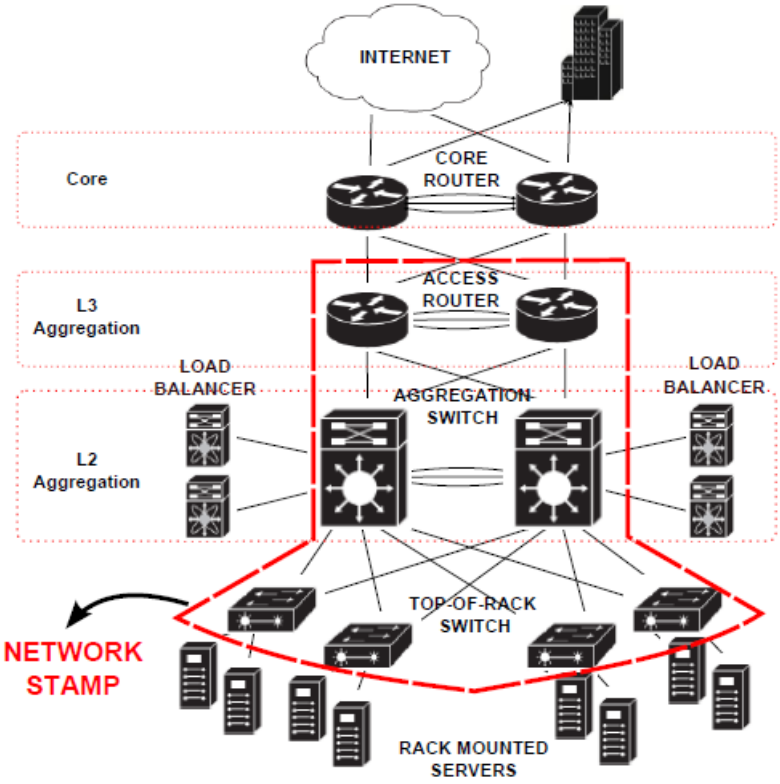


Figure 6.1: Data Center Network Topology [8]

failures have a percentage of happening on the network data center. The percentage based on the study has been done in [49] and it said that the configuration failures represent about 38% of the total failures in the network failures, the software failures represent about 21% of the total network failures, the hardware failures represent about 18% of the total network failures and the unknown failures represent about 23% of the total network failures. Those classes of network failures such as hardware, software bugs, configuration/human mistakes, Internet connectivity and planned maintenance are represented as the network failure sources or in other words they are the problems caused the failures. The failures themselves will be discussed in the next Subsection 6.1.

Types of Failures

Here, we discussed the network failures themselves. The network failures are as the following:

1. ***Link Failure:*** The link in the network of data center is the connection between any two physical devices (e.g. switches, routers, servers, load balancers,...etc) whatever this connection is wired or wireless. The link failure occurs when the connection between two devices on a specific interfaces is broken or getting down. This failure can be detected by SNMP[1] monitoring by checking the interfaces of the devices [4]. The Link failure is considered under the hardware problems and failures.

2. ***Device Failure:*** The devices in data center network are the switches, routers,..etc. This failure occurs when the function of routing and switching the traffic operations

[1]Simple Network Management Protocol: used to manage the devices of the internet protocol network

of the devices is not working properly. The problems cause this failure are variety such as the device powered down for crashing due to errors of hardware. Also this failure considered under the hardware failures.

3. ***Fail Stop:*** This failure is considered under the software failures. This failure happen when a process or a machine dies and don't come back. This failures are very easy to detect. And also it has a bad impact on availability of the applications.

4. ***Byzantine Failure:*** This failure also is considered under the software failures. This failure occurs when a process has failed and it is still running with incorrect operations (spewing garbage). But this failure is very difficult to detect, and has a bad impact on availability of the applications.

5. ***Configuration Failures:*** The switches and routers in the data center network need configuration to do traffic routing and switching. When multiple errors occur in the switches or on the routers this configuration file errors. This failure is easy to detect, but might produce a bad impact on application performance.

6. ***Unknown Failures:*** There are many factors that cause the unknown failures such as switch stops forwarding, imbalance triggered overload, lost configuration or high CPU utilization. These failures often cause abnormally high latency or packet losses.

7. ***Performance Degradation Failure:*** This failure can be considered under any category of the network failures. It is like something has failed which make it slower, but it is still correct (straggler). It is very difficult to detect, and has a bad impact on availability of the applications.

The solutions and the mitigation of the network failures recede in the replication, replacement, deactivating or restarting all these solutions will be discussed in details in Section 6.4. In the next section, we review the physical server failures where them harm the performance of applications hosted in CDCs and this degrade the QoS mentioned in SLA. We will see also how we will solve them.

6.2 Physical Server Failures

In this section, we discuss the physical server failures. The physical servers in the cloud data center are where the applications and data storage are hosted. The servers consist of multiple hard disks, memory modules, network cards, processors,... etc. Each one of the components of the servers may be failed. The physical server hardware failures can lead to a degradation in the performance to end users due to service unavailability [32]. The failures in the physical servers will be discussed in the next Subsection 6.2.

Types of Failures

The failures of the physical servers in data centers produced from the physical components of the servers it self. The physical failures are as the following:

1. ***Hard Disk Failure:*** This is the first physical failure occur in the servers of the data center. It happens due to a physical problems in the hard disks of the servers. This failure is easy to detect where if it occur the server becomes dead. The hard disk failures represent approximately about 70% of the total physical server failures.

2. ***Physical Memory Failures:*** The main memory in the physical servers is the memory used by the server processor for doing the operations processing. The main memory is like the RAM with it's types DRAM and SRAM. This failure happens due to a physical problems in the main memory of the physical servers. This failure is easy to detect because if it occur the server becomes dead. The memory failures represent approximately about 5% of the total physical server failures.

3. ***RAID Controller Failures:*** The raid controller is a hardware device that is used to manage the hard disk drives in the physical servers. The failure of the raid controller occurs if there are any problems in a group of hard disks in the servers. The RAID controller failures represent approximately about 6% of the total physical server failures.

The rest 18% of physical server failures are due to other factors. There is a study about the physical servers failures has been done in [32], this study has been done to 100 servers in the data center to study the failures happen in the 100 servers. The study said that 78% of the failures came from hard disks, 5% of the failures came from Raid Controllers, 3% of failures came from the Main Memory of the servers and the rest 13% of the failures came from a collection of other components with no single dominating.

In the next section, we show the other failures may happen inside cloud data center. The other failures like VM unavailability, single point of failure and the no problem found failure.

6.3 Other Failures

In addition to the network failures and physical server failures, there are many failures in the data centers such as virtual machines unavailability failures, the single point of failure, soft failure or the no problem found (NPF) failures, individual rack failures and the individual component failures. We describe the VMs unavailability failures in the Subsection 6.3.1. Then the single point of failure will be discussed in the Subsection 6.3.2. Finally, in Subsection 6.3.3 the no problem found failure will be discussed.

6.3.1 VM Unavailability Failures

Virtualization is a way for increasing the utilization of the resources in the cloud data center. It also one of the main technologies included in cloud computing. The virtual machine availability faced many challenges that represents the VM unavailability failures. The challenges as following:

1. ***VM Hopping:*** An attacker on one VM can access another VM.

2. ***VM Mobility:*** Quick spread of vulnerable configurations that can be exploited to endanger security.

3. ***VM Diversity:*** The range of operating systems create difficulties when securing and maintaining VMs.

4. ***Cumbersome Management:*** Management of the configuration, network and security where the specific settings is a difficult task.

6.3.2 Single Point of Failure

Single point of failure (SPOF) is a probable risk that affects the system reliability, availability and performance [9]. SPOF happens in both the software and hardware layout. SPOF in data centers makes the system unavailable to the users.

Assume that inside the data center, all the applications run on one single application server. When this single application server fails, then the applications become unreachable to the cloud clients. If we assume the same scenario but with single switch or single router in the data center. If there is any failure in the switch or the router this will let the servers not available again to the users. Fig. 6.2 [9] describes the single point of failure.

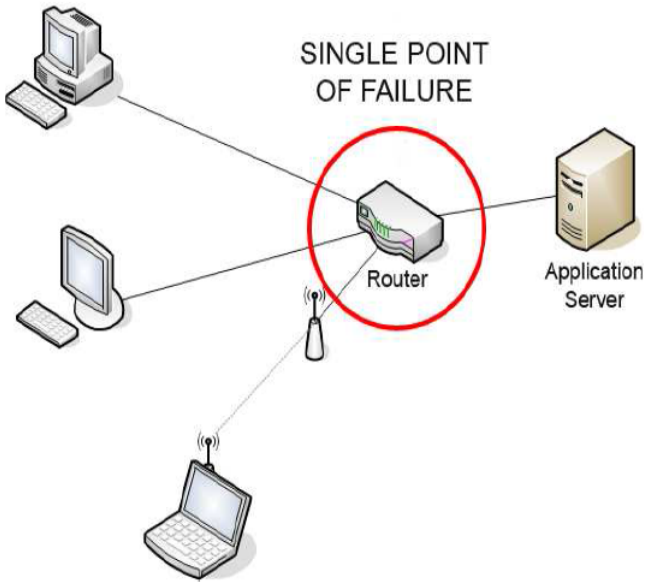


Figure 6.2: Single Point Of Failure [9]

6.3.3 No Problem Found Failures

There are many problems caused the soft failure like the soft failure in hard disk, memory, motherboard, net cable, power, replaced machine and in CPU. The highest percentage of the soft failures is filling in the category "no problem found" failure. The no problem found failures are very hard to root-cause and even harder to fix. There is a study about soft failure in [10] shows that the "no problem found" is the highest probability of occurring which approximately about 43% as in Fig. 6.3.

After discussing most of the data center failures, we need to review how we can solve those previous failures and specially the network failure where it represent the highest probability of occurring. In the next section, we review the solving and mitigation techniques are used to mitigate failures.

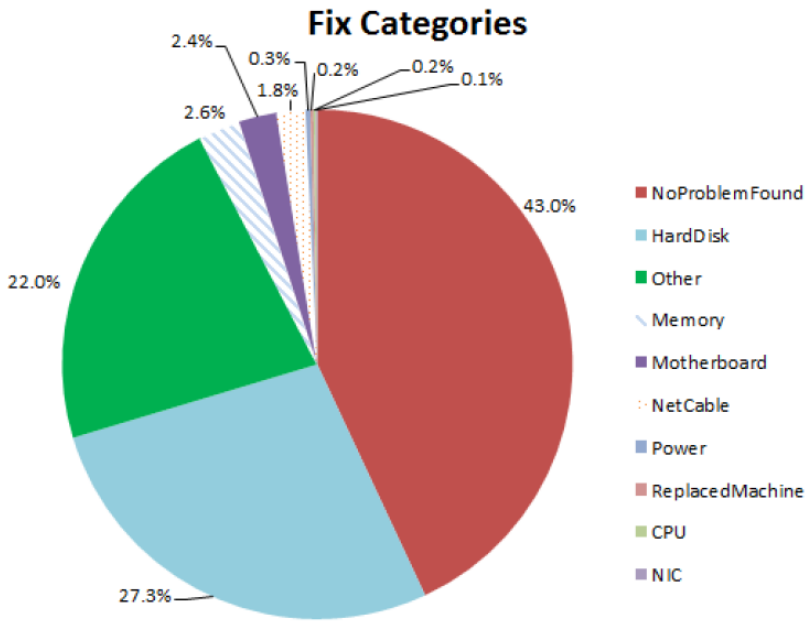


Figure 6.3: Soft Failures in the data centers [10]

6.4 Solving & Mitigating Failures

In this section, we review the mitigation and solving of the previous data center failures. Most of the failures mentioned above can be solved or mitigated with one of the following actions such as Replication, Replacement, Redundancy and Restarting or reactivating. The solving of the failures is consisting of four stages, the first is the failure **detection**. The second is the **mitigation** of the failures and the third stage is the failure **diagnose**. The final stage is the **repairing** of the failure. The data center network failures mitigation are in the Subsection 6.4.1. The other failures mitigation like physical server failures, VMs Unavailability, SPOF and soft failure NPF are in Subsection 6.4.2.

Table (6.1), summarizes all the failures mitigation, the detection, the diagnosis and the repair of the failures.

6.4.1 Network Failure Mitigation

Mitigation of the network data center failures allows the network of the data center to operate and work continuously even if in the failure exists. And this till solve or repair the failures. The network failures can be mitigated by one of the following actions, the redundancy, the replacement or restarting and reactivating. Approximately about 60% of the network failures can be mitigated and solved by restarting or reactivating [49], and the rest of the failures can be solved or mitigated by the redundancy and replacement.

Redundancy in the data center network is three levels, the device level, protocol level and the application level. In the device level redundancy, as shown in Fig. 6.1 & Fig. 6.4 where the redundancy of the links and the network devices like the Core and Access routers, also the Aggregate and the Top of Rack switches. In the protocol level redundancy, the authors in [49] mentioned three protocols for the protocol level redundancy like Link Aggregation Control Protocol (LACP), Virtual switch and Full-mesh CORE s.

In the application level of redundancy, the distribution and replication of the application under multiple of the Top of Rack (ToR) switches, where the ToRs represent a single point of failure. One of the redundancy techniques is the VM migration.

The mitigation for the Link failures is either by redundancy of links or by replacing the cable. For the device failure, the mitigation technique used is the device redundancy (like virtual machine migration) till repair the device. For the fail stop and byzantine failure, they can be mitigated either by restarting or by application redundancy. For the configuration failure, it can be mitigated either by device redundancy till update the configuration of the switch or the router or by reactivating the device (e.g. switch, router). For the unknown failures, they can often be mitigated by restarting and this due to the failures is being undefined or unknown.

6.4.2 Other Failures Mitigation

Here, the other failures like physical server failures, VMs unavailability failures, soft failures and SPOF mitigation. Again, those failures can be mitigated either by redundancy or replacement but not with restarting because they should be repaired. In the other hand, soft failures like "no problem found" can be mitigated by the restarting or reactivating.

For all the failures in the category of the physical server failures such as the hard disk problem, the main memory problem and the raid controllers problems, all those failures can be mitigated by VMs migrations (Redundancy of Servers) till the replacement of the damage hardware is done. Those failures can't mitigated by restarting or reactivating because once it occur the device considered as dead so we have to replace the damaged component. Fig. 6.4 [8] shows the server redundancy and the network device redundancy.

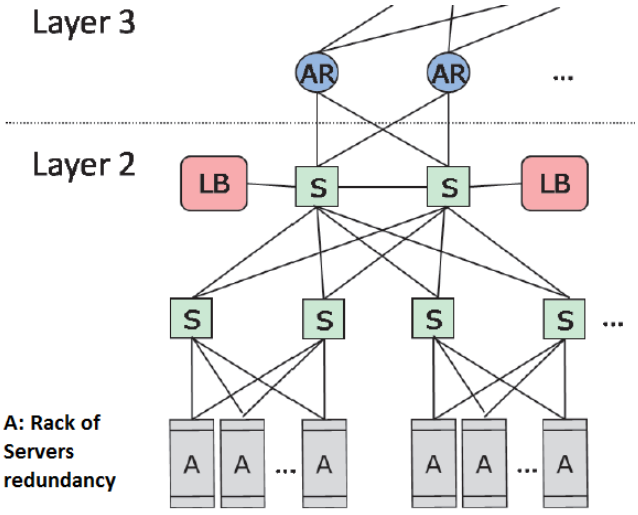


Figure 6.4: Server rack redundancy & Network devices redundancy [8]

The VMs availability is itself one of the mitigation techniques which used to do VM migration, so in case the failure occur in the VMs and they became unavailable on a specific machine (e.g. server). We can mitigate this failure by migrating all the operations not in the same machine but to another completely different machine (server) and with

this migration we solve the problem of VMs unavailability failures by also redundancy of servers.

The eliminating of SPOFs is by the redundancy or by the high availability clusters. The redundancy here is represented in two items, the logical and the physical redundancy [9]. The logical redundancy is the redundancy in the applications and software. The physical redundancy is the redundancy of the devices (e.g. servers, switches, routers,..etc) and this provide the high availability clusters. The clusters are a group of servers connected by the network links in the data centers. Fig. 6.5 [9] shows the multiple cluster regions to mitigate the SPOFs.

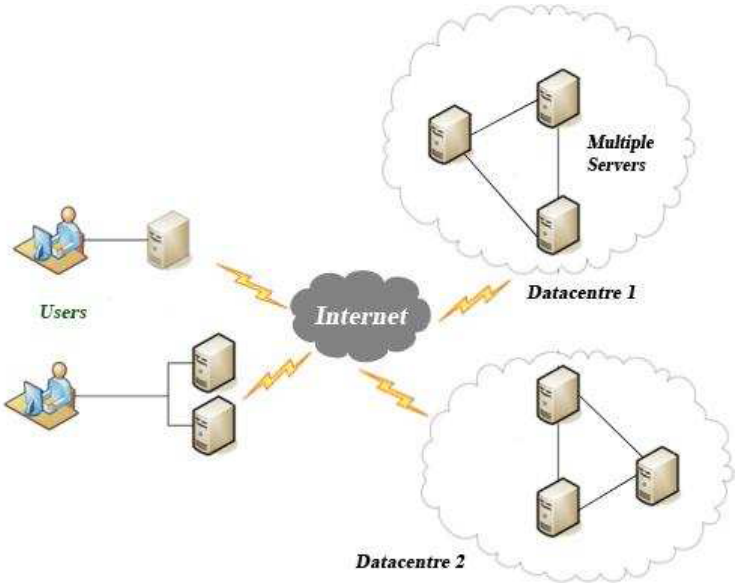


Figure 6.5: Multiple Servers in Multiple Cluster Regions to Eliminate SPOFs [9]

To mitigate the soft failure and the "no problem found" (NPF) failure either by restarting or reactivating because this failures are happen but nobody can detect it easily. In [10], they mitigate and solve the soft failure and the NPF failure by the following:

1. Process modifications.
2. Architectural modifications to hardware components in DCs.
3. Software approaches that can modify and mask the faults.

Table 6.1: The DCs Failures Summary with Mitigation and Repair

DCs Failure	***Failure Category***	***Failure Detection ?***	***Failure Mitigation***	***Failure Diagnosis***	***Failure Repair***	***Percentage of Failure***
Network 60%	Software (e.g. fail stop,.etc.)	Difficult	1. Application Redundancy. 2. Restarting and reactivating.	1. link layer loop. 2. Fail Stop. 3. Byzantine	1. Update software	21%
	Hardware (e.g. Device,Link,..etc.)	Easy	1. Device Redundancy. 2. Restarting and reactivating.	1. Link Problems. 2. Devices Problems. 3. unstable power.	1. Replace device. 2. Repair power.	18%
	Configuration (e.g. Switch,Router,..etc.)	Easy	1. Device Redundancy. 2. Reactivating.	1. errors on one switch or router configuration.	1. Update configuration	38%
	Unknown	Difficult	1. Restarting and reactivating.	1. switch stops forwarding. 2. high CPU utilization. 3. lost configuration.	Not Defined	23%
Physical Server 15%	Hard Disk	Easy	1. Server Migration. 2. Replacement. 3. Redundancy.	Problems in the hard disks	Replace Hard Disks.	71%
	Main Memory	Easy	1. Server Migration. 2. Replacement. 3. Redundancy.	Problems in the memory: 1. Problems in DRAM. 2. Problems in SRAM.	Replace RAM.	5%
	RAID Controller	Easy	1. Server Migration. 2. Replacement. 3. Redundancy.	Problems in a collection of hard disks.	Replace raid controller.	6%
	Other factors	Difficult	1. Server Migration. 2. Redundancy.	Collection of problems may be undefined	Not Defined	18%
VMs Unavailability Failures 10%	Virtual machines not available.	Easy	Server Migration.	Errors in the virtual machines (No enough VMs)	Repair the physical machine.	20%
Single Point of Failures SPOFs 5%	Network decice (e.g. switch, router) as SPOFs	Easy	1. Link Redundancy. 2. Network device Redundancy.	1. Switch dead. 2. Router dead. 3. Link drop.	Put redundancy network devices and links.	40%
	Servers as SPOFs	Easy	1. High availability Clusters. 2. Server Redundancy.	Big Server unaccessable.	Put redundancy servers.	60%
Soft Failures 10%	No Problem Found NPF	Difficult	Restarting or Reactivating	Devices stop working suddenly	Harder to Fix	43%

After solving failures, we need to show the impact of failures on the cloud applications performance to see how this impact will harm the SLA QoS, and see which problem exactly cause the failures. In the next section, we discuss the failures impact on cloud applications.

6.5 Failures Impact on Cloud Applications

In this section, we discuss the impact of the failures on the cloud applications but from applications perspective view. In other words, we discuss which failures have the most effect on the cloud applications which are hosted on cloud data center.

For HIAs, the physical server failures (e.g. Memory, Hard disk problems) and network failures (e.g. hardware, software problems) have the most effect on this class of cloud applications. For WAs, the network failures (e.g. Hardware, configuration problems) has the most effect on this class of cloud applications. For FAs, the network failures (e.g. Hardware, configuration problems) and physical server failures (e.g. Hard disk, raid controller problems) have the most effect on this class of cloud applications. For RTAs, the network failures (e.g. software, Hardware problems) and physical server failures (e.g. Main Memory) have the most effect on this class of cloud applications. For MDA applications, the physical server failures (e.g. Memory, Hard disk problems) and network failures (e.g. hardware, software problems) have the most effect on this class of cloud applications. For HPC applications, the physical server failures (e.g. Memory, Hard disk problems), the unavailability of the VMs and network failures (e.g. hardware, software, configuration problems) have the most effect on this class of cloud applications. For MCC applications, the network failures (e.g. Hardware, configuration problems) and physical server failures (e.g. main memory problem) have the most effect on this class of cloud applications. For DAs, the unavailability of the VMs and network failures (e.g. Hardware, configuration problems) have the most effect on this class of cloud applications.

Table (6.2), summaries the failures that have the most effect on each class of the cloud applications mentioned in the previous chapter 5.

In the next two chapters, we will start our experiment and testing implementation. The next section shows the real scenarios experiments, and it's next section explains the simulation experiments. We will introduce failures to cloud applications and see the impact of those failures. After that, we will solve the failures in the simulation experiments on NS-2 simulator. The solving and mitigating to failures, help us to assure the SLA QoS and the performance of the services as mentioned in SLA document under the response times metric. Also, we will show how we solve the failures to respect the response times in SLA which is the time taken to response for critical failures. We will implement two mitigation technique, the VM migration and FEC. In Chapter 9, we will show our experimental results which we will get from our experiments.

Table 6.2: Failures have the Most Effect on the Cloud Applications

Cloud Application Classes	**Failures have the most effect**	**Problems cause the failures**
Highly Interactive Applications HIA	1. Physical Server Failures. 2. DCs Network Failures.	1. Memory, Hard disk. 2. Hardware, Software.
Web Application	1. DCs Network Failures.	1. Hardware, Configuration.
File Application	1. DCs Network Failures. 2. Physical Server Failures.	1. Hardware, Configuration. 2. Hard disk, RAID controller.
Real Time Applications RTA	1. DCs Network Failures. 2. Physical Server Failures.	1. Hardware, Software. 2. Main Memory.
Massive Data Analysis Applications MDA	1. Physical Server Failures. 2. DCs Network Failures.	1. Memory, Hard disk. 2. Hardware, Software.
High Performance Computing Applications HPC	1. Physical Server Failures. 2. VMs Unavailability Failures. 3. DCs Network Failures.	1. Memory, Hard disk. 2. Unavailability of the VMs. 3. Hardware, Software, Configuration.
Mobil Cloud Computing Applications MCC	1. DCs Network Failures. 2. Physical Server Failures.	1. Hardware, configuration. 2. Main Memory.
Distributed Cloud Applicaions	1. VMs Unavailability Failures. 2. DCs Network Failures.	1. Unavailability of the VMs. 2. Hardware, Configuration.

Chapter VII

Real Applications Experiments

"when you have eliminated the impossible, whatever remains, no matter how improbable, must be the truth."

Sherlock Holmes- The Sign of Four

In this chapter, we discuss how we to test cloud applications in real scenarios. Our real scenarios are tested with a real application hosted on a local device. We used the Charles[1] tool to introduce failures in the network and to get the values of metrics for each application class. Then we introduce a mathematical based model for testing each application class. We test four application classes, WA, FA, DA and RTA. In Section 7.1, we show the testing for web application and it's mathematical model. The mathematical based model and testing of file application are in Section 7.2. The testing for distributed application in addition to it's mathematical model are in Section 7.3. Finally, the real time application testing and it's mathematical model are in Section 7.4.

7.1 Web Application Testing

Web application is one of the most used cloud applications. It is like any web page we can browse on the Internet. The web application is constructed with a web server hosted on cloud data center and web client which is the cloud user. The cloud user can ask for web pages from the server to browse it. The testing scenario is in Subsection 7.1.1 and the mathematical based model for testing WA is in Subsection 7.1.2.

7.1.1 Testing Scenario

The web client ask web server to browse a web page by using HTTP Get request, the server will response for client by HTML[2] page and display it on a browser.

To test web application, we setup a VertrigoServ[3] web server and then developed a very simple web application using PHP[4]. Fig 7.1 shows the topology we designed to test real web application. We used this web application that hosted on web server to test web application performance metrics in case we introduce failures. In case of network failures, the web application will operate but with a degraded performance. In case of

[1] Charles: Web DEBUGGING Proxy, charlesproxy.com
[2] HyperText Markup Language
[3] http://vertrigo.sourceforge.net/
[4] http://php.net/

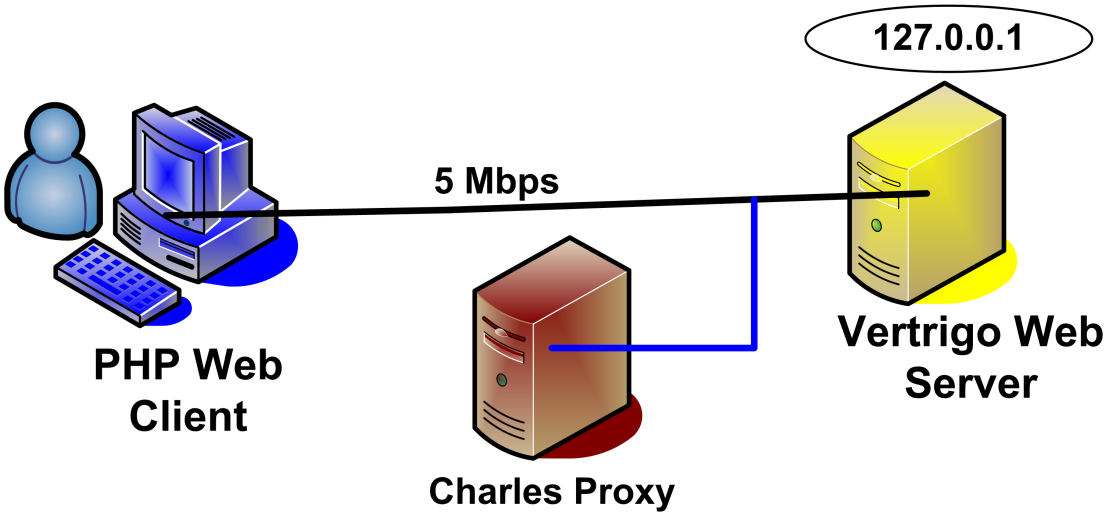


Figure 7.1: Real Scenario Topology to Test Web Application

server failures or the other kinds of failures, web application will not response any more and we have to change the server which host the web application. So, when any kind of network failure occurs, the bandwidth of the network will be less and less till the link is dropped at all. We used Charles proxy to introduce this failure in the bandwidth, and used it to catch the values of performance metrics for web application such as response speed and time, the latency,...etc. RT for WA is the time taken to browse a web page. Latency is the amount of time it takes a packet to travel from source to destination. One of the metrics we will focus on, it is the request and response time which is the time to get (HTTP GET) the pages from web server or the post to web server (HTTP POST/PUT).

Table (7.1) summarizes our experiments, where the performance of web application is represented by response time and the value of bandwidth in KB/s. We also plotted the web application performance degradation represented by response time and bandwidth in Fig. 7.2 .

Table 7.1: Web Application Real Experiments

Bandwidth KB/s	***Response Speed KB/s***	***Response Time (sec.)***	***Latency (msec.)***	***Speed KB/s***
1	0.06	16.666	8000	0.05
5	0.2	5	7300	0.12
10	0.8103	1.234	2040	0.72
20	2.5125	0.398	1938	1.48
28.8	3.704	0.27	1840	1.67
33.6	4.15	0.241	1280	2.09
57.6	7.3	0.137	690	2.35
64	11.161	0.0896	579	5.67
128	22.83	0.0438	523	7.24
256	44.053	0.0227	517	9.23
1024	127.877	0.00782	1.74	12.43

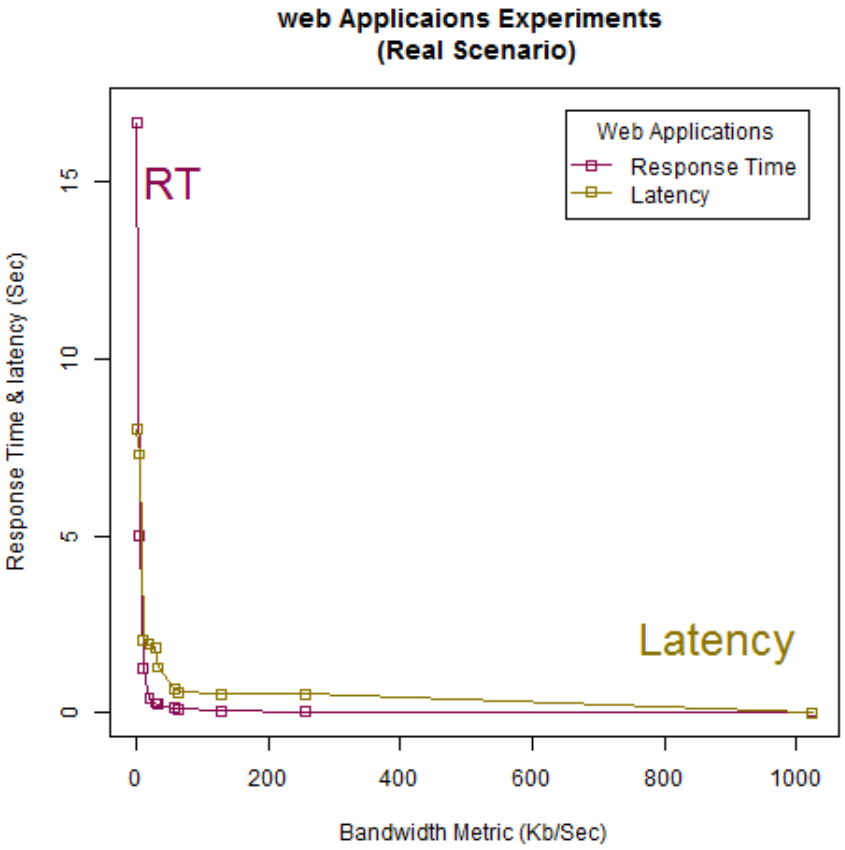


Figure 7.2: Web Application Response Time & Latency Metrics

7.1.2 Mathematical Model

There are many model for testing web application such as modeling web application by using Finite State Machines (FSMs) as in [86], modeling web application by using State Charts as in [79] and also the model for testing web applications by using NModel as in [85].

In this thesis work, from the previous experiments for testing web application and by using the graph in Fig. 7.2 which represents the performance of web application, we introduce a mathematical based model for testing the performance of web application. The model is based on the value of bandwidth represented in X and from this value we can get the value of the response time represented on Y. The value of response time indicates the performance of web application, if it high or low. From the curve line in Fig. 7.2, we got the equation represent this curve line which is the mathematical equation for the model used for testing web application. Here it is the power equation for the mathematical model:

$$Y = 17.912 * X^{-1.196}$$

X is the value of Bandwidth, and Y is the response time value. If you have the value of bandwidth you can get approximate value for the response time, and from this we can know the performance of web application. With this evaluation to the performance of WA we can see how the SLA QoS will be degraded if three is any failures. And as in SLA document, response time for any critical failure should be less to guarantee a high

performance for services (applications) so we should solve those failures. In the next section, we test file application.

7.2 File Application Testing

File application is one of the recently used applications over the world. It is the application of exchanging files between source and destination through the cloud. File application is constructed with file server hosted on the cloud and file client who is the cloud user. The cloud user can upload and download files to/from file server hosted on cloud data center from his personal computer/ to his computer. The testing scenario is in Subsection 7.2.1 and the mathematical based model for testing FA is in Subsection 7.2.2.

7.2.1 Testing Scenario

The file client ask file server to upload or download by using FTP PULL or PUSH. The file server response by downloading the requested file or by acknowledgment for the uploaded files. The cloud user can access files even online or locally after downloading, and he can upload files to server as archive and access later time.

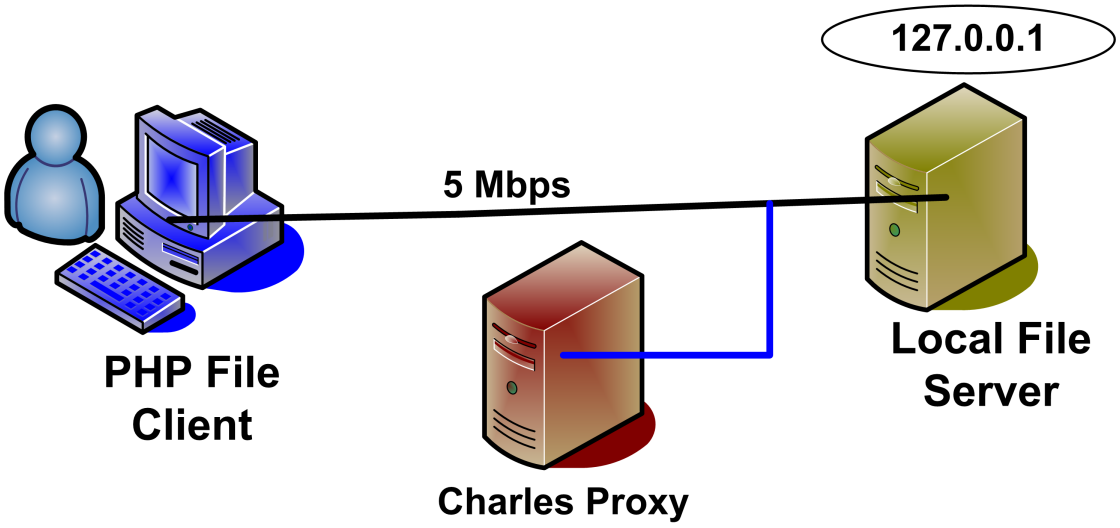


Figure 7.3: Real Scenario Topology to Test File Application

To test file application, we developed a file server and a web interface to access this file server by using PHP & HTML. This web interface is used to upload and download the files from the file server. Fig. 7.3 shows the topology we design to test real file application. We used this file application and file server to test the performance metrics of file application in case we introduce failures. In case of network failures, file application will operate but with a degraded performance. In case of server failures or the other kinds of failures, the file application will not response any more and we have to change the file server hosted on the cloud. So, when any kind of network failure occurs, the bandwidth of network will be degraded till the link is dropped at all. We used the Charles proxy to introduce this failure in the bandwidth, and used it to catch the values of performance metrics for file application such as the response speed and time for uploading and downloading, the latency,...etc. Two of the metrics we will focus on, they are the upload and

download response time which are the time to push (FTP PUSH) to the file server or the pull from the file server (FTP PULL).

Table 7.2: File Application Real Experiments

Bandwidth KB/s	***Upload Response Time (sec.)***	***Download Response Time (sec.)***	***Upload Latency (msec.)***	***Download Latency (msec.)***
1	20	11.111111	422	400
5	3.703704	2.0408163	21	20
10	1.428571	1.0309278	32	31
20	0.531915	0.390625	31	39
28.8	0.362319	0.1618123	127	130
33.6	0.306748	0.0067222	125	125
57.6	0.175131	0.0047517	125	125
64	0.116279	0.0044234	125	125
128	0.062696	0.0038058	146	149
256	0.02745	0.0022555	156	160
1024	0.02495	0.0021332	312	315

Table (7.2) summarizes our experiments, where the performance of file application is represented by the download/upload response times and the value of the bandwidth in KB/s. We also plotted file application performance degradation represented by response time (for upload/download) and bandwidth in Fig. 7.4.

7.2.2 Mathematical Model

In this thesis work, from the previous experiments of testing file application and by using the graph in Fig. 7.4 which represents the performance of file application, we introduce a mathematical based model for testing the performance of file application. For file application, we introduce two model based one for testing the upload to file server and the second for testing download from file server. The models are based on the value of bandwidth represented in X and from this value we can get the value of response time (even for uploading or downloading) represented on Y. The value of response time indicates the performance of file application, if it high or low. From the two curve lines in Fig. 7.4, we got the equations represent those curves line which are the mathematical equations for model used for testing file application. Here it is the power equations for the mathematical model, For uploading:

$$Y = 14.739 * X^{-1.061}$$

For downloading:

$$Y = 10.752 * X^{-1.509}$$

X is the value of bandwidth, and Y is the response time (even for upload or download)value. If you have the value of bandwidth you can get approximate value for the response time for upload and download, and from this we can know the performance of the file application. With this evaluation to the performance of FA we can see how the SLA QoS will be degraded if three is any failures. And as in SLA document, response

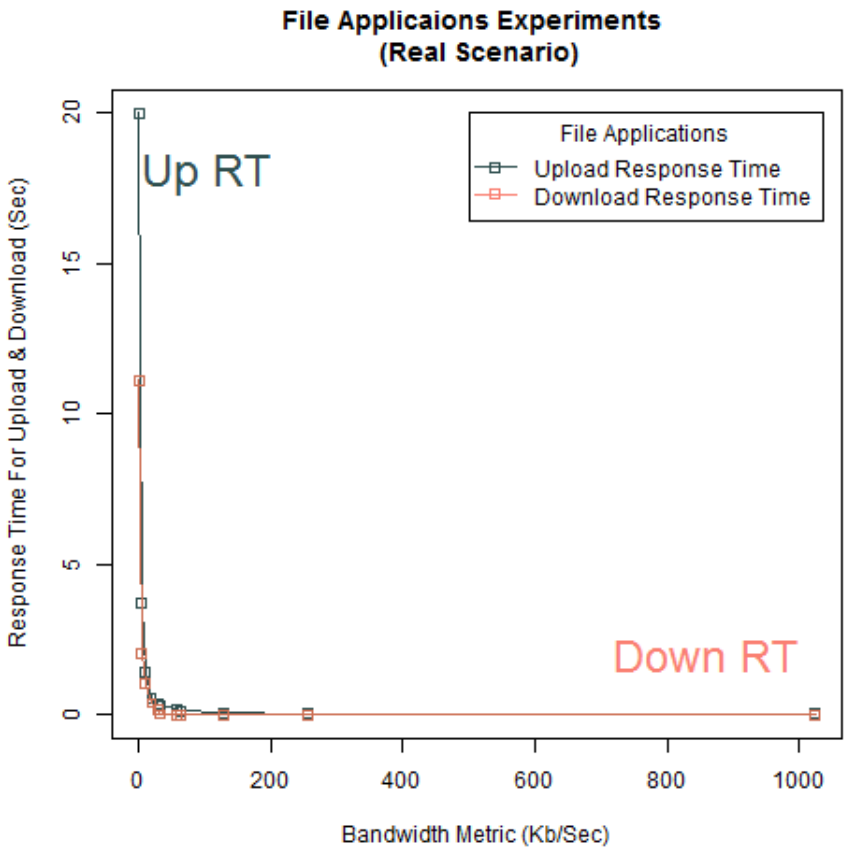


Figure 7.4: File Application Upload & Download Response Time Metrics

time for any critical failure should be less to guarantee a high performance for services (applications) so we should solve those failures. In the next section, we test distributed application.

7.3 Distributed Application Testing

Distributed applications is also one of the widely used applications over the world. In distributed application, The cloud application functionality divides among multiple application components that can be scaled out independently. Distributed application represented in e-mail servers putted all over the world. So in this thesis work we used an e-mail application as an example for distributed application. The e-mail application constructed with an e-mail server (Mail delivery agent or Mail transfer agent) and an e-mail client (Mail user agent). The Mail user agent on the cloud can send e-mails and receive e-mails in his mail box. The testing scenario is in Subsection 7.3.1 and the mathematical based model for testing DA is in Subsection 7.3.2.

7.3.1 Testing Scenario

The e-mail client ask email server (mail transfer agent) to send emails by using SMTP protocol. The mail server response by transferring the desired email to other email server that represent the mail delivery agent. Then, email server (mail delivery agent) deliver

the desired email to the desired mail user agent by using the POP3 protocol. The cloud user can access his emails or send new ones online from a web interface for the mail user agent any time from any where.

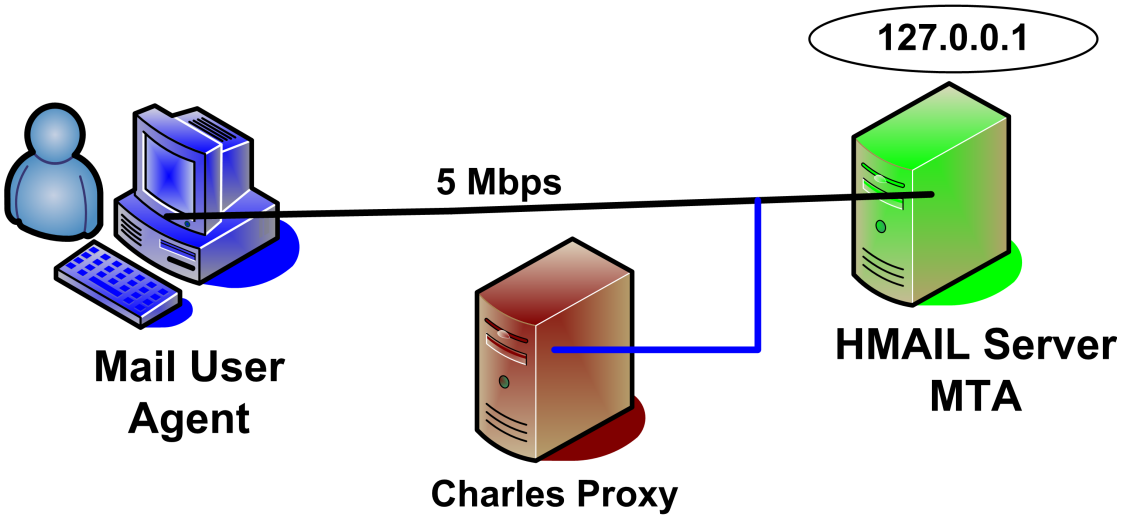


Figure 7.5: Real Scenario Topology to Test Distributed Application

To test distributed application, we setup an email server and a web interface to access this email server by using PHP & HMAIL Server[1]. This web interface is used to send and explore emails from the email server. Fig. 7.5 shows the topology we designed to test real DA. We used this email application and email server to test the performance metrics of distributed application in case we introduce failures. In case of network failures, distributed application will operate but with a degraded performance. In case of server failures or the other kinds of failures, distributed application will not response any more and we have to change the email server hosted on the cloud. So, when any kind of network failure occurs, the bandwidth of network will be degraded till the link is dropped at all. We used the Charles proxy to introduce this failure in the bandwidth, and used it to catch the values of performance metrics for distributed application such as the response speed and time for sending emails, latency, delay time...etc. DT is the time taken to deliver or receive an email from email user to another. Latency is the amount of time it takes a packet to travel from source to destination. One of the metrics we will focus on, it is the delay time which is the time delayed to receive emails.

Table (7.3) summarizes our experiments, where the performance of distributed application is represented by delay time and the value of bandwidth in KB/s. We also plotted distributed application performance degradation represented by delay time and bandwidth in Fig. 7.6.

7.3.2 Mathematical Model

There are many models for testing distributed application such as testing the distributed systems by using an accurate scale model as in [88], testing the distributed application by developing an integrated testing environment as in [82] and also the model based testing for distributed application by using the middleware software at early stages of a

[1] https://www.hmailserver.com/

Table 7.3: Distributed Application Real Experiments

Bandwidth KB/s	***Response Speed KB/s***	***Delay Time (sec.)***	***Latency (msec.)***	***Speed KB/s***
1	0.04	25	3740	0.4
5	0.3	3.333333	314	0.29
10	0.68	1.470588	308	0.66
20	1.37	0.729927	312	1.36
28.8	3.8	0.263158	314	2.48
33.6	4.42	0.226244	326	2.78
57.6	7.61	0.131406	329	3.23
64	11.04	0.09058	92	7.98
128	22.21	0.045025	151	12.37
256	44.16	0.022645	190	17.01
1024	175.4	0.005701	518	5.92

development process of the application as in [89].

In this thesis work, from the previous experiments of testing distributed application and by using the graph in Fig. 7.6 which represents the performance of distributed application, we introduce a mathematical based model for testing performance of distributed application. The model is based on the value of bandwidth represented in X and from this value we can get the value of delay time represented on Y. The value of the delay time indicates the performance of distributed application, if it high or low. From the curve line in Fig. 7.6, we got the equation represent this curve line which is the mathematical equation for the model used for testing distributed application (email application). Here it is the power equation for the mathematical model:

$$Y = 22.578 * X^{-1.249}$$

X is the value of bandwidth, and Y is the delay time value. If you have the value of bandwidth you can get approximate value for delay time, and from this we can know the performance of the distributed application. With this evaluation to the performance of DA we can see how the SLA QoS will be degraded if three is any failures. And as in SLA document, response time for any critical failure should be less to guarantee a high performance for services (applications) so we should solve those failures. In the next section, we test real time application.

7.4 Real Time Application Testing

Real time application is the most used application nowadays. It is a class of cloud applications that works in a time frame fashion which means that the user senses as instantly or currently. Real time application is like the conversational services such as the video and voice calls and chatting application. In the real experiments, we use the chatting application as an example for real time application and for simulation phase we simulate voice and video applications. The testing scenario is in Subsection 7.4.1 and the mathematical based model for testing RTA is in Subsection 7.4.2.

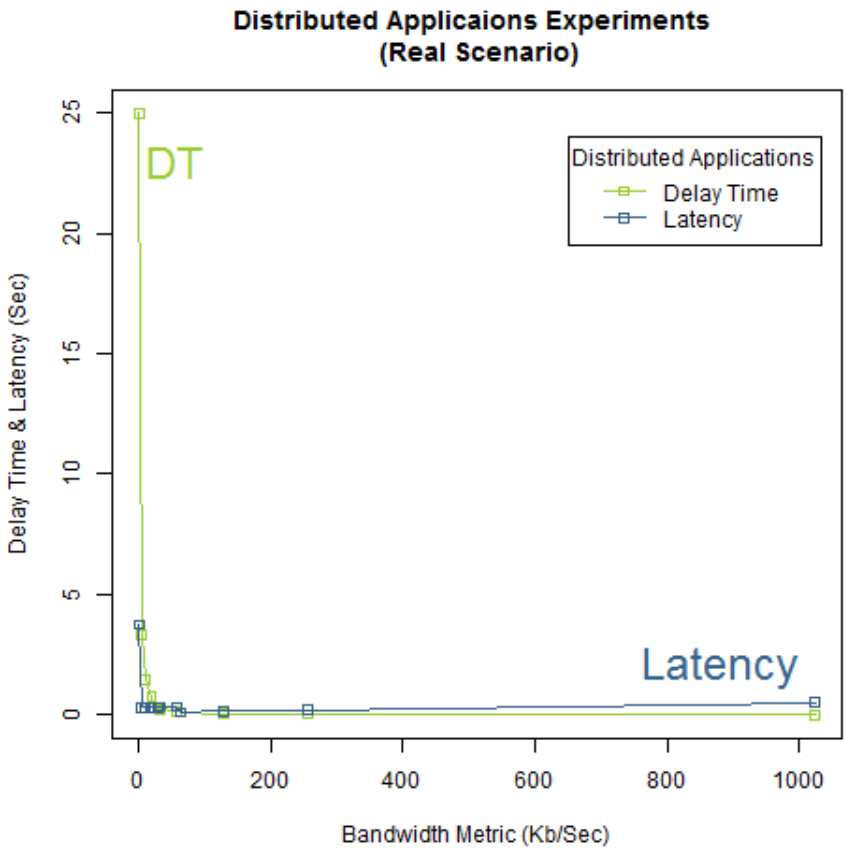


Figure 7.6: Distributed Application Delay Time & Latency Metrics

7.4.1 Testing Scenario

The chatting application is one of the conversational systems which belong to the real time applications. we used chatting application to test real time application. To test real time application, we setup a chatting server and a web interface to access this chat server by using PHP & PHP chat Server[1]. This web interface is used to do chatting. Fig. 7.7 shows the topology we designed to test real RTA. We used this chatting application and the chat server to test performance metrics of real time application in case we introduce failures. In case of network failures, real time application will operate but with a degraded performance. In case of server failures or the other kinds of failures, the real time application may be response also or we can do server mitigation to change the chat server hosted on cloud. So, when any kind of network failure occurs, the bandwidth of network will be degraded till the link is dropped at all. We used the Charles proxy to introduce this failure in bandwidth, and used it to catch the values of performance metrics for real time application such as response time, latency, delay time, throughput,...etc. One of the metrics we will focus on, it is the delay time which is the delayed time consumed till the conversation be delivered to the other user . Latency refers to a delay in packet delivery.

Table (7.4) summarizes our experiments, where the performance of real time application is represented by delay time and the value of bandwidth in KB/s. We also plotted real time application performance degradation represented by delay time and bandwidth

[1] http://www.phpfreechat.net/overview

Table 7.4: Real-Time Application Real Experiments

Bandwidth KB/s	*Response Speed KB/s*	*Delay Time (msec.)*	*Latency (msec.)*	*Response Time (sec.)*	*Speed KB/s*
1	0.08	18124	23.09	12.5	0.04
5	0.48	4557	3.37	2.083333	0.34
10	0.98	1873	3.27	1.020408	0.74
20	12.82	911	3.2	0.078003	1.54
28.8	18.29	756	3.06	0.054675	1.88
33.6	20.75	685	2.92	0.048193	1.98
57.6	34.98	642	2.33	0.028588	2.04
64	56.23	316	1.28	0.017784	4.91
128	103.58	210	1.1	0.009654	6.23
256	129.46	186	0.364	0.007724	8.57
512	201.28	785	0.352	0.004968	1.64

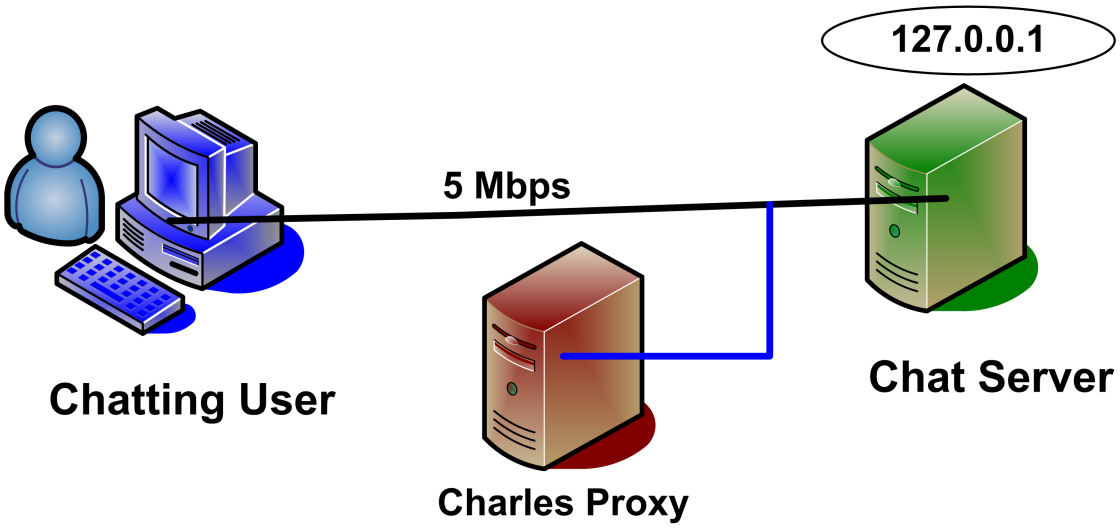


Figure 7.7: Real Scenario Topology to Test Real Time Application

in Fig. 7.8.

7.4.2 Mathematical Model

In this thesis work, from the previous experiments of testing real time applications and by using the graph in Fig. 7.8 which represents the performance of real time application, we introduce a mathematical based model for testing the performance of real time application. The model is based on the value of the bandwidth represented in X and from this value we can get the value of delay time represented on Y. The value of delay time indicates the performance of real time application, if it high or low. From the curve line in Fig. 7.8, we got the equation represent this curve line which is the mathematical equation for the model used for testing of the real time application (chatting application). Here it is the power equation for the mathematical model:

$$Y = 9.3583 * X^{-0.657}$$

X is the value of bandwidth, and Y is the delay time value. If you have the value

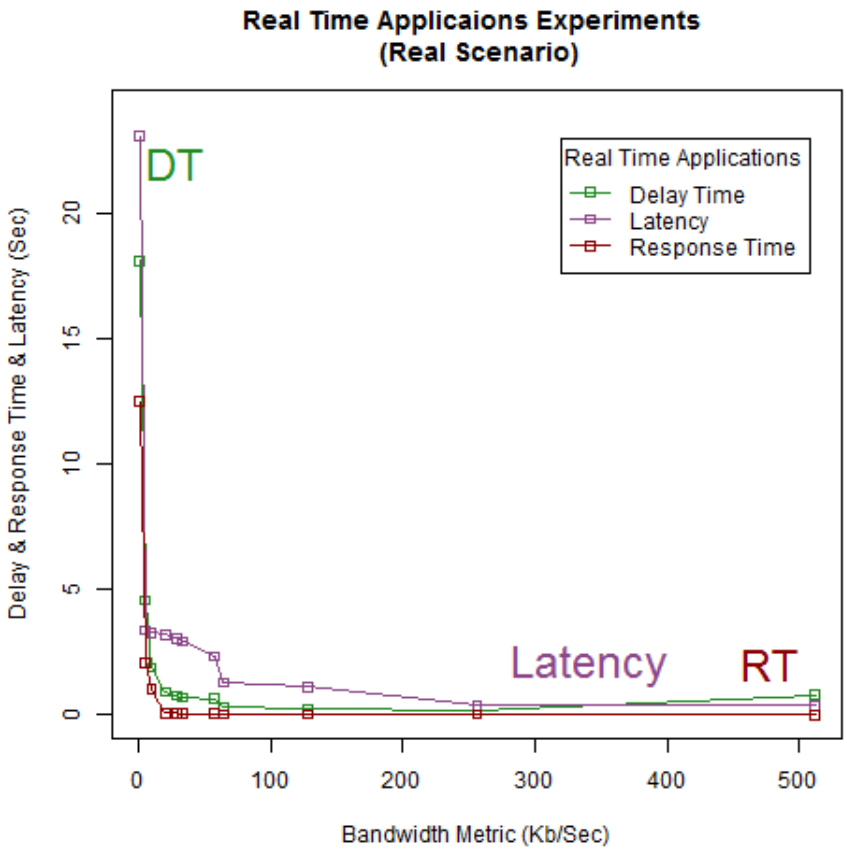


Figure 7.8: Real Time Application Delay, Response Time& Latency Metrics

of bandwidth you can get approximate value for delay time, and from this we can know the performance of real time application. With this evaluation to the performance of RTA we can see how the SLA QoS will be degraded if three is any failures. And as in SLA document, response time for any critical failure should be less to guarantee a high performance for services (applications) so we should solve those failures.

In the next chapter, we will discuss our experiments and simulation for application classes on NS-2 simulator. We will show how we will mitigate the failures to keep the applications performance in acceptable state. The failure mitigation will help us to assure the SLA QoS and services performance.

Chapter VIII

Simulator Experiments

"When you follow two separate chains of thought, you will find some point of intersection which should approximate to the truth."

Sherlock Holmes- The Disappearance of Lady Frances Carfax

In this chapter, we review the testing of cloud applications with introducing failures but now in NS-2 simulator. We discuss how we are testing cloud applications in the simulator. We use NS-2 simulator with TCL Scripts to build topology for testing cloud application classes. We use network layer and transport layer protocols to simulate applications. We introduce failures during the simulation such as network failures or error bit rate,..etc. Inside the simulation, we can compute more metrics for the application and introduce different kinds of failures. We simulate five classes of application, WA, FA, DA, HIA and RTA. The simulating and testing of the five classes are in first Section 8.1. After simulating cloud application classes and see what is the effect of failures on the performance of application classes. We do an additional step which not be done in real experiments. The additional step is the solving and mitigating the failures introduced to applications. We use mitigation techniques to solve failures, in our work we use VM migration as the mitigation technique to mitigate the failures. We implement VM migration as a failure mitigation technique in Section 8.2. In the last Section 8.3, we mitigate the BER failure which introduced to real time application (voice, video) and highly interactive application by using FEC. After mitigating the failures, we check on the performance of applications again.

8.1 Cloud Application Simulations with Failures

In this section, we show how we do the simulation experiments for the five classes of application. Also we show how we introduce failures in the simulation scenarios. Then, we calculate the different important metrics values for each class of applications. From the metrics values we can see the performance of applications as we did in the real experiments. Simulation and testing of web application are in first Subsection 8.1.1. Testing and simulation of file application are in Subsection 8.1.2. In Subsection 8.1.3, we discuss the testing and simulation of distributed application. Then the testing and simulation of highly interactive application are in Subsection 8.1.4. Finally, the testing and simulation of real time application are in last Subsection 8.1.5.

8.1.1 Web Application Simulating

Web application is depend on HTTP protocol with HTTP GET for request a web page and HTTP POST/PUT to upload web page or any control information. To simulate and test web application in NS-2, we simulate HTTP protocol over TCP protocol where TCP is a reliable and acknowledgment transport protocol, this by using web cache (HTTP/-Cache) as an application, TCP agent and TCP sink agent. We construct a topology to test web application. The topology is consisting of web server, web client and a cache server (memory) as a temporal server to get the web pages from the server to web client. The topology of testing web application in NS-2 is in Fig. 8.1.

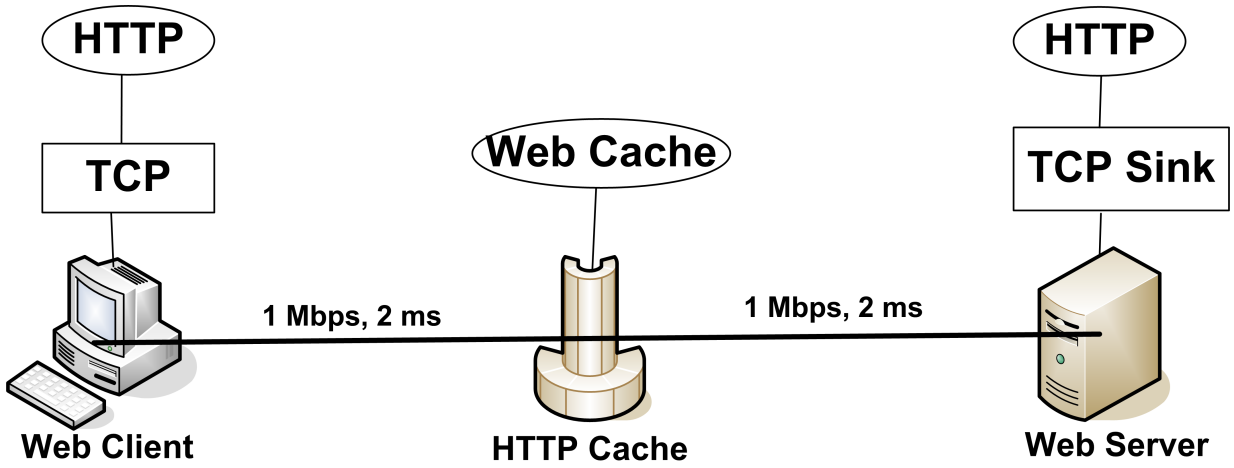


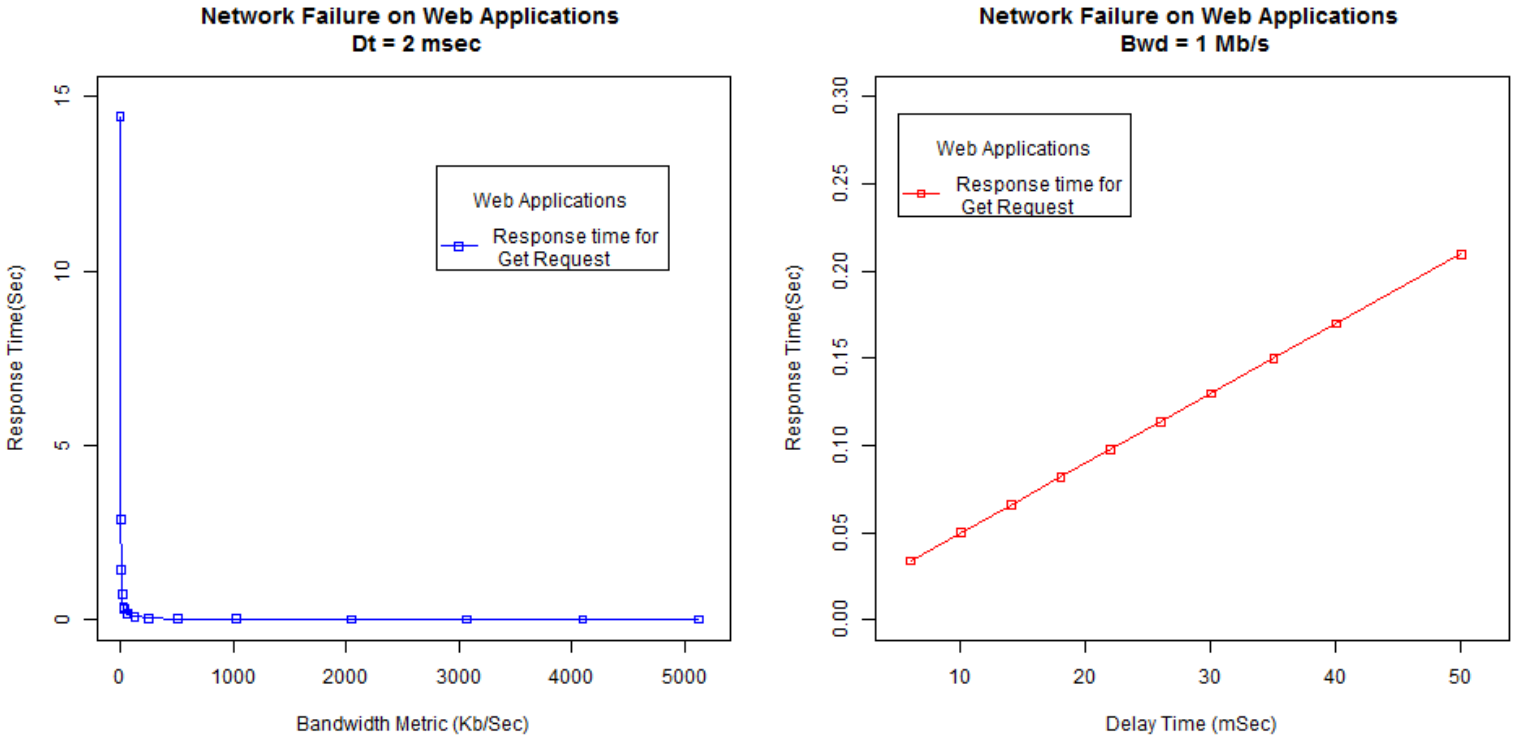
Figure 8.1: Web Application Testing Topology

We use NS-2 network link properties to introduce failures like degradation in the bandwidth of the link, the delay time on the link and the bit error rate on the link. Then we use trace files and the HTTP log files to calculate the important metrics such as the response time, latency, throughput,..etc. We calculate each metric one time in each case of failures cases (e.g. bandwidth degradation, increasing of delay time or BER). In case of web application, we just calculate the response time in the two cases of failures, the bandwidth degradation and delay time increasing as a network failures. We don't introduce the bit error rate as a failure to web application where it have no effect on the performance. We use Perl scripting language to analysis the trace and log files and calculate response time. RT is the time taken when request page till the page is browsed.

Table (8.1) summarizes the output of our experiments for web applications simulating. We use the values of the response time in the two cases of failures to evaluate the performance of the web applications. And this by plotting bandwidth (in KB/s) or delay time (in milliseconds) with the response time. So, web application performance degradation is represented by response time and bandwidth or delay time as in Fig. 8.2a & Fig. 8.2b .

Table 8.1: Web Application Experiments Summary on NS-2

Bandwidth KB/s	*Response Time (sec.)*	*Delay Time (mSec.)*	*Response Time (sec.)*
1	14.428	***6***	0.033816
5	2.8888	***10***	0.049816
10	1.4464	***14***	0.065816
20	0.7252	***18***	0.081816
28.8	0.348833	***22***	0.097816
33.6	0.300143	***26***	0.113816
57.6	0.181427	***30***	0.129816
64	0.161375	***35***	0.149816
128	0.084687	***40***	0.169816
256	0.046344	***50***	0.209816



(a) RT with Bandwidth

(b) RT with Delay

Figure 8.2: Web Application Performance with Failures

8.1.2 File Application Simulating

File application is depend on FTP protocol with FTP PULL for download a file and FTP PUSH to upload file or any control information. To simulate and test file application in NS-2, we simulate FTP protocol over TCP transport layer protocol where TCP is a reliable and acknowledgment transport protocol, this by using FTP application, TCP agent and TCP sink agent. We construct a topology to test file application. The topology is consisting of two nodes file server and file client and connect them with a link. Both file server node and file client node have FTP application above TCP agent on the client as a requester (for upload or download) and above TCP sink agent on file server as a receiver(response by acknowledgment in case of upload or by file in case of download request). The topology of testing file application in NS-2 is in Fig. 8.3.

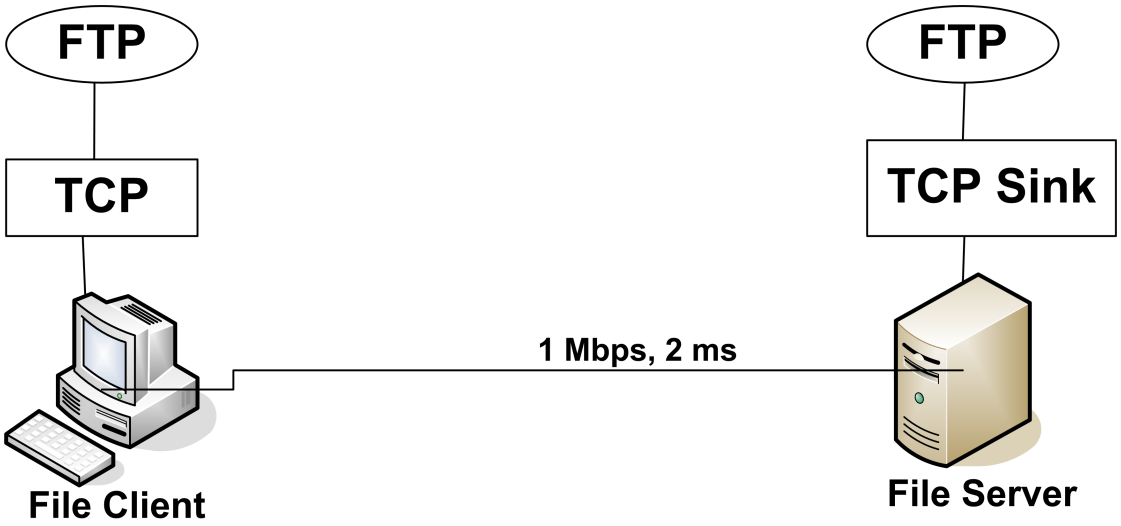


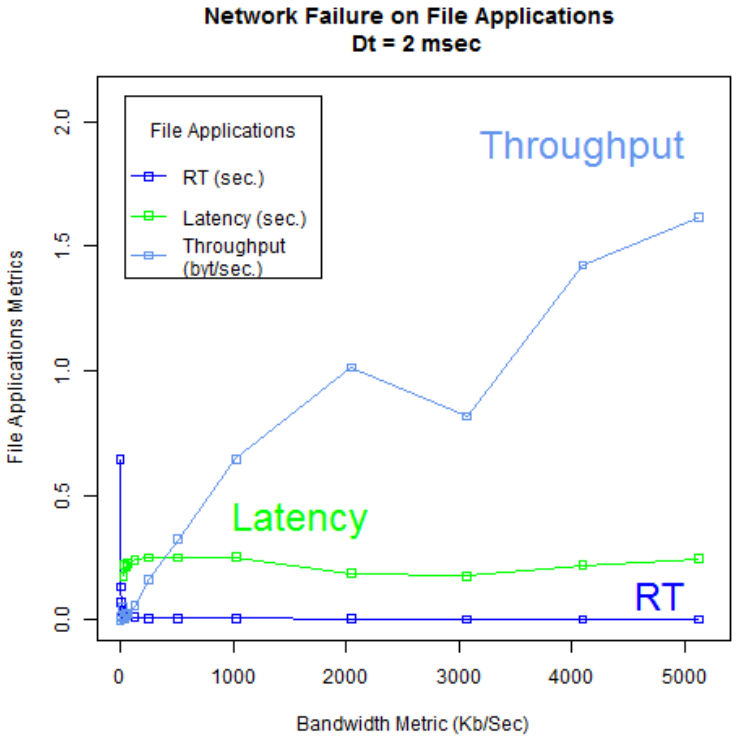
Figure 8.3: File Application Testing Topology

We use NS-2 network link properties to introduce failures like degradation in the bandwidth of the link, the delay time on the link and the error bit rate on the link. Then we use trace files and TCP log files to calculate the important metrics such as the response time, latency, throughput,..etc. We calculate each metric one time in each case of failures cases (e.g. bandwidth degradation, increasing of delay time or BER). In case of file application, we just calculate the response time, latency and throughput in the two cases of failures, the bandwidth degradation and the delay time increasing as a network failures. We don't introduce BER as a failure to file application where it has no effect on the performance. We use Perl scripting language to analysis the trace and log files and calculate the response time, latency and network throughput. Throughput is a measure of how many packets on network can process in a given amount of time. Latency is the amount of time it takes a packet to travel from source to destination.

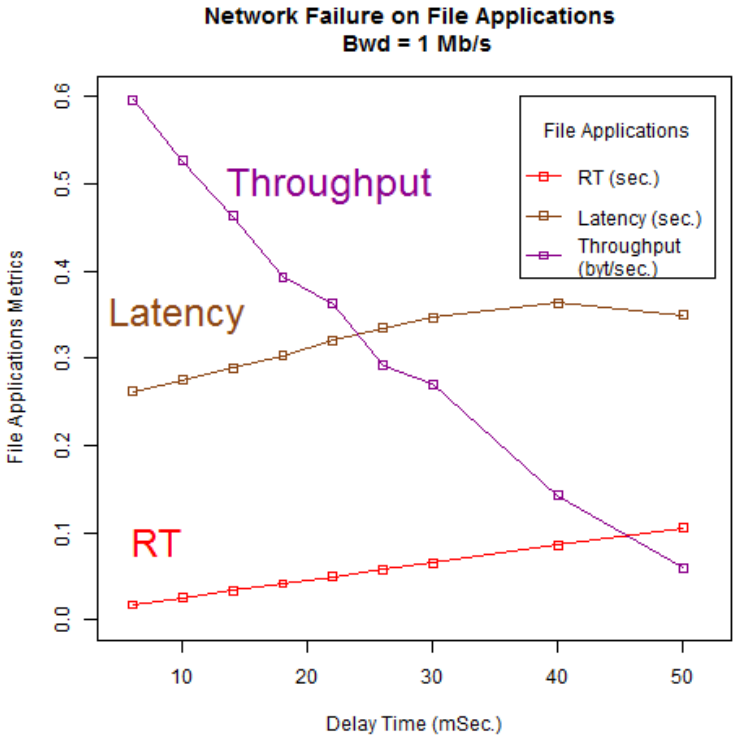
Table (8.2) summarizes the output of our experiments for file application simulating. We use the values of the response time in the two cases of failures to evaluate the performance of file application. And this by plotting bandwidth (in KB/s) or delay time (in milliseconds) with response time, latency and throughput. So, file application performance degradation is represented by response time, latency and throughput and bandwidth or delay time in Fig. 8.4a & Fig. 8.4b.

Table 8.2: File Application Experiments Summary on NS-2

Bandwidth KB/s	*Response Time (sec.)*	*Latency (sec.)*	*Throughput (byte/sec.)*	*Delay Time (mSec.)*	*Response Time (sec.)*	*Latency (sec.)*	*Throughput (byte/sec.)*
1	0.644	0.644	0.000112	*6*	0.017248	0.262143	0.596177
5	0.132	1.879636	0.0094	*10*	0.025248	0.274729	0.52702
10	0.068	1.857137	0.014033	*14*	0.033248	0.288549	0.463191
20	0.0304	2.013051	0.04961	*18*	0.041248	0.303121	0.393264
28.8	0.026222	0.174603	0.002604	*22*	0.049248	0.320361	0.36298
33.6	0.023048	0.217429	0.005311	*26*	0.057248	0.334324	0.292074
57.6	0.015307	0.212016	0.016546	*30*	0.065248	0.346967	0.27008
64	0.014	0.224	0.022998	*35*	0.078659	0.353681	0.19583
128	0.009	0.240983	0.054869	*40*	0.085248	0.363599	0.142433
256	0.0065	0.249855	0.161577	*50*	0.105248	0.349243	0.059194



(a) RT, Latency & Throughput with Bandwidth



(b) RT, Latency & Throughput with Delay

Figure 8.4: File Applications Performance with Failures

8.1.3 Distributed Application Simulating

Distributed application is like the email application, so we will simulate an email application. The email application is depend on SMTP protocol with mail transfer agent for sending an email and mail delivery agent to deliver emails which hold POP3 protocol. To simulate and test distributed application in NS-2, we simulate SMTP and POP3 protocols over UDP transport layer protocol where UDP is a best effort and non acknowledgment transport protocol, this by using SMTP application, UDP agent and UDP sink agent. We construct a topology to test distributed application. The topology is consisting of two nodes, email server and email client and connect them with a link. The email client node have SMTP application above UDP agent on the client as a sender of emails. The email server node have an UDP Sink agent and SMTP application to transfer the emails to other servers. The topology of testing distributed application in NS-2 in Fig. 8.5.

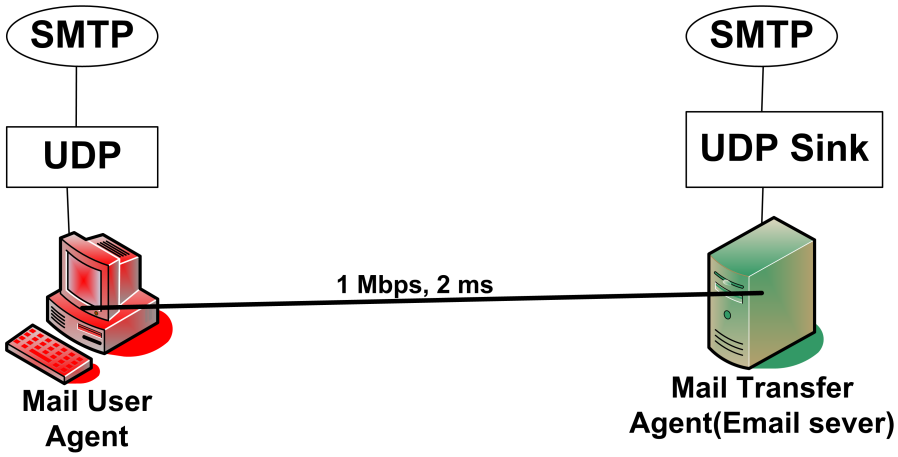


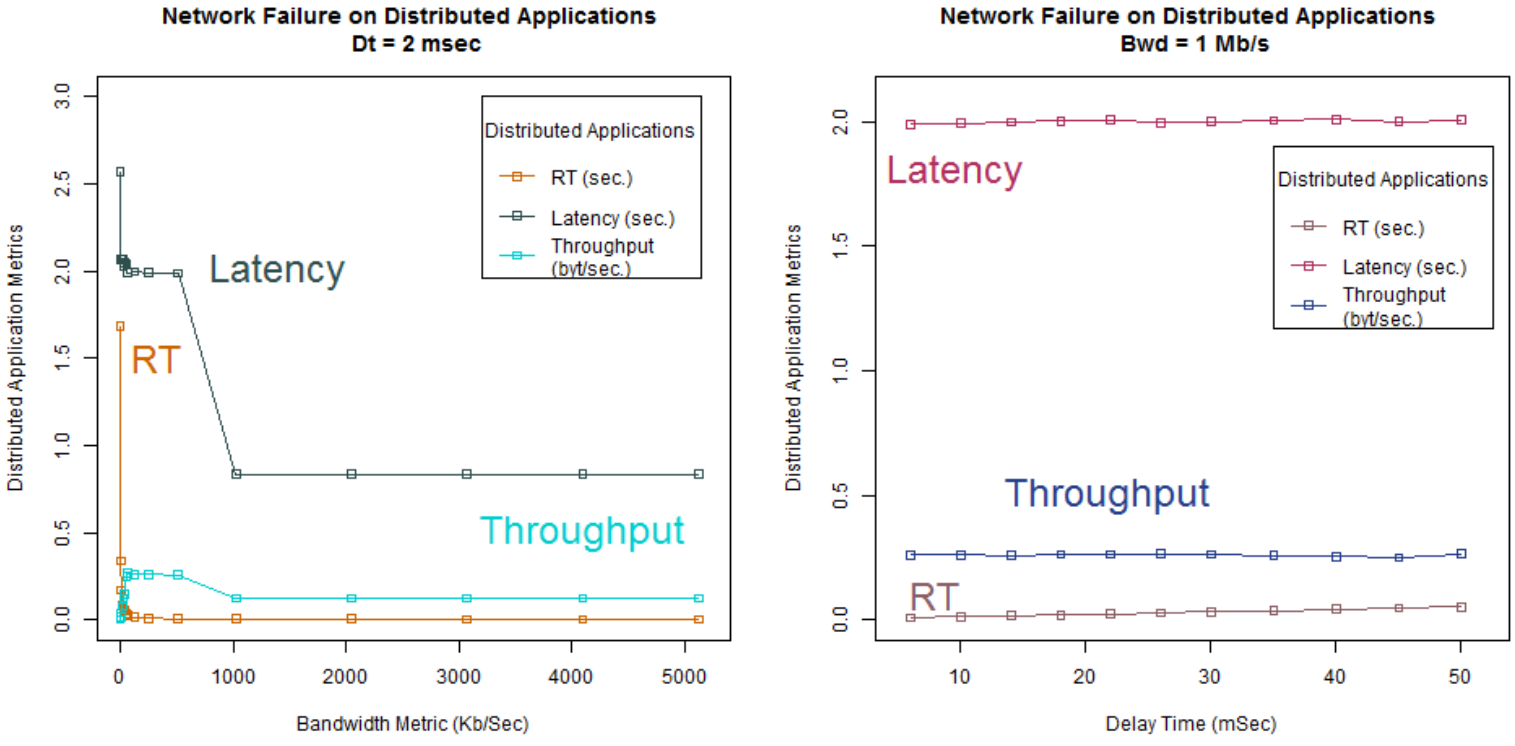
Figure 8.5: Distributed Application Testing Topology

We used NS-2 network link properties to introduce failures like degradation in the bandwidth of the link, the delay time on the link and the error bit rate on the link. Then we use trace files and the UDP log files to calculate the important metrics such as the response time (Here is message delivery time), latency, throughput,..etc. We calculate each metric one time in each case of failures cases (e.g. bandwidth degradation, increasing of delay time or BER). In case of distributed application, we just calculate the response time, latency and throughput in the two cases of failures, the bandwidth degradation and the delay time increasing as a network failures. We don't introduce BER as a failure to distributed application where it has no effect on performance. We use Perl scripting language to analysis the trace and log files and calculate the response time, latency and network throughput.Throughput is a measure of how many packets on network can process in a given amount of time. Latency is the amount of time it takes a packet to travel from source to destination.

Table (8.3) summarizes the output of our experiments for distributed application simulating. We use the values of the response time in the two cases of failures to evaluate the performance of distributed application. And this by plotting bandwidth (in KB/s) or delay time (in milliseconds) with response time, latency and throughput. So, distributed application performance degradation is represented by response time, latency and throughput and bandwidth or delay time in Fig. 8.6a & Fig. 8.6b.

Table 8.3: Distributed Application Experiments Summary on NS-2

Bandwidth KB/s	*Response Time (sec.)*	*Latency (sec.)*	*Throughput (byte/sec.)*	*Delay Time (mSec.)*	*Response Time (sec.)*	*Latency (sec.)*	*Throughput (byte/sec.)*
1	1.682	2.56816	0.002637	*6*	0.00768	1.989895	0.260461
5	0.338	2.06416	0.018497	*10*	0.01168	1.993895	0.261419
10	0.17	2.06416	0.038523	*14*	0.01568	1.997895	0.258803
20	0.086	2.06416	0.077261	*18*	0.01968	2.001895	0.262563
28.8	0.060333	2.031493	0.122427	*22*	0.02368	2.005895	0.262244
33.6	0.052	2.04816	0.143476	*26*	0.02768	1.995965	0.263484
57.6	0.031682	2.036853	0.245223	*30*	0.03168	1.999965	0.262677
64	0.02825	1.989456	0.265	*35*	0.03668	2.004965	0.258011
128	0.015125	1.99734	0.258852	*40*	0.04168	2.009965	0.254536
256	0.008562	1.990778	0.260254	*50*	0.04668	2.000958	0.249114



(a) RT, Latency & Throughput with Bandwidth

(b) RT, Latency & Throughput with Delay

Figure 8.6: Distributed Application Performance with Failures

8.1.4 Highly Interactive Application Simulating

Highly interactive application is a new class of application we will test it in simulation testing for applications. HIAs are like office productivity (Online office), virtual reality and gaming online. HIAs are depend on the IRTP and RTP Transport protocols. To simulate and test high interactive application in NS-2, we simulate CBR traffic over RTP higher transport layer protocol where it based on the UDP transport protocol which is a best effort and non acknowledgment transport protocol. Simulation is done by using CBR traffic application and RTP sender and receiver agents. We construct a topology to test highly interactive application. The topology is consisting of two nodes sender and receiver and connect them with a link. The sender and receiver nodes have the CBR Traffic application above RTP agent. The topology of testing highly interactive application in NS-2 is in Fig. 8.7.

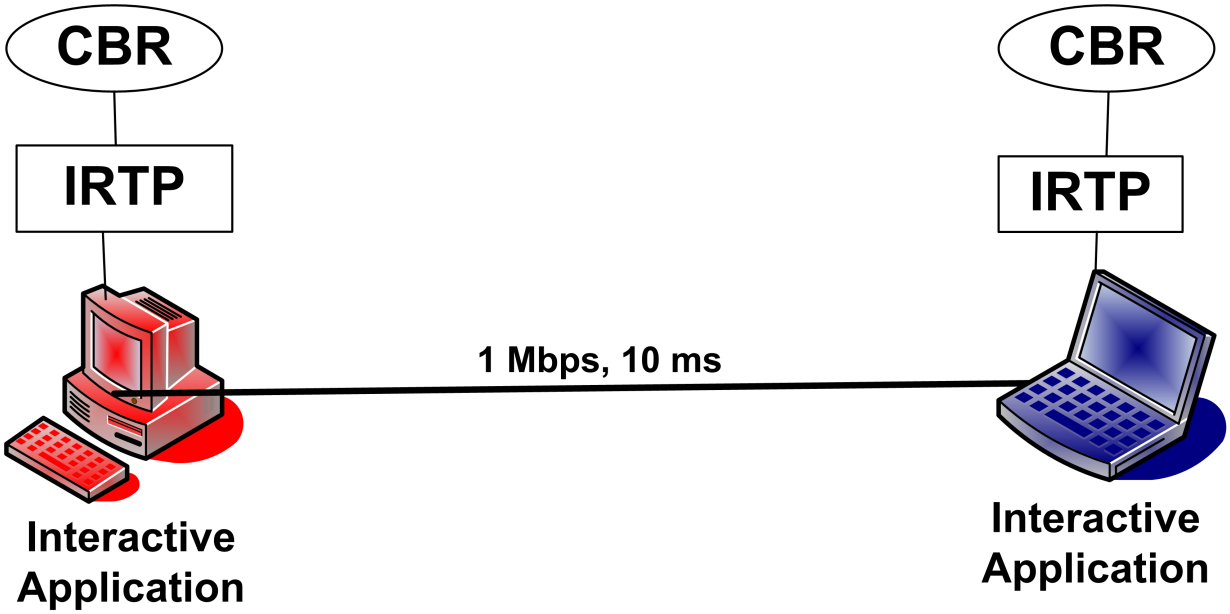
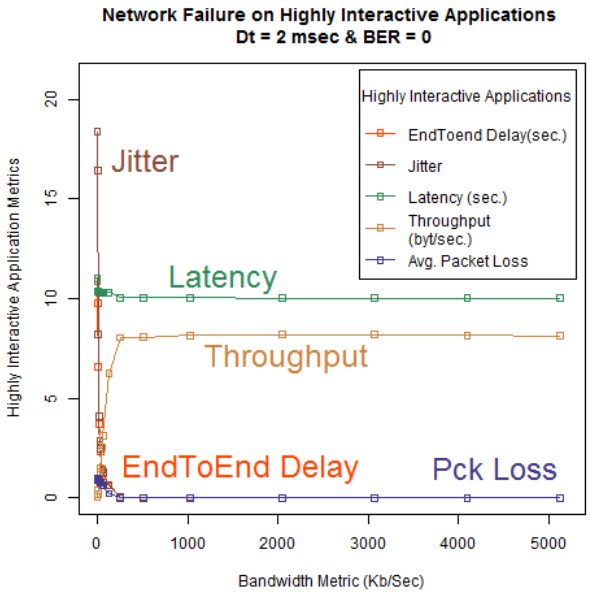


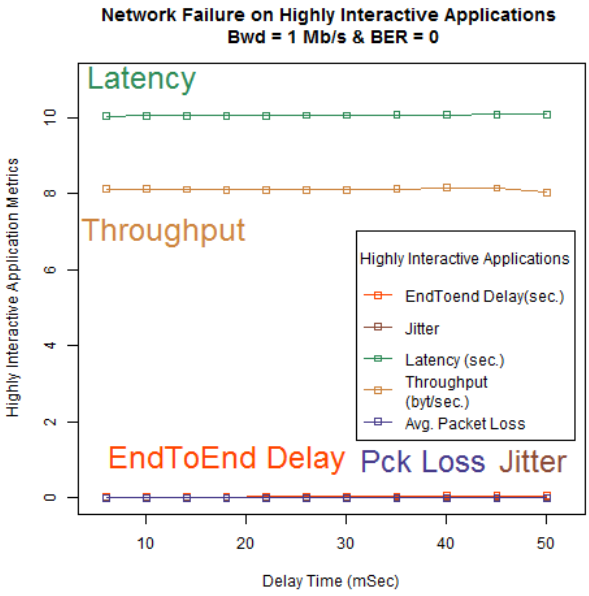
Figure 8.7: Highly Interactive Application Testing Topology

We use NS-2 network link properties to introduce failures like degradation in the bandwidth of the link, the delay time on the link and the error bit rate on the link. Then we use trace files to calculate the important metrics such as the response time, latency, throughput, end to end delay, jitter, packet loss..etc. We calculate each metric one time in each case of the failures cases (e.g. bandwidth degradation, increasing of delay time or BER). In case of highly interactive application, we just calculate the response time, end to end delay, jitter, packet loss, latency and throughput in the three cases of failures, the bandwidth degradation, error bit rate and delay time increasing as a network failures. We use Perl scripting language to analysis trace files and calculate response time, latency, end to end delay, jitter, packet loss and network throughput. Throughput refers to the total amount of data transmitted between two nodes. Latency is the amount of time it takes a packet to travel from source to destination. Packet loss is simply a measure of the amount of packets that are dropped. End-to-end delay is the time it takes for a packet to travel between two nodes. Jitter is is defined as the difference in end to end delay of the transmitted packets.

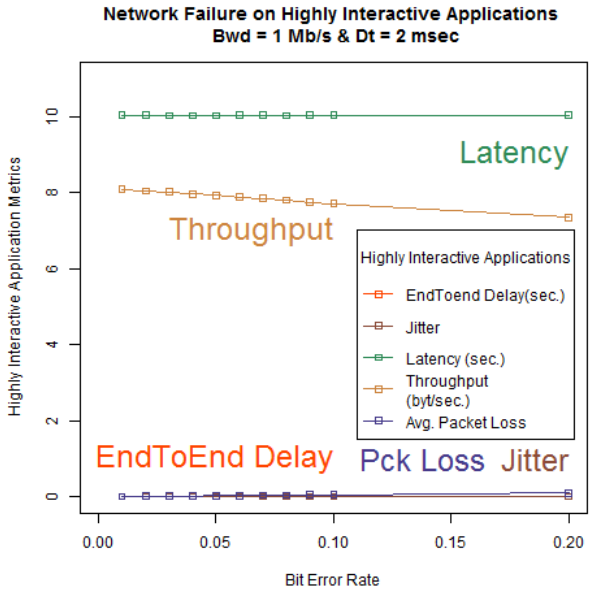
We use the values of the metrics in the three cases of failures to evaluate the performance of the highly interactive application. And this by plotting bandwidth (in KB/s), delay time (in milliseconds) or bit error rate value with response time, end to end delay, jitter, packet loss, latency and throughput. So, highly interactive application performance degradation is represented by response time, end to end delay, jitter, packet loss, latency and throughput and bandwidth, delay time or BER in Fig. 8.8a, Fig. 8.8b & Fig. 8.8c.



(a) Metrics with Bandwidth



(b) Metrics with Delay



(c) Metrics with Bit Error Rate

Figure 8.8: Highly Interactive Application Performance with Failures

8.1.5 Real Time Application Simulating

RTAs are a widely used applications. There are many sub classes of applications belong to RTAs class such as voice conversational application and video streaming. Also chatting application like what we tested in the real experiments. In the simulation of RTA, we will simulate the voice and video applications. The voice application is the VoIP application. VoIP can carry text, live video, images and high quality stereo sound in addition to the screen sharing and all these depend on speed and reliability of the Internet connection. The audio stream is divided into small pieces, each small piece is small enough to fit in a packet which is stamped with the destination address and sent through the network. The receiver should reconstruct the packets sequentially for ideal reproduction.

Voice over Internet protocol is a new technology that let users to make telephone calls using a broadband Internet connection instead of an analog phone line. There are many protocols used for video streaming and VoIP such as SIP, SCTP, RTP and RTCP. RTP & RTCP are working independently of the underlying transport and network layer. RTP & RTCP are a network protocol for delivering audio and video over IP networks. They are used extensively in communication and entertainment systems that involve streaming media such as telephony, video teleconference applications, television services and web-based push-to-talk features. RTP & RTCP run on the top of the UDP protocol.

There are many metrics which are important regards the performance of voice and video streaming applications. First metric is the QoS of the network which means that the packets of the voice or video have high priority in the network (Internet backbone), in our work we assume that this metric is verified by default. The second important metric is the data rated performance metrics such as throughput and mean packet delay. Also latency, end-to-end delay, Jitter and the packet loss rate are important metrics. Packet loss is when too much traffic in the network causes the network to drop packets. Latency is the delay for packet delivery. Throughput refers to the total amount of data measured in bytes or bit per seconds. End-to-end delay is the time used by the packet to travel from node to another node. Jitter is the variations in delay of packet delivery. Finally, the user perception or the QoE metric such as R Score and the MOS. The user perception metric is depend on the opinion of RTA users. The R Score capture the effect of mouth-to-ear delay and losses in the packet-switched network. The R Score is mapped to the MOS in RTA. MOS is the opinion of RTA users about the quality levels (e.g. Good, Poor).

8.1.5.1 VoIP Simulation & Test

To simulate and test voice application, we simulate VoIP protocol in NS-2. We use CBR and VBR applications traffic above to UDP and UDP sink agents in source and destination of voice topology. VBR is an exponential traffic which represent the other traffic types on the link. The topology of testing voice applications in NS-2 is in Fig. 8.9.

We use NS-2 network link properties to introduce failures like degradation in the bandwidth of the link, the delay time on the link and the bit error rate on the link. Then we use trace files to calculate the important metrics such as the response time, latency, throughput, end to end delay, jitter, packet loss..etc. We calculate each metric one time in each case of failures cases (e.g. bandwidth degradation, increasing of delay time or BER). In case of voice application we just calculate the response time, end to end delay, jitter, packet loss, latency and throughput in the three cases of failures, the bandwidth

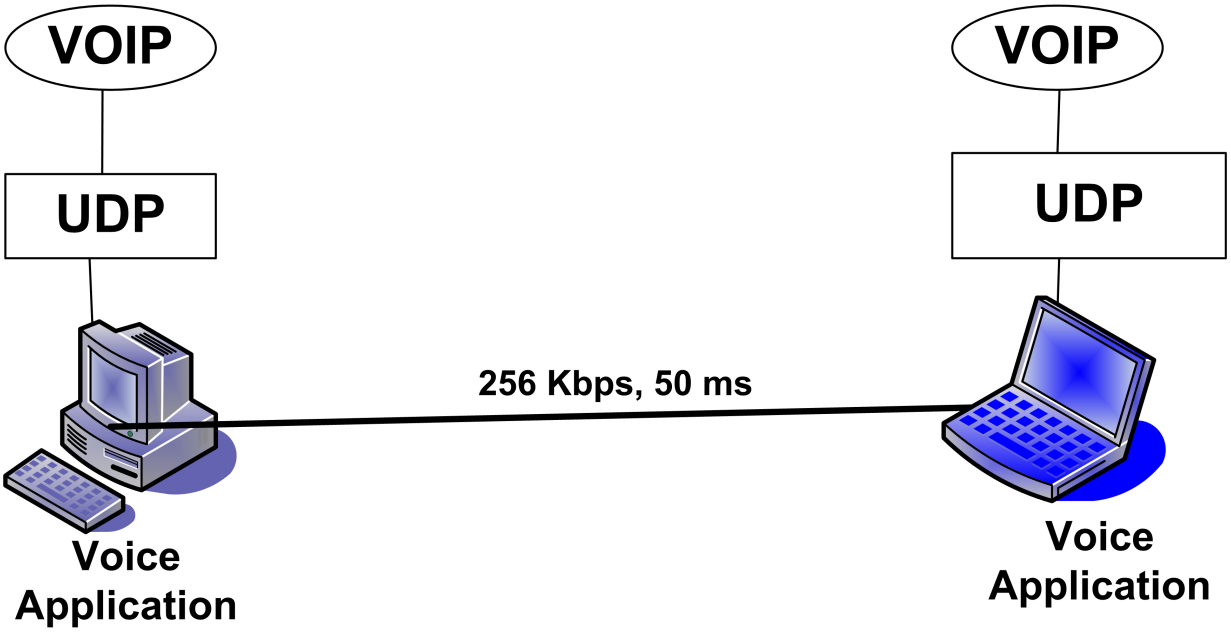
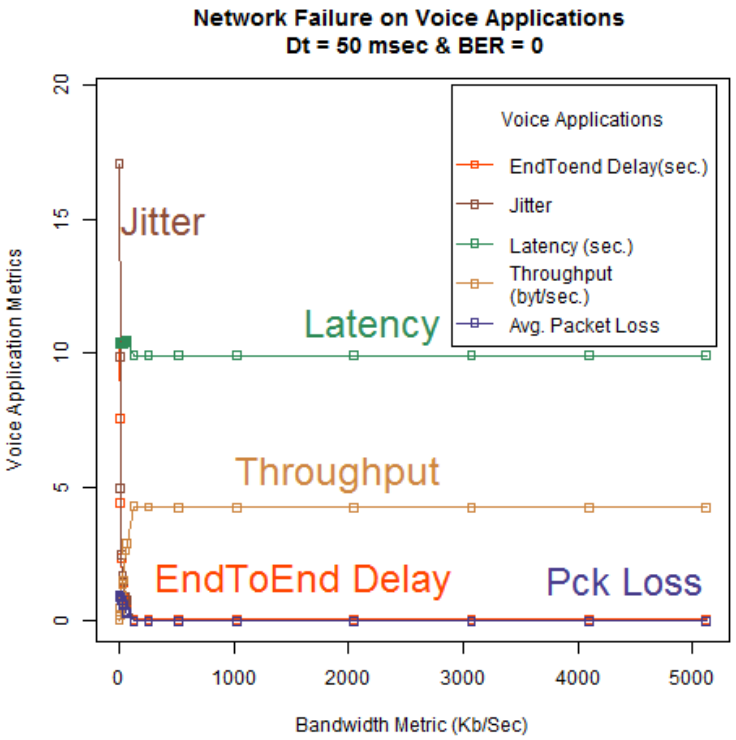


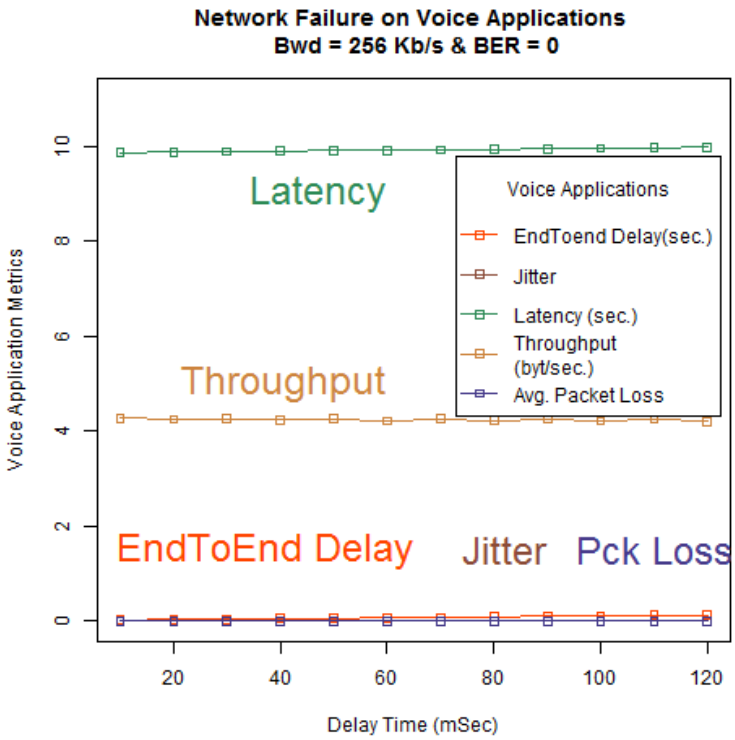
Figure 8.9: Voice Application Testing Topology

degradation, bit error rate and delay time increasing as a network failures. We use Perl scripting language to analysis the trace file and calculate the response time, latency, end to end delay, jitter, packet loss and network throughput.

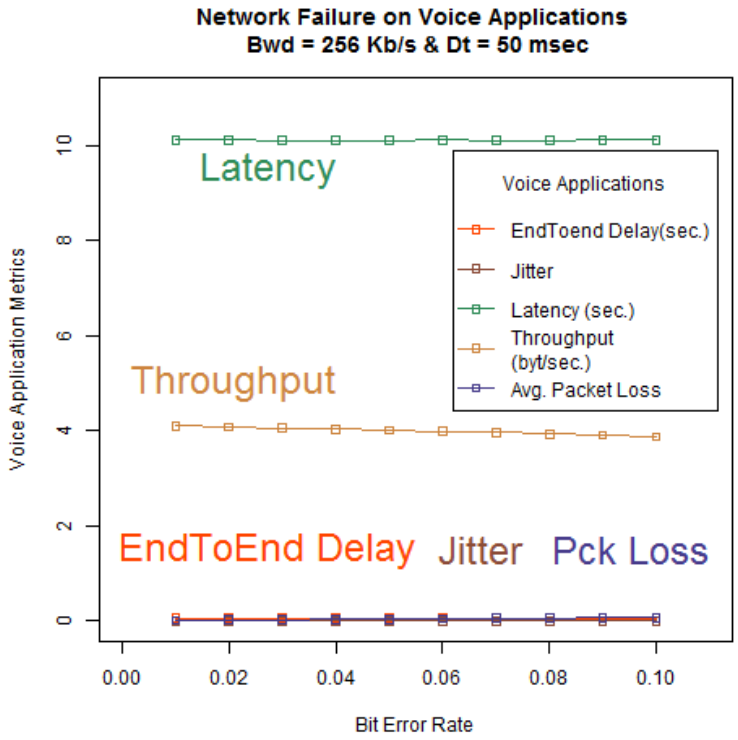
We used the values of metrics in the three cases of failures to evaluate the performance of voice application which represent RTA. And this by plotting bandwidth (in KB/s), delay time (in milliseconds) or bit error rate value with response time, end to end delay,jitter, packet loss, latency and throughput. So, voice application performance degradation is represented by response time, end to end delay, jitter, packet loss, latency and throughput and bandwidth, delay time or BER in Fig. 8.10a, Fig. 8.10b & Fig. 8.10c.



(a) Metrics with Bandwidth



(b) Metrics with Delay



(c) Metrics with Bit Error Rate

Figure 8.10: Voice Application Performance with Failures

8.1.5.2 Video Simulation & Test

To simulate and test video stream application, we simulate RTP/RTCP protocols in NS-2. We use the RTP and RTCP agents in addition to RTP session. We construct a topology to test video application. The topology is consisting of three nodes sender, receiver and video server which connect the sender and receiver with a link. The sender and receiver nodes have the Trace Traffic application above RTP agent on video sender and RTCP agent on video receiver. The topology of testing video application in NS-2 is in Fig. 8.11.

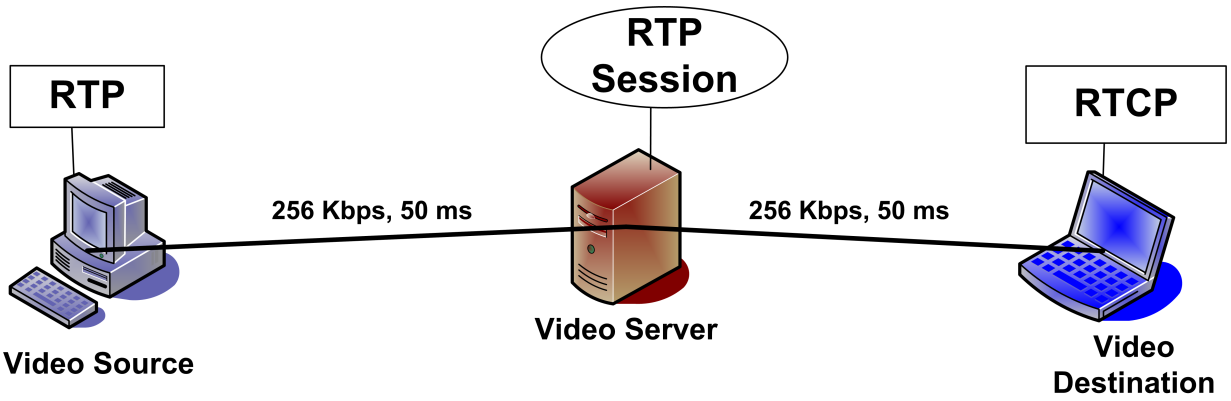
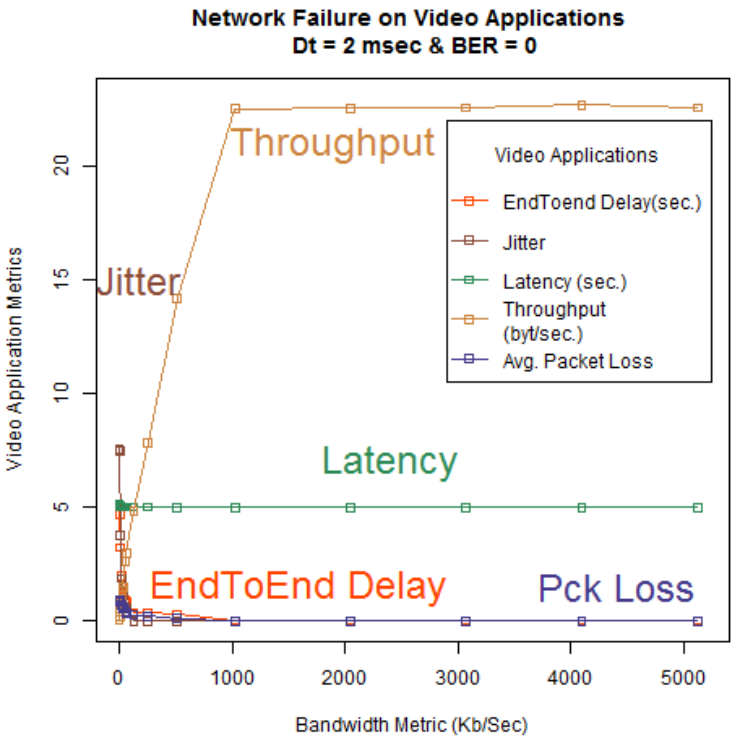


Figure 8.11: Video Application Testing Topology

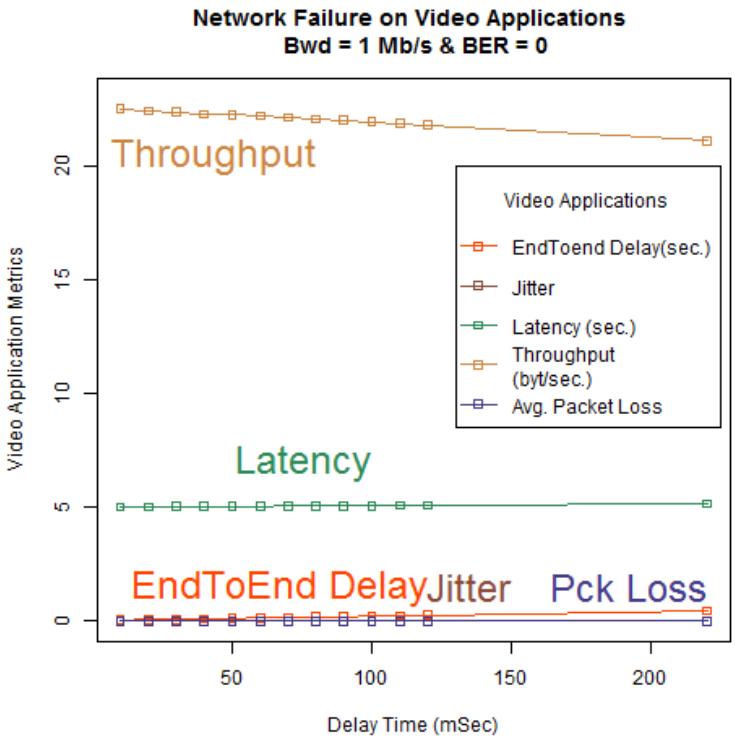
We use NS-2 network link properties to introduce failures like degradation in the bandwidth of the link, the delay time on the link and the bit error rate on the link. Then we use trace files to calculate the important metrics such as the response time, latency, throughput, end to end delay, jitter, packet loss..etc. We calculate each metric one time in each case of failures cases (e.g. bandwidth degradation, increasing of delay time or BER). In case of video application, we just calculate the response time, end to end delay, jitter, packet loss, latency and throughput in the three cases of failures, the bandwidth degradation, the error bit rate and the delay time increasing as a network failures. We use Perl scripting language to analysis trace files and calculate response time, latency, end to end delay, jitter, packet loss and network throughput.

We use the values of metrics in the three cases of failures to evaluate the performance of the video streaming application which represent RTA. And this by plotting bandwidth (in KB/s), delay time (in milliseconds) or BER value with response time, end to end delay, jitter, packet loss, latency and throughput. So, video application performance degradation is represented by response time, end to end delay, jitter, packet loss, latency and throughput and bandwidth, delay time or bit error rate in Fig. 8.12a, Fig. 8.12b & Fig. 8.12c.

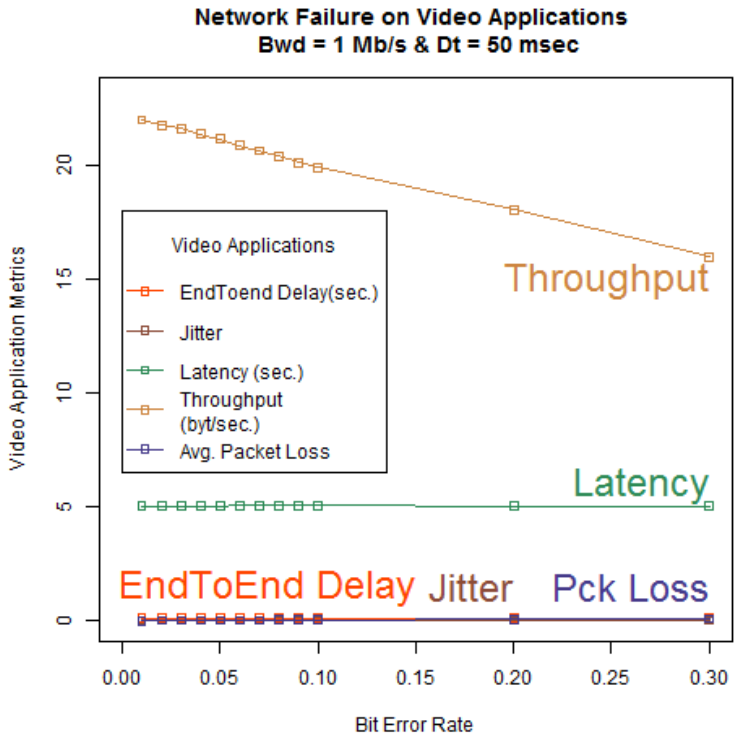
In the next couple of sections, we will discuss how we solve and mitigate the failures we introduced to application classes. We will solve all the failure by using VM migration, and specially we will solve the BER failure with FEC in addition to VM migration. The solving of failures let the performance of applications return back to acceptable state and let SLA QoS usually high, by this solving we can assure SLA services performance, QoS and response times.



(a) Metrics with Bandwidth



(b) Metrics with Delay



(c) Metrics with Bit Error Rate

Figure 8.12: Video Application Performance with Failures

8.2 Failures Mitigation with VM Migration

In this section, we discuss how we mitigate the failures we introduced to applications in the previous section. As we discussed in chapter 6, most of failures can be mitigated or solve by redundancy. One of the redundancy techniques is the device redundancy or the virtual machine migration. We use VM migration as a mitigation technique and a solution to all the failures introduced to applications. In this section we implement this VM migration in NS-2 simulator to the five classes of applications tested in the simulator in the previous section. Then after implementing VM migration in the topology of testing the applications, we show the impact of our solution on the failures and how it solve or mitigate failures. Finally, we will see the performance of applications during the mitigation of the failures. And see how VM migration increase the performance of the applications after being degraded from the failures like in the previous section. We implement the VM migration by two ways, the first is doing the migration instantly and the second is doing the migration completely. The two way of VM migration for web application are in Subsection 8.2.1. The two way of VM migration for file application are in Subsection 8.2.2. In Subsection 8.2.3, we discuss the two way of VM migration for distributed application. And in Subsection 8.2.4, we discuss the two way of VM migration for highly interactive application. Finally, the two way of VM migration for real time application are in Subsection 8.2.5.

8.2.1 VM Migration in WA

To implement virtual machine migration in web application topology, we add a new node as a web client and link this node with the HTTP cash node only. Once the failures values are increased or the performance of web application degraded, the first client node stop working and move the process of sending and receiving of web pages instantly to the new client node which represent the new VM, and this process of moving the processes instantly from old client to new node is the migration of VM.

Complete Migration for WA

In the complete migration of VM, we used the same previous topology in addition to add just a network link between the old HTTP client node and the new one. This link will be used to move completely not instantly the connection (sending or receiving web pages) from the old client node to the new one when failure happen or the performance of web application is degraded. The new topology to mitigate failures by VM migration (instantly and completely) in Fig. 8.13.

After implementing complete VM migration, we check again on the metrics of web application and the performance of the web application. We introduce the failures again to see how VM migration mitigates the failures. Then we plot the VM migration with the metrics of the application and failure values to see the impact of VM migration on performance degradation of web application which represented by the metrics (e.g. response time) and network failures (e.g. bandwidth degradation and delay time increasing), the plots are in Fig. 8.14.

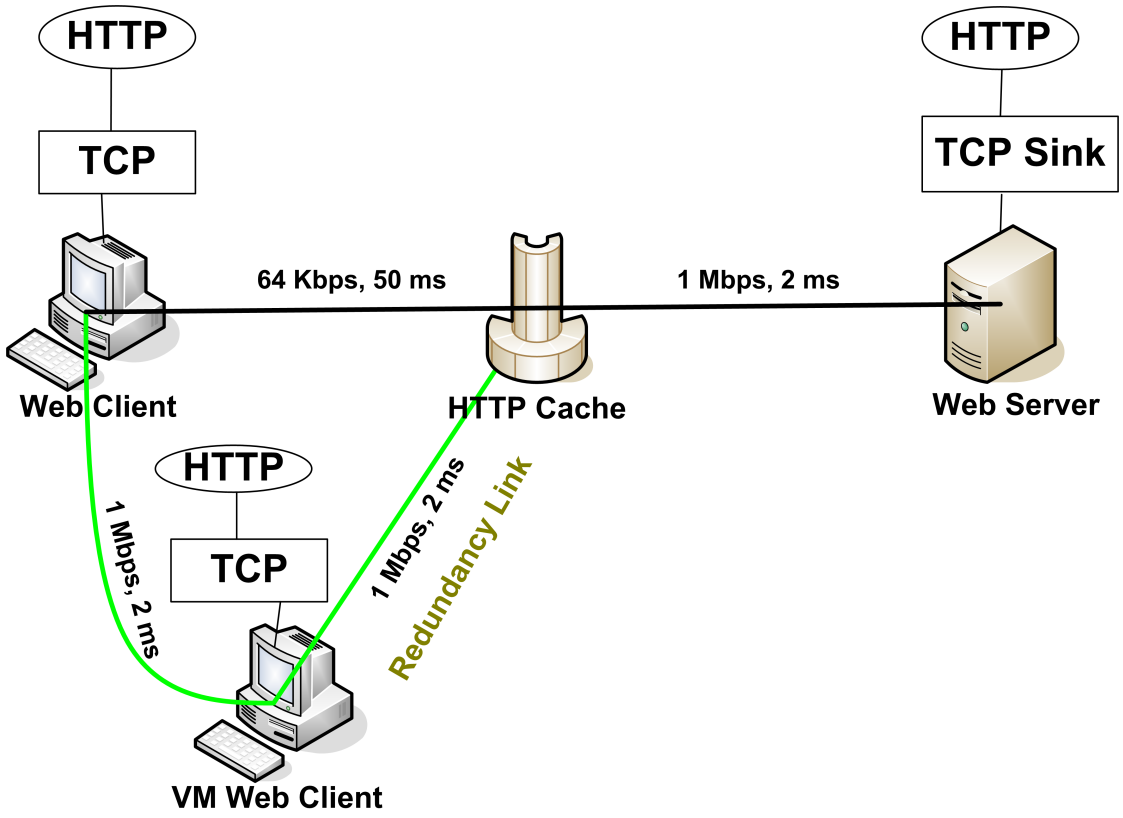
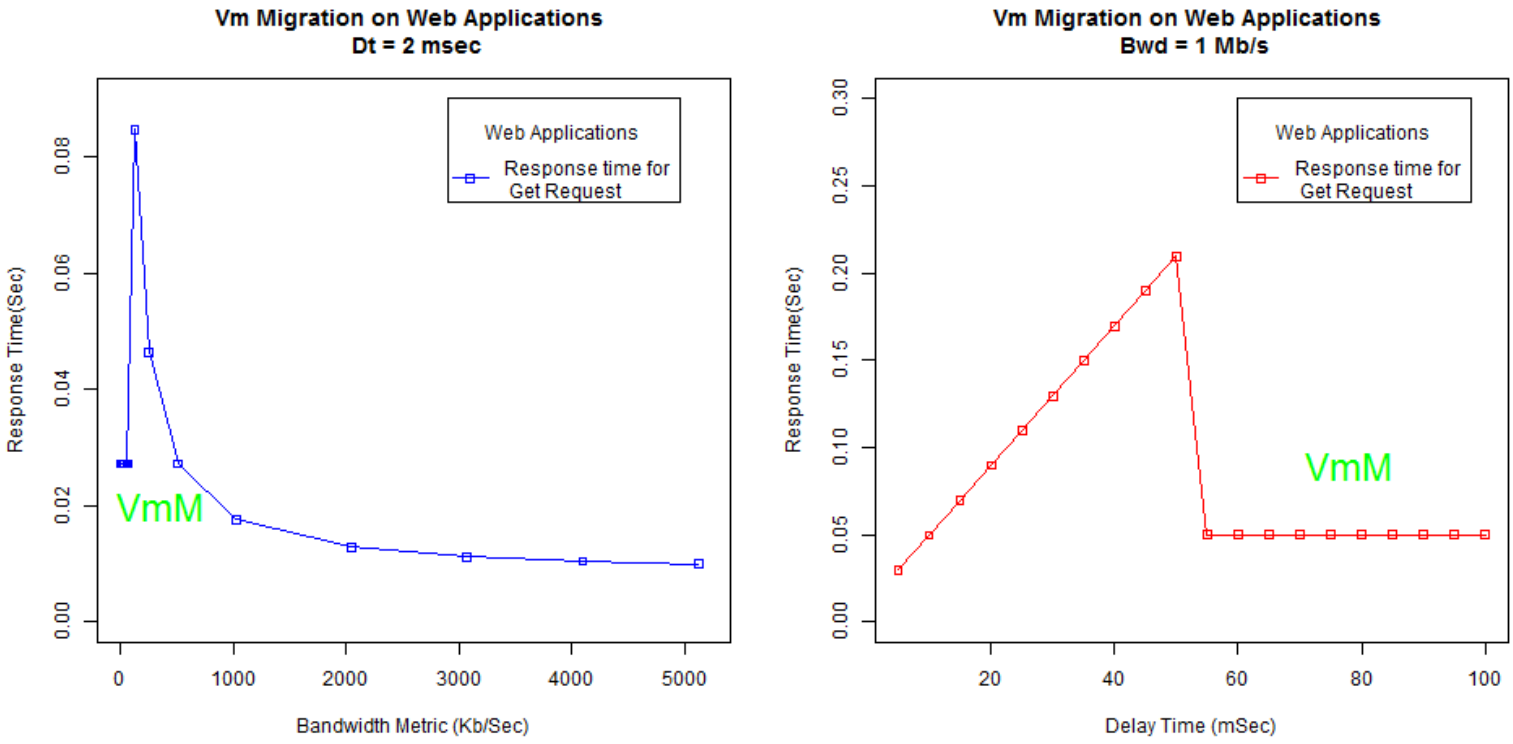


Figure 8.13: VM Migration in Web Application Topology



(a) RT with Bandwidth

(b) RT with Delay

Figure 8.14: Web Application Performance with Mitigating Failures

The values of metrics for the instantly and complete VM migration are the same, just only one difference in the value of the total time of simulation , where it increased in the complete VM migration and this because of the time taken to move form the old HTTP client node to the new one.

Mitigating failures reduce the values of metrics and increase web application performance and this help us to assure SLA QoS and keep services with good performance as mentioned in SLA contract between cloud provider and customer.

8.2.2 VM Migration in FA

To implement virtual machine migration in file application topology, we add a new node as a file client and link this node with file server node only. Once the failures values are increased or the performance of file application degraded, the first file client node stop working and move the process of uploading and downloading of files instantly to the new file client node which represent the new VM, and this process of moving the processes instantly from old file client to new node is the migration of VM.

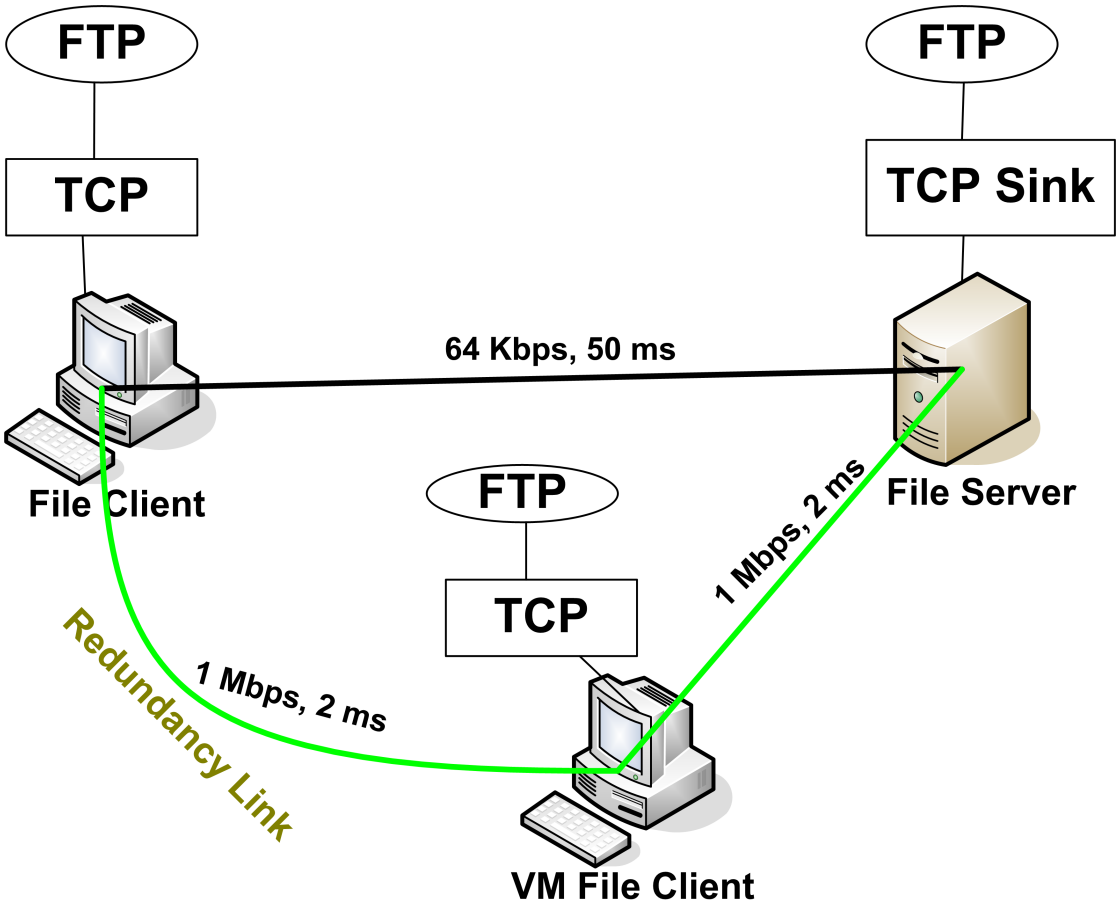
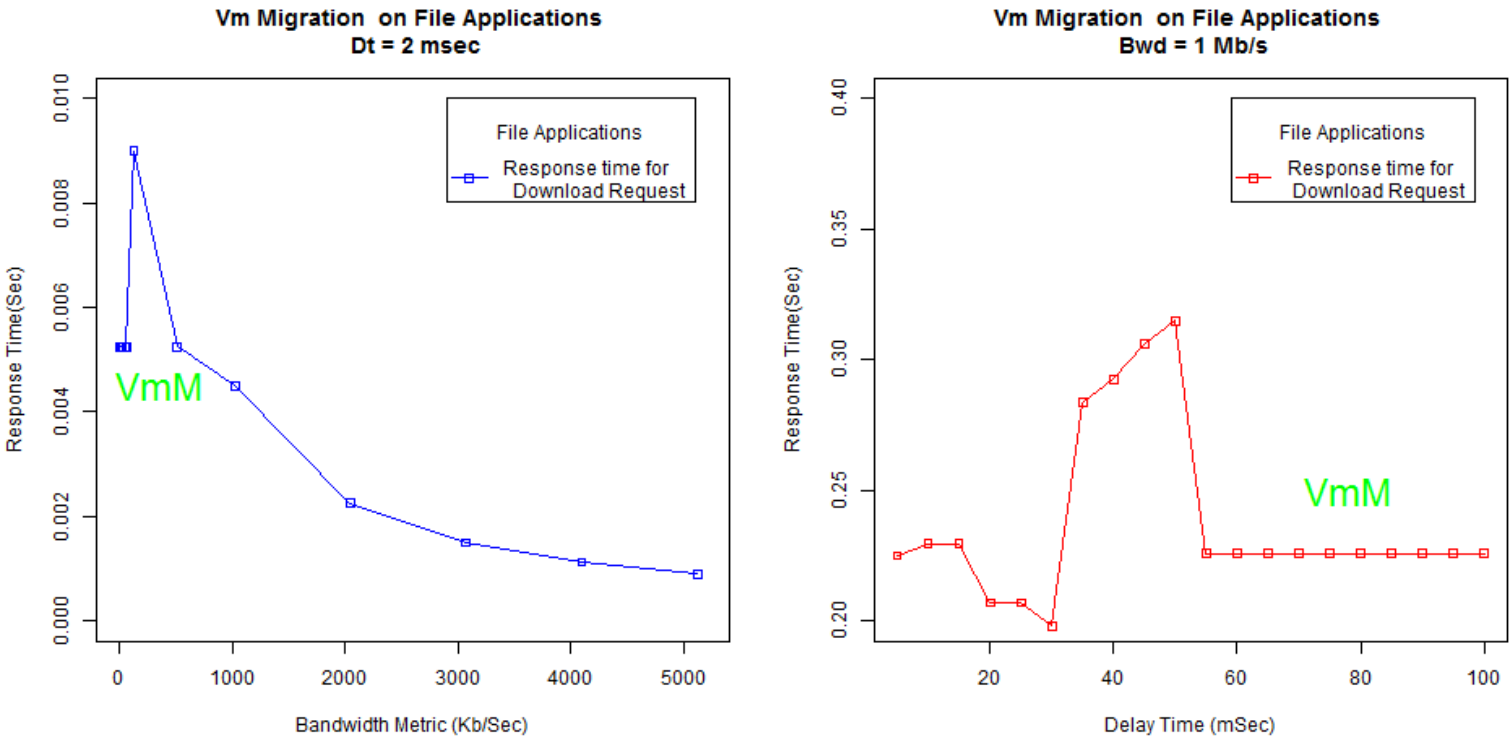


Figure 8.15: VM Migration in File Application Topology

Complete Migration for FA

In the complete migration of VM, we use the same previous topology in addition to add just a network link between the old file client node and the new one. This link will be used to move completely not instantly the connection (uploading or downloading of files) from the old file client node to the new one when failures occur or the performance of file application is degraded. The new topology to mitigate failures by VM migration (instantly and completely) in Fig. 8.15.

After implementing the complete VM migration, we check again on the metrics of file application and the performance of the file application. We introduce failures again to see how VM migration mitigates the failures. Then we plot the VM migration with the metrics of the application and the failure values to see the impact of VM migration on performance degradation of file application which represented by the metrics (e.g. response time, latency, throughput) and network failures (e.g. bandwidth degradation, delay time increasing and BER), the plots are in Fig. 8.16.



(a) RT with Bandwidth

(b) RT with Delay

Figure 8.16: File Application Performance with Mitigating Failures

The values of metrics for the instantly and complete VM migration are the same, just only one difference in the value of the total time of simulation , where it increased in the complete VM migration and this because of the time taken to move form old file client node to the new one.

Mitigating failures reduce the values of metrics and increase file application performance and this help us to assure SLA QoS and keep services with good performance as

mentioned in SLA contract between cloud provider and customer.

8.2.3 VM Migration in DA

To implement virtual machine migration in distributed application topology, we add a new node as an email client (mail user agent) and link this node with the email server node only. Once the failures values are increased or the performance of distributed application degraded, the first email client node stop working and move the process of sending and receiving of emails instantly to the new email client node which represent the new VM, and this process of moving the processes instantly from old email client to new node is the migration of VM.

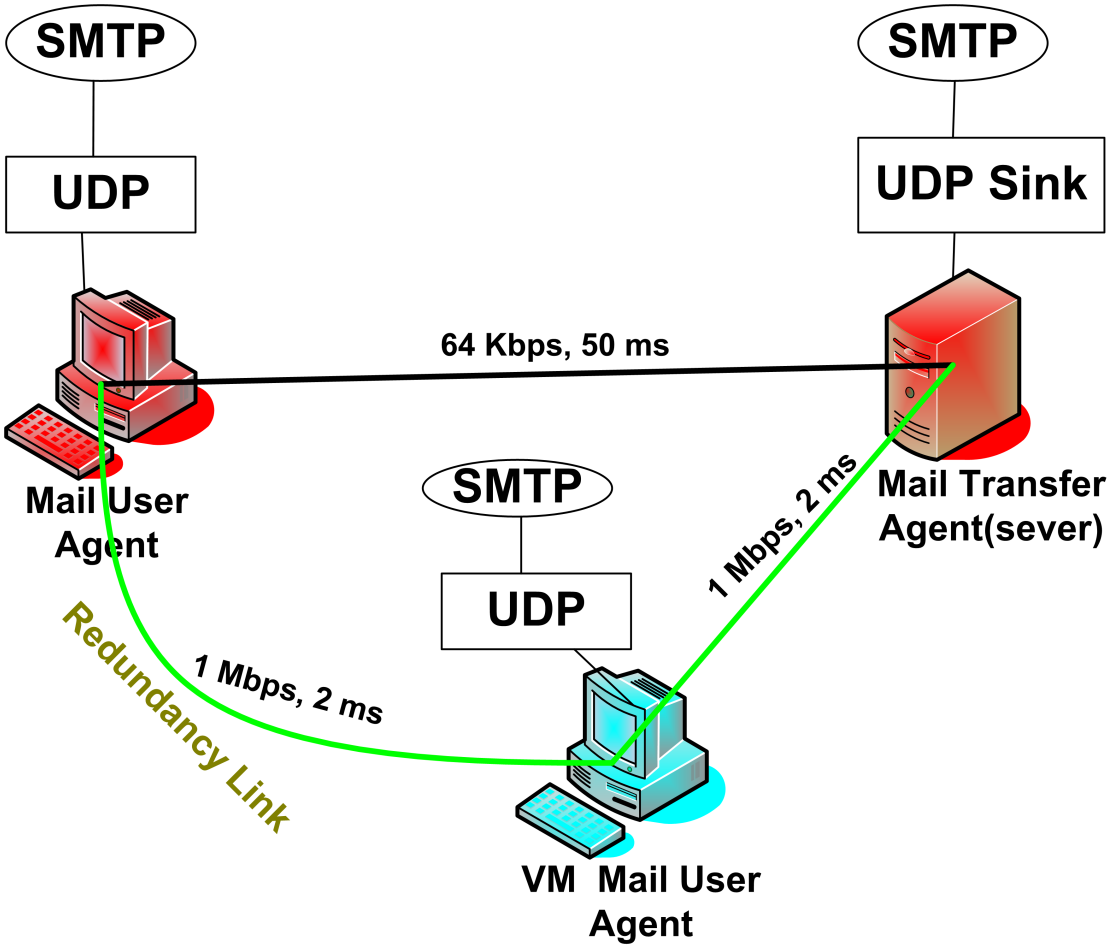


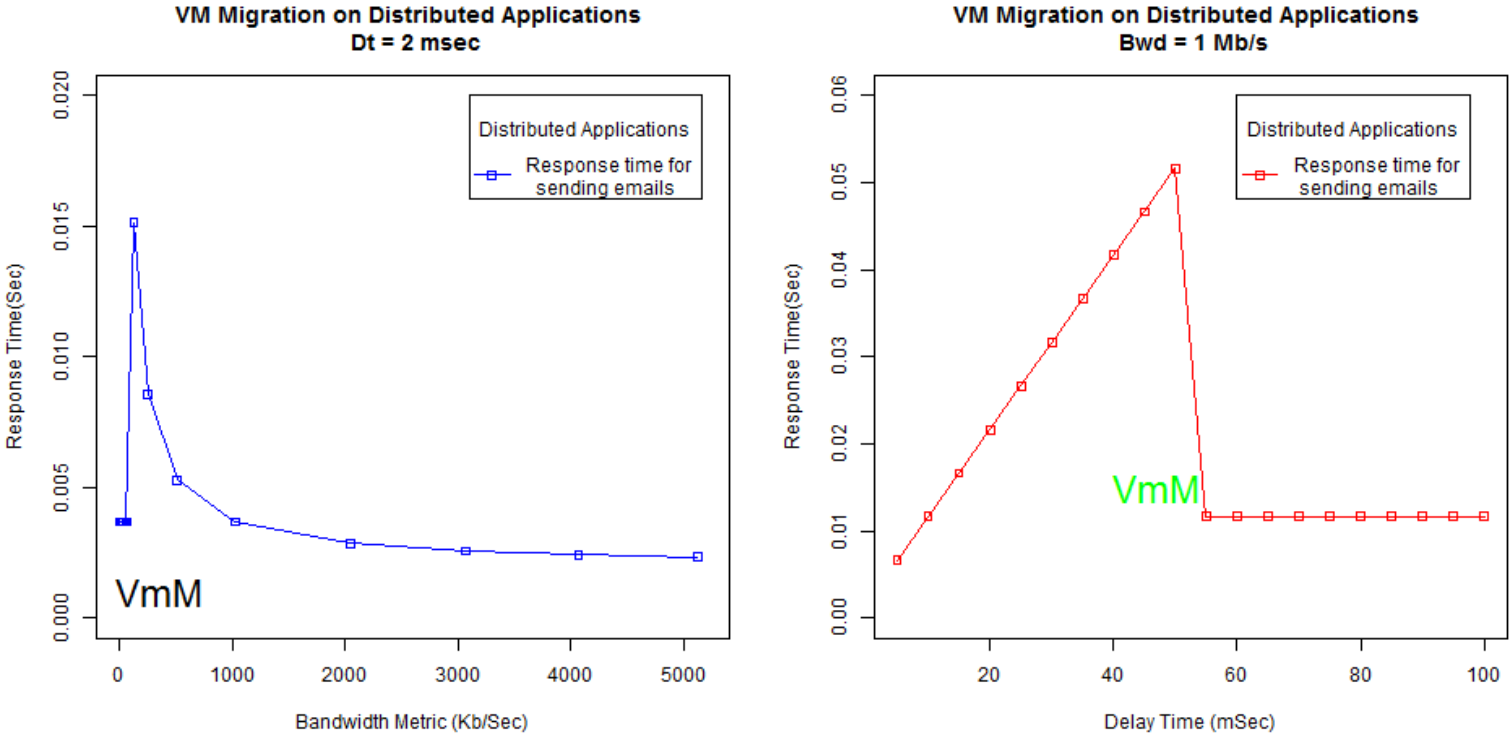
Figure 8.17: VM Migration in Distributed Application Topology

Complete Migration for DA

In the complete migration of VM, we use the same previous topology in addition to add just a network link between the old email client node and the new one. This link will be used to move completely not instantly the connection (sending or receiving of emails) from the old email client node to the new one when the failure occurs or the performance of distributed application is degraded. The new topology to mitigate failures by VM

migration (instantly and completely) in Fig. 8.17.

After implementing the complete VM migration, we check again on metrics of distributed application and the performance of distributed application. We introduce failures again to see how the VM migration mitigates the failures. Then we plot the VM migration with metrics of the application and the failure values to see the impact of the VM migration on performance degradation of distributed application which represented by the metrics (e.g. response time (message delivery time), latency, throughput) and network failures (e.g. bandwidth degradation, delay time increasing and BER), the plots are in Fig. 8.18.



(a) RT with Bandwidth (b) RT with Delay

Figure 8.18: Distributed Application Performance with Mitigating Failures

The values of metrics for the instantly and complete VM migration are the same, just only one difference in the value of total time of simulation , where it increased in the complete VM migration and this because of the time taken to move form old email client node to the new one.

Mitigating failures reduce the values of metrics and increase distributed application performance and this help us to assure SLA QoS and keep services with good performance as mentioned in SLA contract between cloud provider and customer.

8.2.4 VM Migration in HIA

To implement virtual machine migration in highly interactive application topology, we add a new node as a HIA client and link this node with the second HIA node only.

Once the failures values are increased or the performance of highly interactive application degraded, the first HIA client node stop working and move the process of sending and receiving instantly to the new HIA client node which represent the new VM, and this process of moving the processes instantly from old HIA client to new node is the migration of VM.

Complete Migration for HIA

In the complete migration of VM, we use the same previous topology in addition to add just a network link between the old HIA client node and the new one. This link will be used to move completely not instantly the connection (sending or receiving) from the old HIA client node to the new one when the failure occurs or the performance of highly interactive application is degraded. The new topology to mitigate failures by VM migration (instantly and completely) in Fig. 8.19.

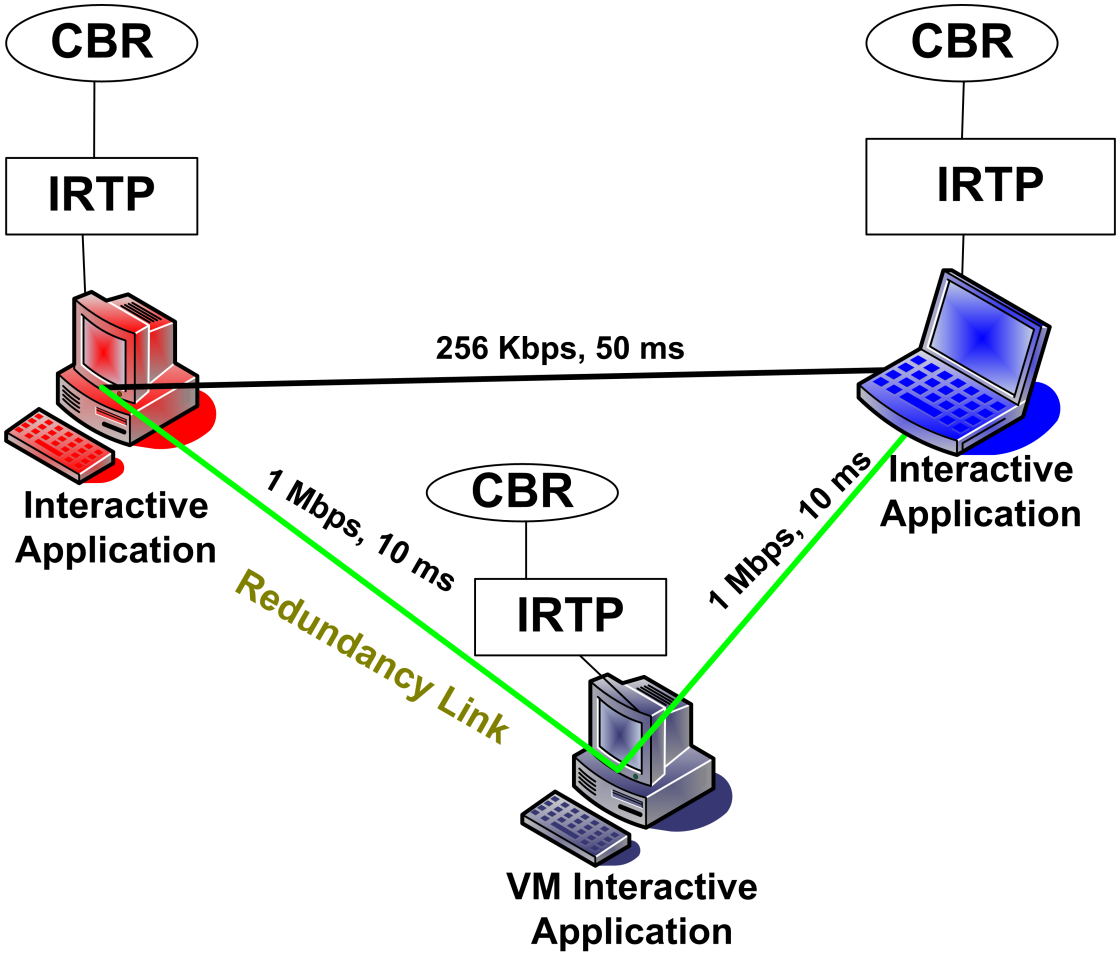


Figure 8.19: VM Migration in Highly Interactive Application Topology

After implementing the complete VM migration, we check again on metrics of highly interactive application and the performance of highly interactive application. We introduce failures again to see how VM migration mitigates failures. Then we plot VM migration with metrics of application and the failure values to see the impact of VM

migration on the performance degradation of highly interactive application which represented by the metrics (e.g. response time, latency, throughput, end-to-end delay, jitter and packet loss) and network failures (e.g. bandwidth degradation, delay time increasing and BER), the plots are in Fig. 8.20.

The values of metrics for the instantly and complete VM migration are the same, just only one difference in the value of total time of simulation , where it increased in the complete VM migration and this because of the time taken to move form old HIA client node to the new one.

Mitigating failures reduce the values of metrics and increase highly interactive application performance and this help us to assure SLA QoS and keep services with good performance as mentioned in SLA contract between cloud provider and customer.

8.2.5 VM Migration in RTA

To implement virtual machine migration in real time application topology. In voice application, we add a new node as a voice client and link this node with the second voice node only. For video application topology, we add two new nodes as a video client and video receiver then link the nodes with the video server node only. Once the failures values are increased or the performance of both real time applications (voice & video) are degraded, the first (voice or video) client node stop working and move the process of sending and receiving of audio instantly to the new (video or voice) client node which represent the new VM, and this process of moving the processes instantly from old (voice or video) client to the new node is the migration of VM.

8.2.5.1 Complete Migration for VoIP

In the complete migration of VM for voice application, we use the same previous topology in addition to add just a network link between the old voice client node and the new one. This link will be used to move completely not instantly the connection (sending or receiving the voice media) from the old voice client node to the new one when the failure occurs or the performance of voice real time application is degraded. The new topology to mitigate failures by VM migration (instantly and completely) in Fig. 8.22.

After implementing the complete VM migration, we check again on metrics of voice application and the performance of voice application. We introduce failures again to see how VM migration mitigates failures. Then we plot VM migration with metrics of the application and failures values to see the impact of VM migration on performance degradation of voice application which represented by the metrics (e.g. response time, latency, throughput, end-to-end delay, jitter, packet loss and MOS) and network failures (e.g. bandwidth degradation, delay time increasing and BER), the plots are in Fig. 8.21.

The values of metrics for the instantly and complete VM migration are the same, just only one difference in the value of total time of simulation , where it increased in the complete VM migration and this because of the time taken to move form old voice client node to the new one.

Mitigating failures reduce the values of metrics and increase voice application performance and this help us to assure SLA QoS and keep services with good performance as mentioned in SLA contract between cloud provider and customer.

8.2.5.2 Complete Migration for Video

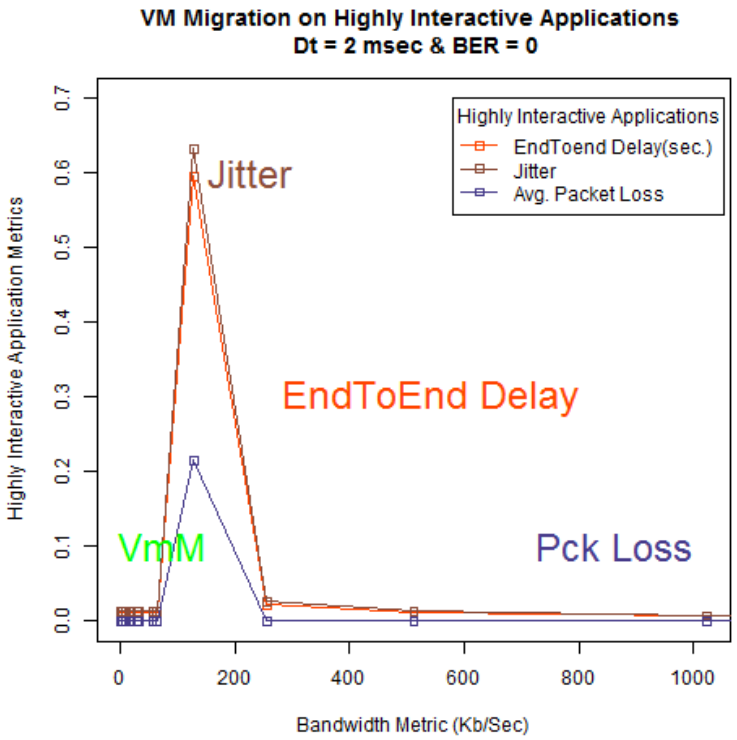
In the complete migration of VM for video streaming application, we use the same previous topology in addition to add just a network link with new node between the old video client node and the new client one. The same change has been done to the receiver nodes. The new links will be used to move completely not instantly the connection (sending or receiving the video media) from the old video client node to the new one when the failure occurs or the performance of video real time application is degraded. The new topology to mitigate failures by VM migration (instantly and completely) in Fig. 8.23.

After implementing the complete VM migration, we check again on metrics of video application and the performance of video application. We introduce failures again to see how VM migration mitigates failures. Then we plot VM migration with metrics of the application and failures values to see the impact of VM migration on performance degradation of video application which represented by the metrics (e.g. response time, latency, throughput, end-to-end delay, jitter, packet loss and MOS) and network failures (e.g. bandwidth degradation, delay time increasing and BER), the plots are in Fig. 8.24.

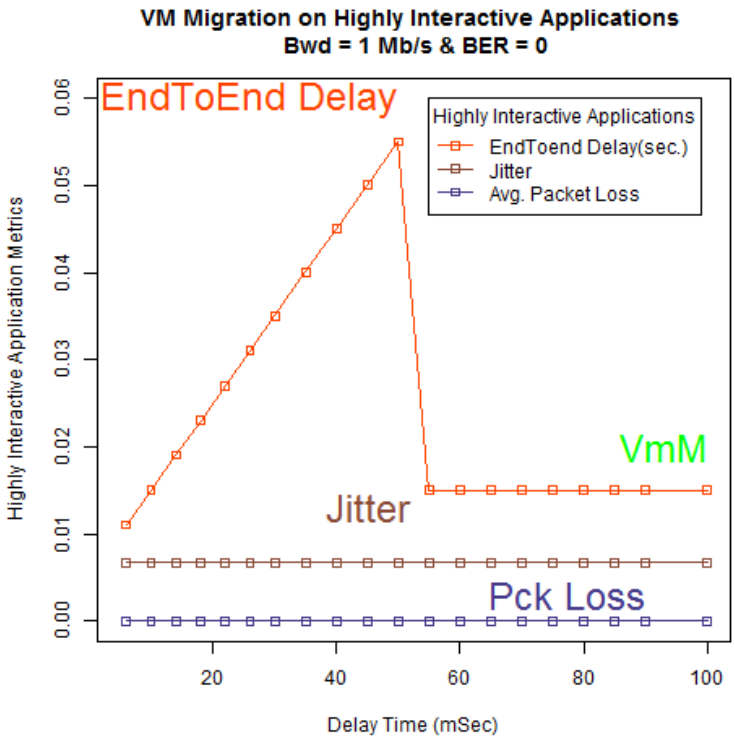
The values of metrics for the instantly and complete VM migration are the same, just only one difference in the value of total time of simulation , where it increased in the complete VM migration and this because of the time taken to move form old video client node to the new one.

Mitigating failures reduce the values of metrics and increase video application performance and this help us to assure SLA QoS and keep services with good performance as mentioned in SLA contract between cloud provider and customer.

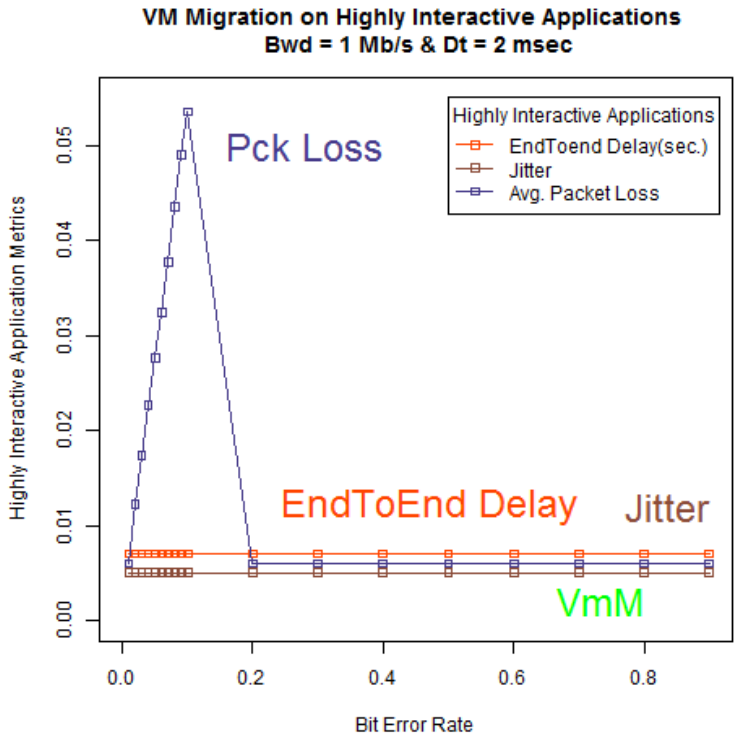
In the next section, we discuss the second mitigation technique we have implemented. In addition to VM migration mitigation technique, we will use FEC as a mitigation technique for the BER failure.



(a) Metrics with Bandwidth

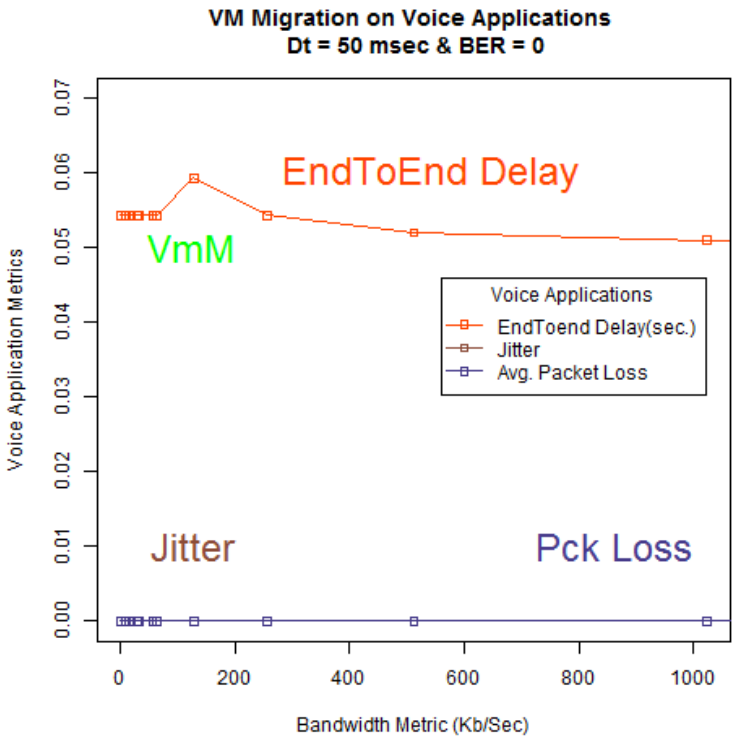


(b) Metrics with Delay

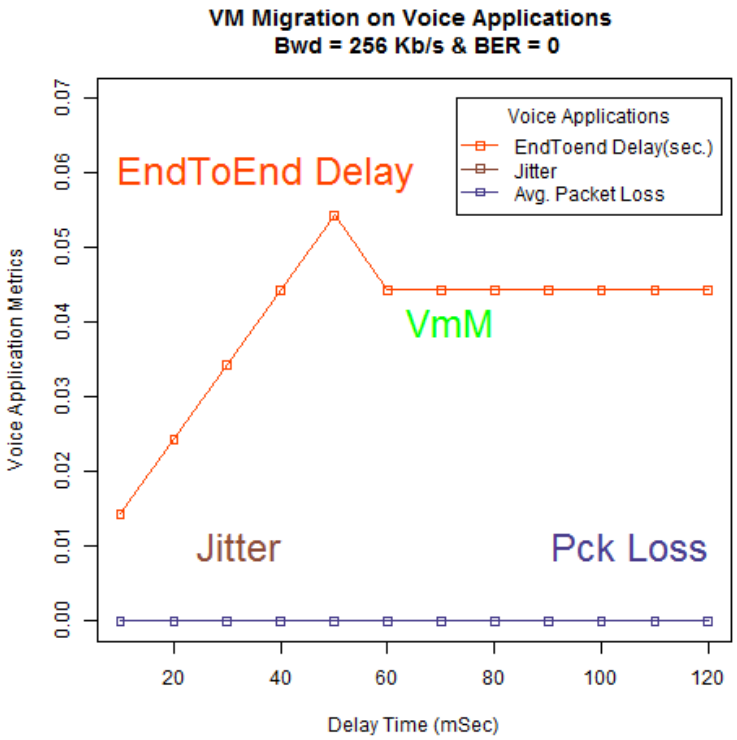


(c) Metrics with Bit Error Rate

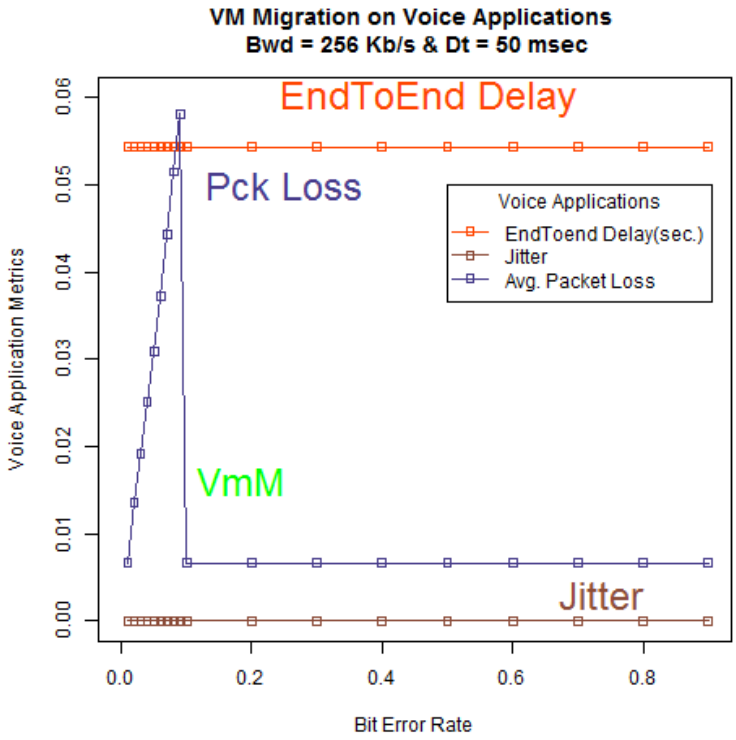
Figure 8.20: Highly Interactive Application Performance with Mitigating Failures



(a) Metrics with Bandwidth



(b) Metrics with Delay



(c) Metrics with Bit Error Rate

Figure 8.21: Voice Application Performance with Mitigating Failures

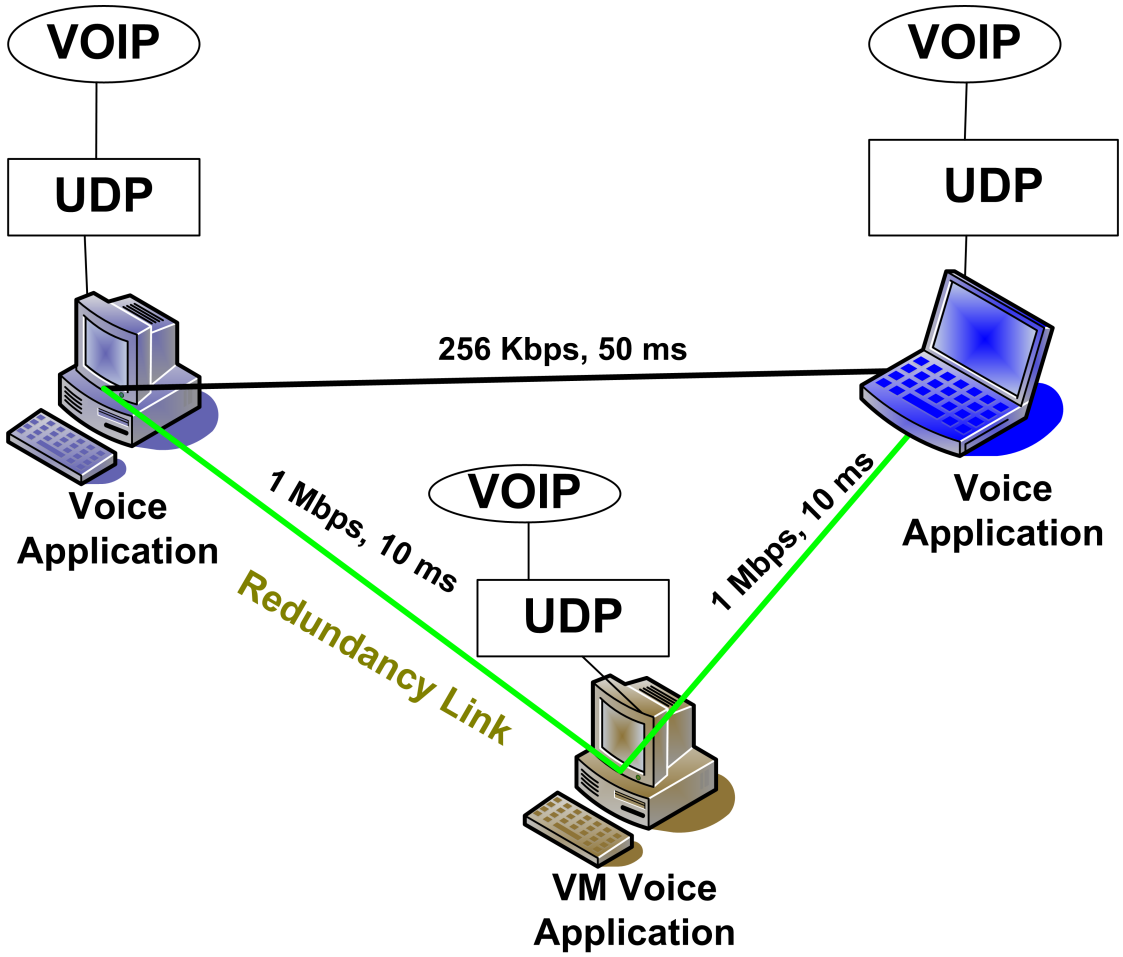


Figure 8.22: VM Migration in Voice Application Topology

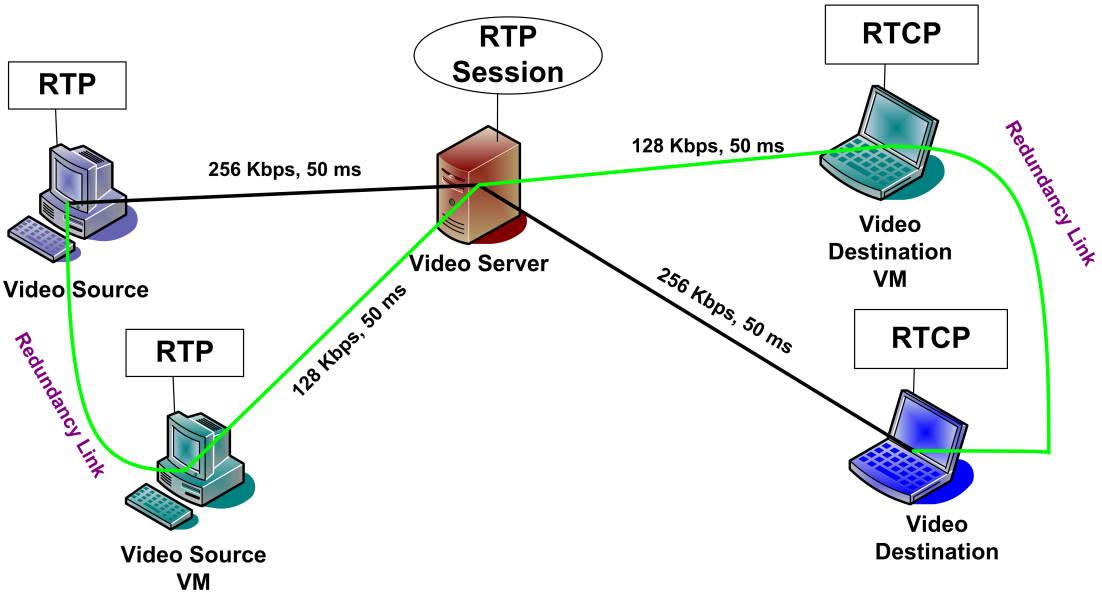
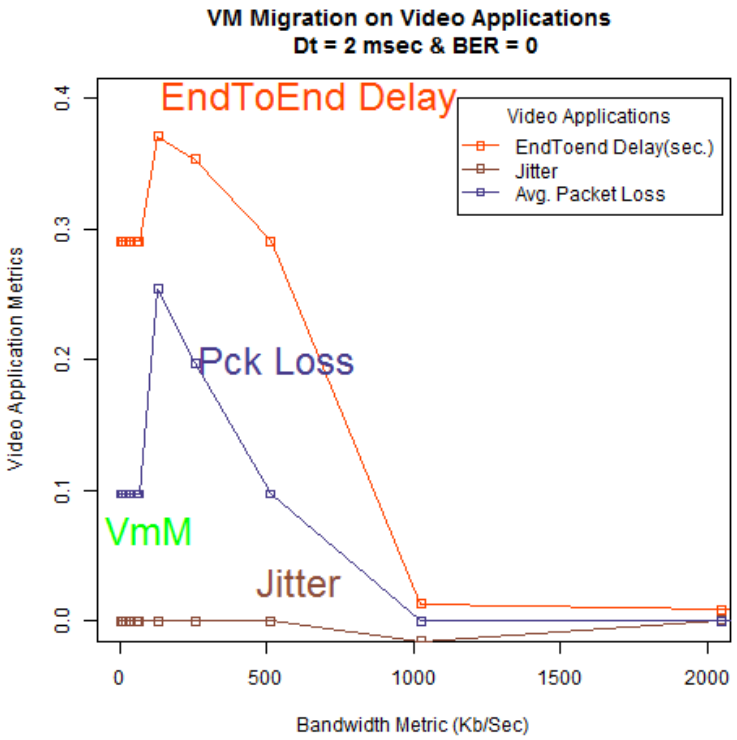
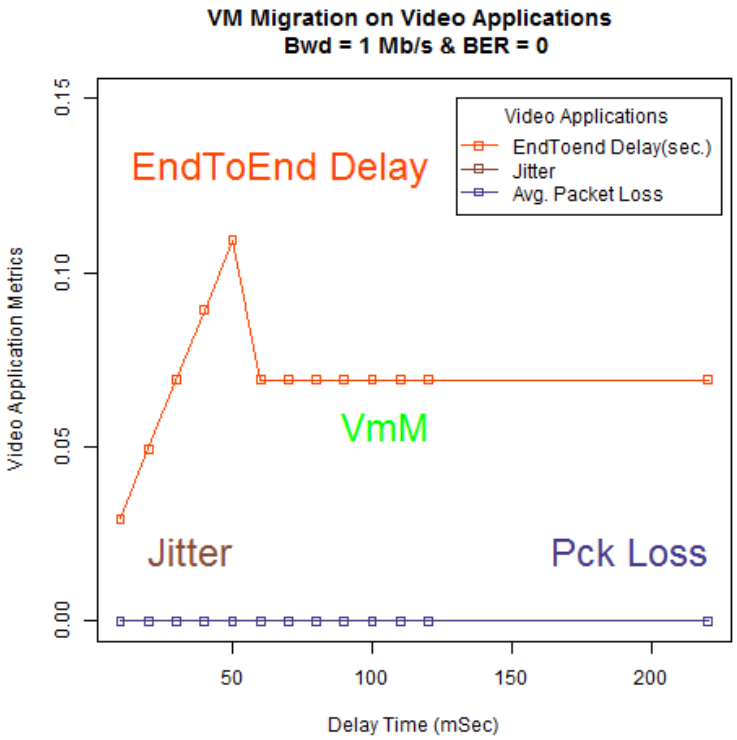


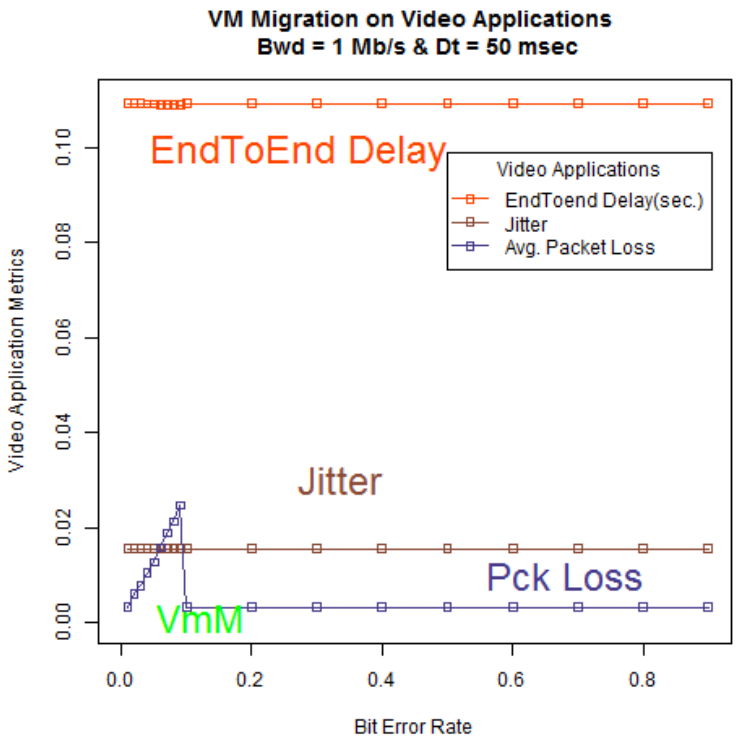
Figure 8.23: VM Migration in Video Application Topology



(a) Metrics with Bandwidth



(b) Metrics with Delay



(c) Metrics with Bit Error Rate

Figure 8.24: Video Application Performance with Mitigating Failures

8.3 Mitigate BER with FEC

In this section, we show the second mitigation technique for failures, we use this technique to mitigate the BER failure in the link. The technique is the forward error correction. We use FEC to mitigate BER on the same link without any virtual machine migration. Firstly, we explain what is FEC in Subsection 8.3.1. Then we show how we implement the failure mitigation with FEC in NS-2 simulator in Subsection 8.3.2.

8.3.1 Forward Error Correction

Forward error correction is a digital signal processing used to correct errors and enhance data reliability in the communications channels. FEC can correct and detect a limited number of errors without any re-transmitting of data stream. There are two types of FEC codes used to detect and correct errors such as Convolution codes and Block codes (e.g. BCH code).

Forward error correction does the error correction and detection by introducing redundant data, called error correcting code and this through the sender. The sender adds redundant data to its messages, which allows the receiver to detect and correct errors (within some bound) without the need to ask the sender for additional data and without a reverse channel to request the re-transmission of data. And this is the advantage of FEC, where the re-transmission of data can often be avoided, at the cost of higher bandwidth requirements on average, and is therefore applied in situations where re-transmissions are relatively costly or impossible.

The first FEC code developed, was called a Hamming code, was introduced in the early 1950s. In the Hamming codes, the errors are obtained in the data transmission where the transmitter sends redundant data. There is only small portion of the data without apparent errors can be recognized by the receiver. And this allows the single source to broadcast data to multiple sources.

8.3.2 Mitigate BER Failure

Bit error rate failure, it introduces to highly interactive application and the real time application (voice and video). It doesn't introduced to web, file and distributed applications because it have no impact on the important metrics (e.g. response time, delay time) for those applications. For HIA and RTA it affect the number of packet loss metric of the applications. In the previous section, we mitigate this failure with VM migration, and here we solve it without migration and by using Forward error correction on the link that have the failure to detect and correct the errors.

FEC module in Ns-2 is used in the receiving node, The cyclic redundancy check is performed for error detection, if bit errors should be ignored, or if the packet should be dropped whenever it contains bit errors.

To implement Forward error correction in NS-2 simulator for HIA and RTA, we use the first basic topology for the applications like in HIA, we use the two nodes sender and receiver with out any redundant nodes (VM). In voice also, we used the two nodes voice sender and voice receiver with out any redundant nodes (VM). In video, we used the two nodes sender and receiver in addition to the video server with out any redundant nodes (VM). Then we use the FEC model implemented in the NS-2 simulator and the variable

FEC strength in the error model and assign it to the value 2. Also we can use the new implemented FEC correction module[1] from university of Columbia.

After the implementation of FEC, we measure the packet loss metric again for the HIA and RTA, we found that the number of packets lost is decreased because the packets is corrected by FEC. But still there are a dropped packets which contains a bit errors. We saw that the mitigation with VM migration is more better where we change totally the link that have the failure, but this under the condition that there is an easy way to migrate.

Fig. 8.25 shows how the number of packets lost on the link that have the failure is decreased after using FEC as a mitigation technique for BER failure in HIA and RTA (e.g. voice and video).

In next chapter, we will discuss our experimental results, we have gotten from our work. And see how the performance of applications and services return back to good state after solving failures. And how we assure SLA QoS and response times metric mentioned in SLA document.

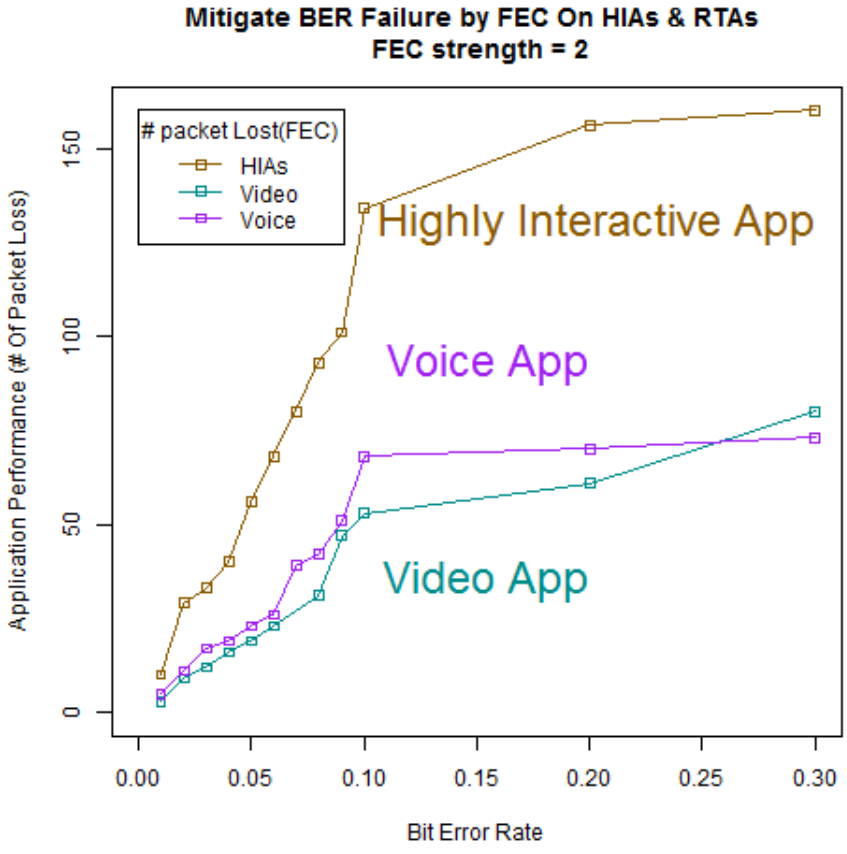


Figure 8.25: Mitigating BER with FEC in RTA & HIA

[1] http://ightwave.ee.columbia.edu/

Chapter IX

Experimental Results

"History admits no rules, only outcomes"

David Mitchell-Cloud Atlas

In this chapter, we explain our results we got from the previous experiments we did in the two previous chapters 7& 8. Also, we will analysis our experimental results to see what exactly the effect of failures on cloud applications. Then see which class of applications can still work with failures and which can not. Finally, we discuss if there are any way for mitigating failures without VM migration in other words, see if we can mitigate the failures in the same machine or same link. And we already did this by mitigating BER by using the Forward error correction as a mitigation technique in the same link with out any redundancy nodes or links. The experimental results of real application scenarios are in Section 9.1. Then, the experimental results of simulating applications on NS-2 are in Section 9.2. The VM migration mitigation technique results are in Section 9.3. The FEC mitigation technique for BER failure is in Section 9.4. Finally, we do a comparison between the two mitigation techniques (VM migration, FEC) in the last Section 9.5.

9.1 Real Experiments Results

In this section, we review the experimental results for the real scenarios experiments of the applications. We tested four classes of applications web, file, distributed and real time applications. The plot in Fig. 9.1 shows the response time for web and file applications and delay time for the distributed and real time applications. The plot shows the performance degradation of the classes of application. The performance of the applications is degraded by the decreasing of the bandwidth value and the increasing in the delay time as a network failure.

File, web and distributed applications can continue work properly till the value of bandwidth reach to 64 kb/s, after that the performance of those applications start degraded. This value of bandwidth is very small and it may happen when the network crumble or dead. For real time application, the performance of this application start degradation more early than the other applications do and this when the value of the bandwidth reach to 128 kb/s or less. So, real time application need high stable network and it can not working properly any more after this value of bandwidth.

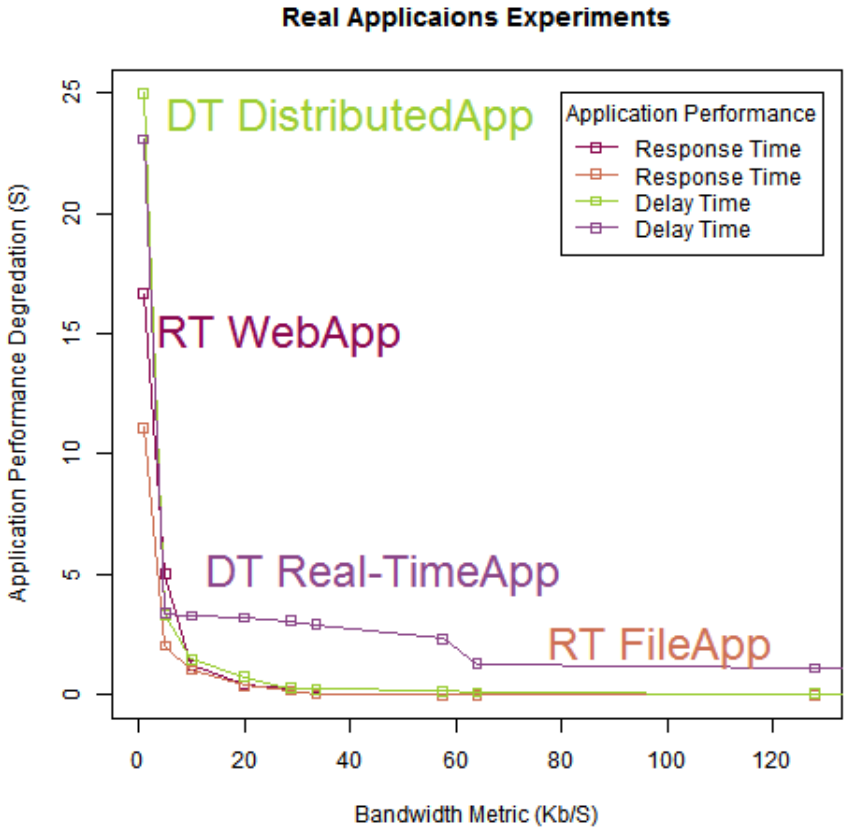


Figure 9.1: Real Applications Experimental Results

From that, we can say that file and distributed application not need for failure mitigation, and also for web application. This because it's rarely when the network bandwidth reach to 64 kb/s and even if it reach to this value, these applications can work with less performance till the network crumble totally and in this case we should solve the problem. Real time application need a mitigation for failures to continue work properly. We did the mitigation in the simulation experiments. In the future work, we will try to mitigate the failures in the real experiments.

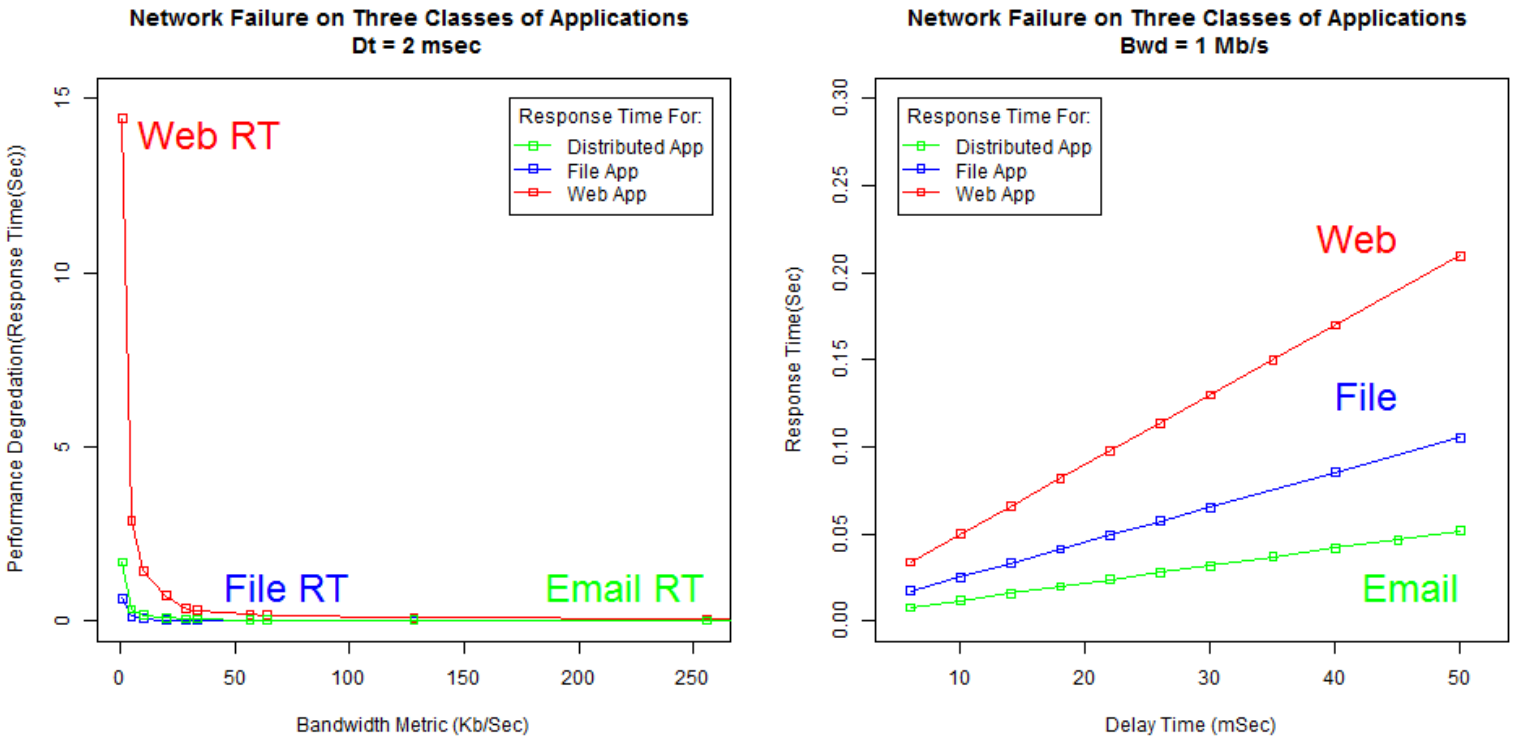
In the next sections, we show the simulation experiments results and how the application services performance become good after solving failures.

9.2 Simulator Experiments Results

In the simulator experiments, we simulated five classes of cloud applications. The web, file, distributed, highly interactive and real time (voice & video) applications. The experimental results for web, file and distributed applications are in Subsection 9.2.1. The experimental results for real time and highly interactive applications are in Subsection 9.2.2.

9.2.1 Web, File & Distributed Applications

The plots in Fig. 9.2 show the experiments of web, file and distributed applications. We can see in Fig. 9.2a, the performance of web application start degradation after 64 kb/s of bandwidth, but for file and distributed applications they start degradation when the bandwidth close to 0 kb/s. So, file and distributed applications have less effect from failures than web application. Web application needs the network failure mitigation but late. For distributed and file applications, they don't need any mitigation for failures where they can contain working even if there are failures, but they will work with less performance however we mitigate failures and will see the results of mitigation in the next section.



(a) Performance Degradation with Bandwidth

(b) Performance Degradation with Delay Time

Figure 9.2: Web, File & Disributed Applications Experimental Results

Also, Fig. 9.2b verifies the previous speech but in other kind of problem which is the increasing of delay time on the link. The response time for web is higher than 0.1 seconds when DT on link failure is became more than or equal to 10 millisecond. But in file and distributed applications the response time is not higher than 0.1 seconds even when DT on link failure is became greater than or equal 50 milliseconds. From that we verify that file and distributed applications don't need for failures mitigation and they can work with less performance in case of failure. And web application needs for failure mitigation like VM migration and just rarely.

9.2.2 HIA & RTA

The plots in Fig. 9.3 show the experiments of highly interactive application and real time application which represented by voice and video applications. Fig. 9.3a shows the

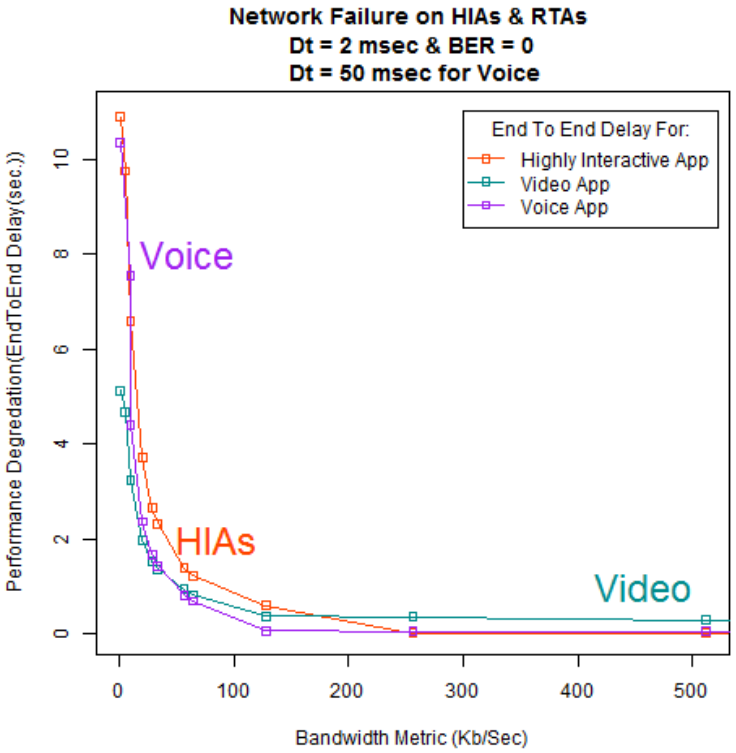
end-to-end delay for voice, video and highly interactive applications with the bandwidth of the network. The three applications have a high effect from the failures when the bandwidth reach to 128 kb/s. The performance of the highly interactive application have high impact from the bandwidth decreasing failure. HIA is effected more than voice and video applications where it need a high bandwidth. For video and voice applications, their performance is degraded but not that much like HIA and this because of the QoS on the network for the conversational applications (e.g. voice, video). So, voice and video applications need stable network not high bandwidth like HIA.

In the second Fig. 9.3b, the delay time increasing on the link failure is introduced for the applications. Video application has the most performance degradation effect because it is degraded when DT reaches to 10 milliseconds or more. For voice and HIA, they have less effect in the performance from that failure because it is degraded when DT reaches to 50 milliseconds or more . Finally, as seen in Fig. 9.3c, the number of packet loss for voice application and HIA are more than the number of packet loss for video application because of the bit error rate failure. This because video application starts degradation when BER equal 0.1, HIA starts degradation when BER equal 0.03 and voice application starts degradation when BER equal 0.01.

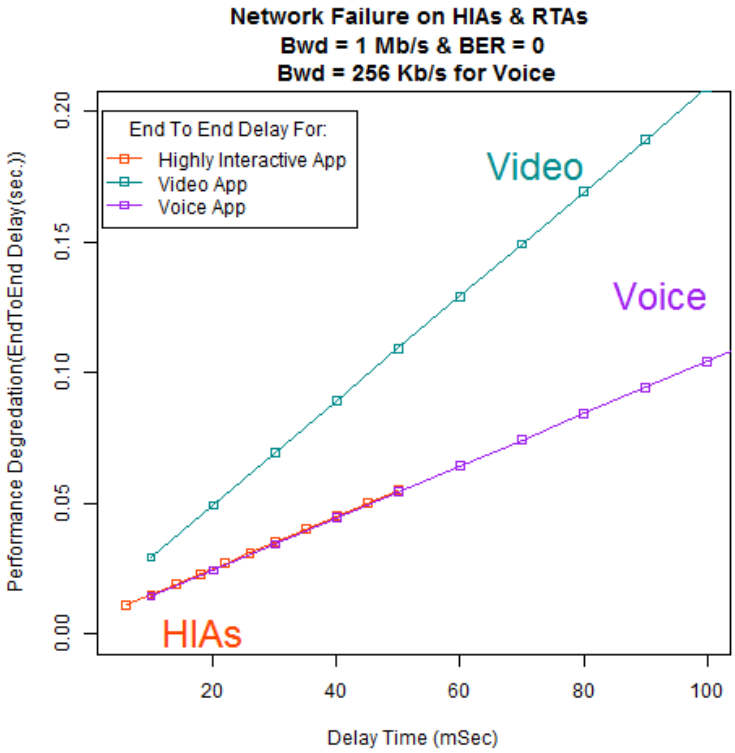
Generally, HIA and RTA need to failures mitigation as we did in the previous chapter. For the bandwidth decreasing and delay time increasing failures we used VM migration to mitigate them even for bit error rate we used the same technique. But, we also can mitigate the bit error rate failure without migration by using FEC on the network channel. Voice and HIA can work properly in the delay time increasing failures under the condition that delay time value not exceed 50 milliseconds value. But in video, it can not work properly with high delay time because when delay time in the network increase the end-to-end delay time is duplicated and this not good for video applications.

Fig. 9.4 contains a comparison between the performance degradation of applications in the real experiments and in simulator experiments.

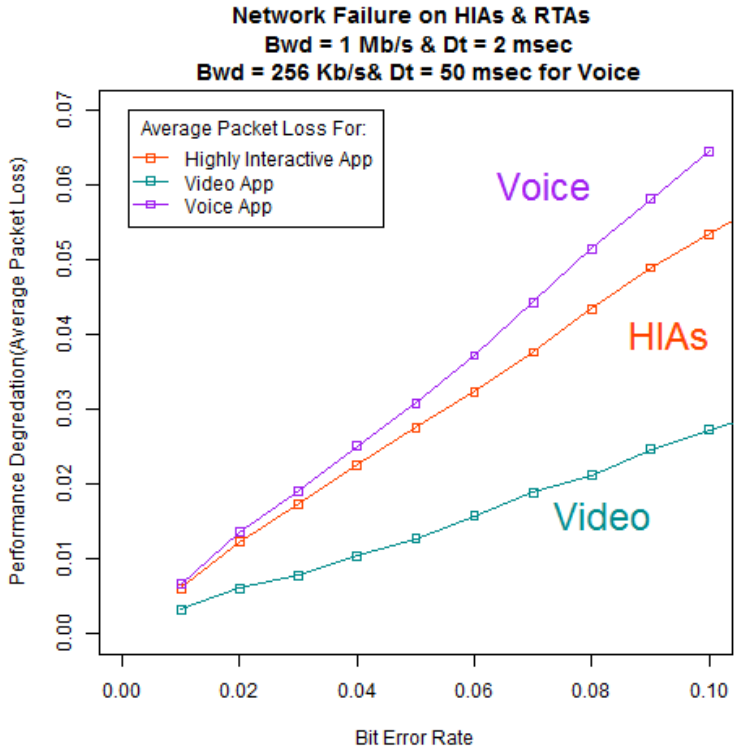
In the next section, we will show the results of failures mitigation and show the impact of solving failures on applications performance. We need to solve failures, because this will help us to assure a good SLA QoS and stable services performance even the response times mentioned in SLA.



(a) Performance Degradation with Bandwidth



(b) Performance Degradation with Delay



(c) Performance Degradation with Bit Error Rate

Figure 9.3: HIA & RTA Experimental Results

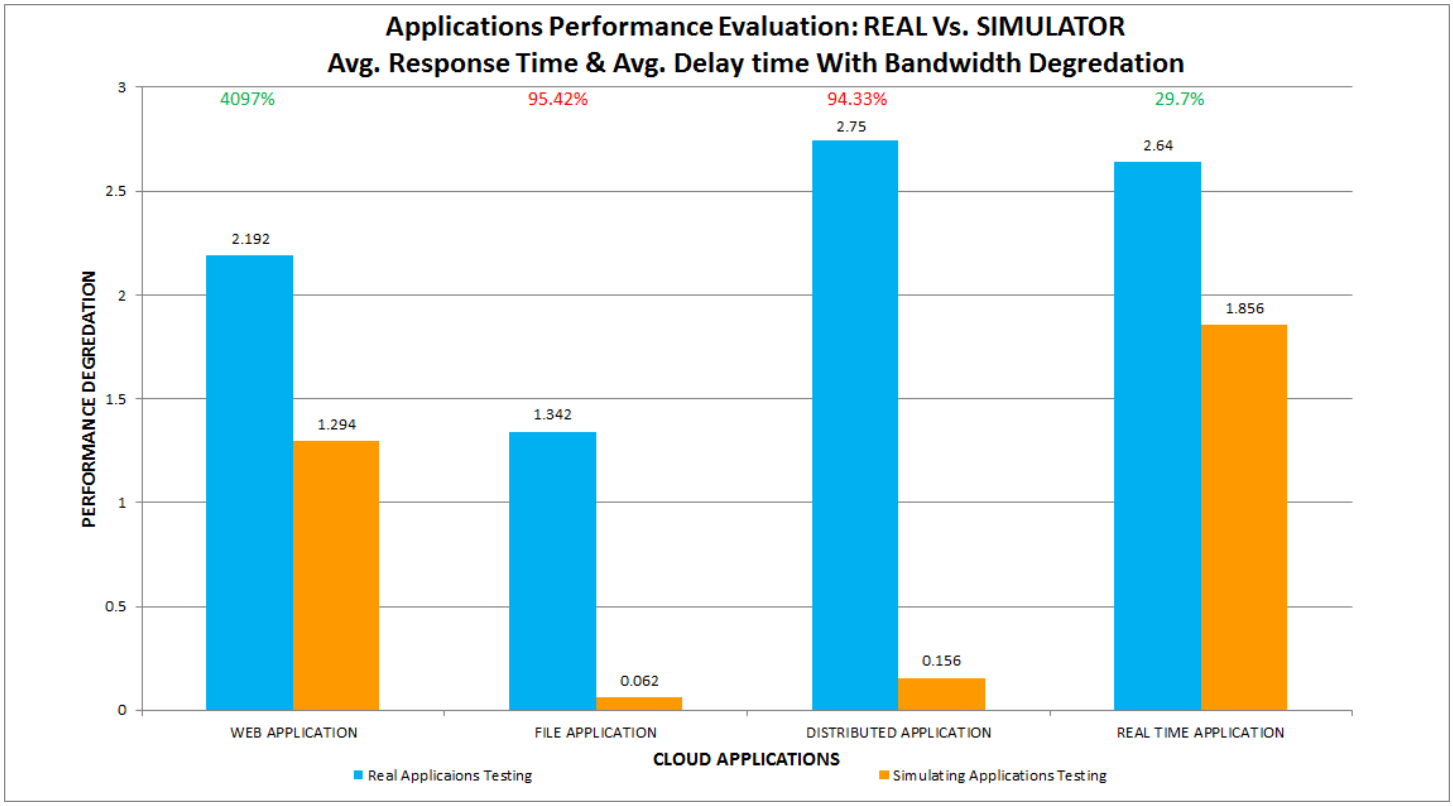


Figure 9.4: Real Vs. Simulation Experimental Results

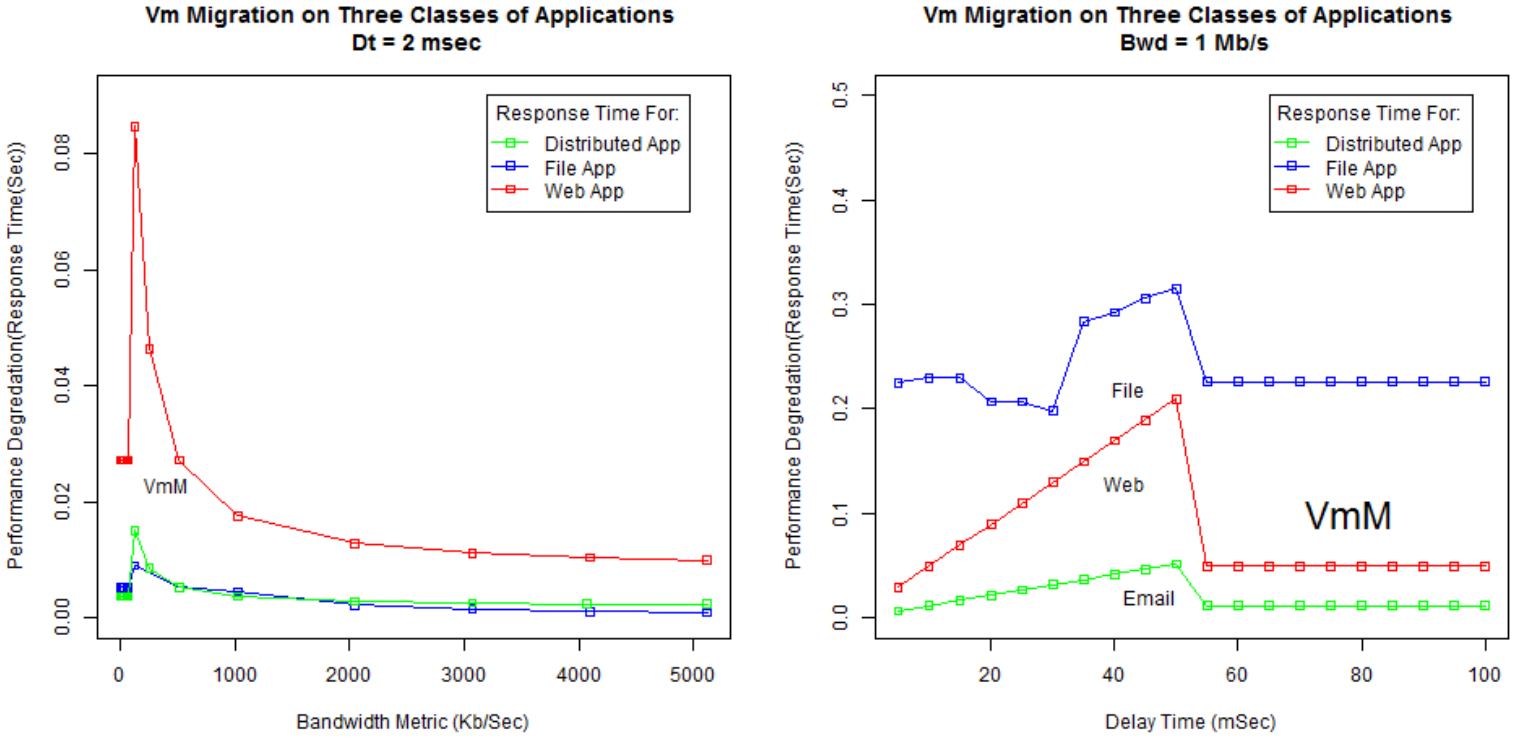
9.3 VM Migration Mitigation Technique Results

In this section, we show the results of the mitigation technique (VM migration) we implemented in the simulation experiments for five classes of the application. The experimental results for VM migration in web, file and distributed applications are in Subsection 9.3.1. The experimental results for VM migration in highly interactive application and real time application are in Subsection 9.3.2.

9.3.1 Web, File & Distributed Applications

The plots in Fig. 9.5 show the experiments of web, file and distributed applications. Fig. 9.5a shows the impact of the virtual machine migration on the performance of web, file and distributed applications during the bandwidth decreasing failure. The response time for the applications after increasing because of the failure, once the value of the bandwidth reach to 64 kb/s the mitigation technique start working and migrate the processing to the virtual machine. Then the response time return to decrease again and the performance of the applications increased.

Also, Fig. 9.5b shows the impact of the virtual machine migration on the performance of web, file and distributed applications during the delay time increasing failure. The response time for the applications after increasing because of the failure, once the value of the delay time reach to 50 millisecond the mitigation technique start working and migrate the processing to the virtual machine. Then the response time return to decrease



(a) Bandwidth Decreasing Failure Mitigation

(b) Delay Time Increasing Failure Mitigation

Figure 9.5: Web, File & Disributed Applications Experimental Results for VM Migration & Applications Performance Increasing

again and the performance of the applications increased.

With this solution to failures which may happen inside CDCs, we can assure a good performance for service applications and also assure a high QoS that mentioned in SLA document.

9.3.2 HIAs & RTAs

The plots in Fig. 9.6 show the experiments of highly interactive and real time (voice, video) applications. Fig. 9.6a) shows the impact of the virtual machine migration on the performance of HIA, voice and video applications during the bandwidth decreasing failure. The end-to-end delay time for the applications after increasing because of the failure, once the value of the bandwidth reach to 64 kb/s the mitigation technique start working and migrate the processing to the virtual machine. Then the end-to-end delay time return to decrease again and the performance of the applications increased.

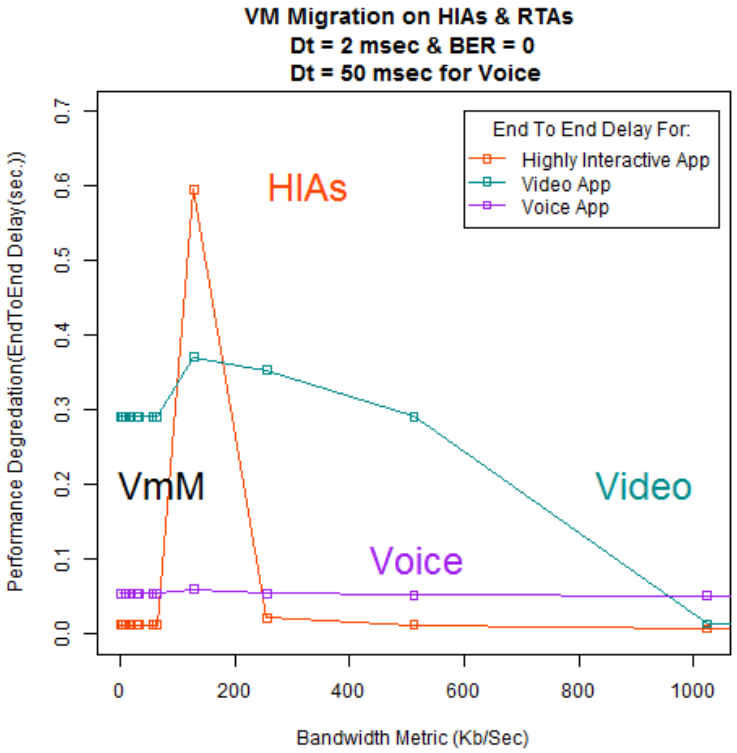
Also, Fig. 9.6b shows the impact of the virtual machine migration on the performance of HIA, voice and video applications during the delay time increasing on the link failure. The end-to-end delay time for the applications after increasing because of the failure, once the value of the delay time reach to 50 millisecond the mitigation technique start working and migrate the processing to the virtual machine. Then the end-to-end delay time return to decrease again and the performance of the applications increased.

Finally, Fig. 9.6c shows the impact of the virtual machine migration on the performance of HIA, voice and video applications during the bit error rate failure. The number of packet loss for the applications after increasing because of the failure, once the value of the error bit rate reach to 0.1 the mitigation technique start working and migrate the processing to the virtual machine. Then the number of packet loss return to decrease again and the performance of the applications increased.

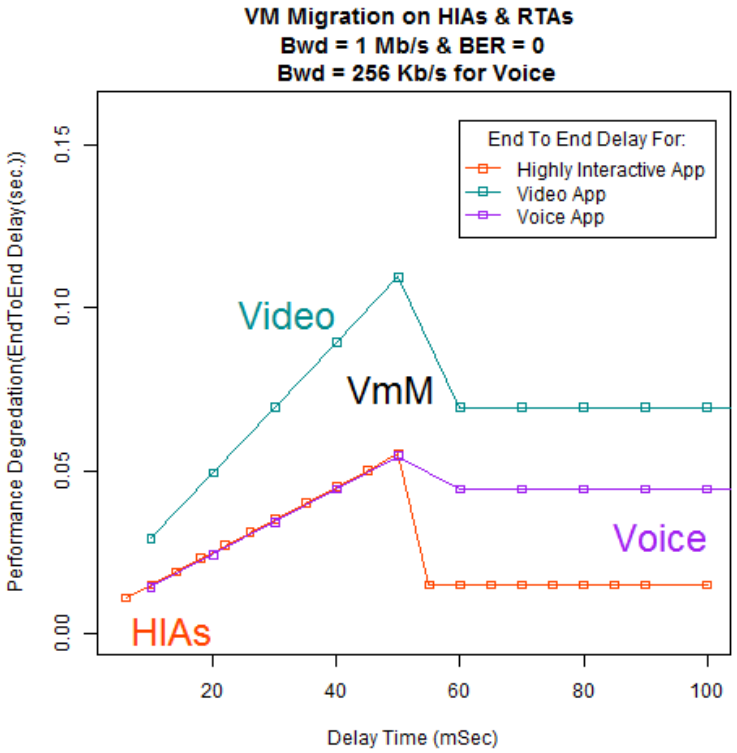
With this solution to failures which may happen inside CDCs, we can assure a good performance for service applications and also assure a high QoS that mentioned in SLA document.

For bit error rate failure, we can mitigate this failure by using FEC on the channel instead of using the virtual machine migration mitigation technique (See this in the next section 9.4). And for the other two failures, we used the virtual machine migration to mitigate these failures.

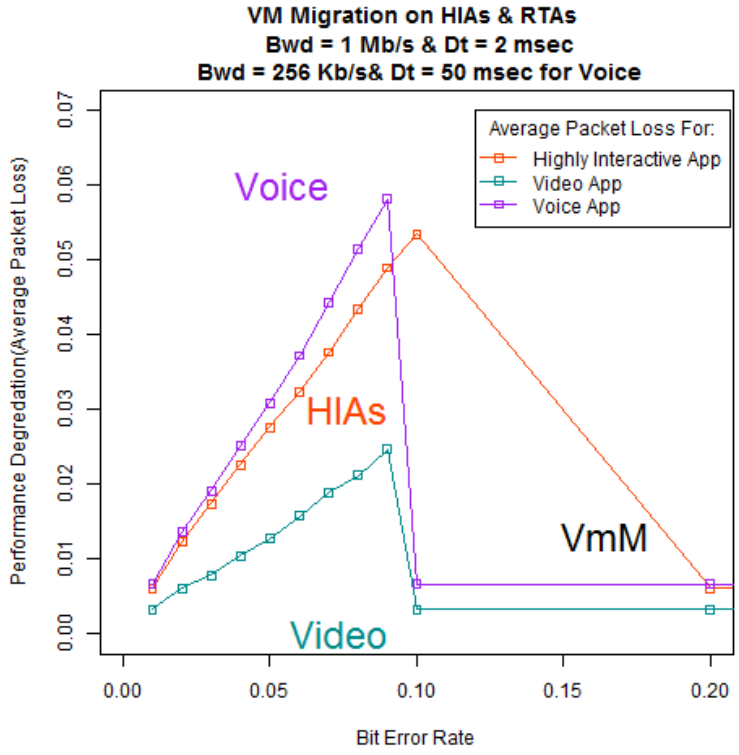
Table (9.1) summarizes all the experimental results for our work. And in the next chapter we will discuss the whole work in this master thesis. Also the following figures have a comparison between the applications performance degradation before and after failure mitigation. Fig. 9.7 shows the applications performance with failures Vs. after bandwidth degradation failure mitigation and this by representing the average response time and average end-to-end delay in web, file, distributed, highly interactive, voice and video applications. Fig. 9.8) shows the applications performance with failures Vs. after delay time increasing on the link failure mitigation and this by representing the average response time and average end-to-end delay in web, file, distributed, highly interactive, voice and video applications. Finally, Fig. 9.9 shows the applications performance with failures Vs. after bit error rate failure mitigation and this by representing the average packet loss in highly interactive, voice and video applications.



(a) Bandwidth Decreasing Failure Mitigation



(b) Delay Time Increasing Failure Mitigation



(c) Bit Error Rate Failure Mitigation

Figure 9.6: HIA & RTA Experimental Results for VM Migration & Applications Performance Increasing

Table 9.1: The Experimental Results of our Work

Application Classes	*Tested On?*	*Need failure mitigation?*	*Percentage of need*	*Mitigation technique used?*	*When Performance degraded?*			*Can work properly with Failures?*	*Performance Increasing*
					Bandwidth (kb/s)	*Delay time (msec.)*	*BER*		
Web	Real Scenario &Simulator	Need	Rarely	VMs Migration	≤ 64	≥ 10	> 0	No	97.89 %
File	Real Scenario &Simulator	Not Need	Very Rarely	VMs Migration	≤ 5	≥ 50	> 0	Yes	92.89 %
Distributed	Real Scenario &Simulator	Not Need	Very Rarely	VMs Migration	≤ 1	≥ 50	> 0	Yes	97.1 %
Highly Interactive	Simulator	Need	Often	VMs Migration or FEC for BER	≤ 64	≥ 50	≥ 0.03	No	98.1 %
Voice	Simulator	Need	Usually	VMs Migration or FEC for BER	≤ 28	≥ 50	≥ 0.01	No	97.13 %
Video	Simulator	Need	Usually	VMs Migration or FEC for BER	≤ 28	≥ 10	≥ 0.1	No	83.66 %
Chatting	Real Scenario	Need	Usually	VMs Migration	≤ 64	≥ 60	> 0	Yes	90.4 %

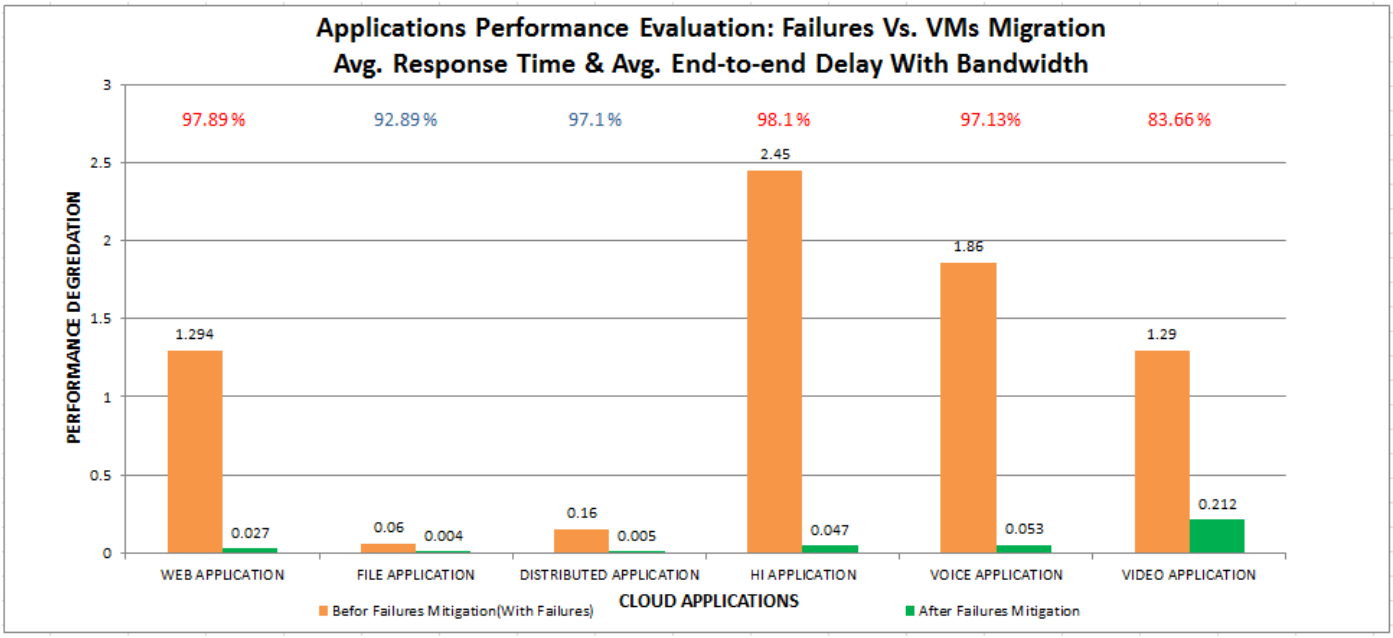


Figure 9.7: Failures Vs. VM Migration with Bandwidth Degradation

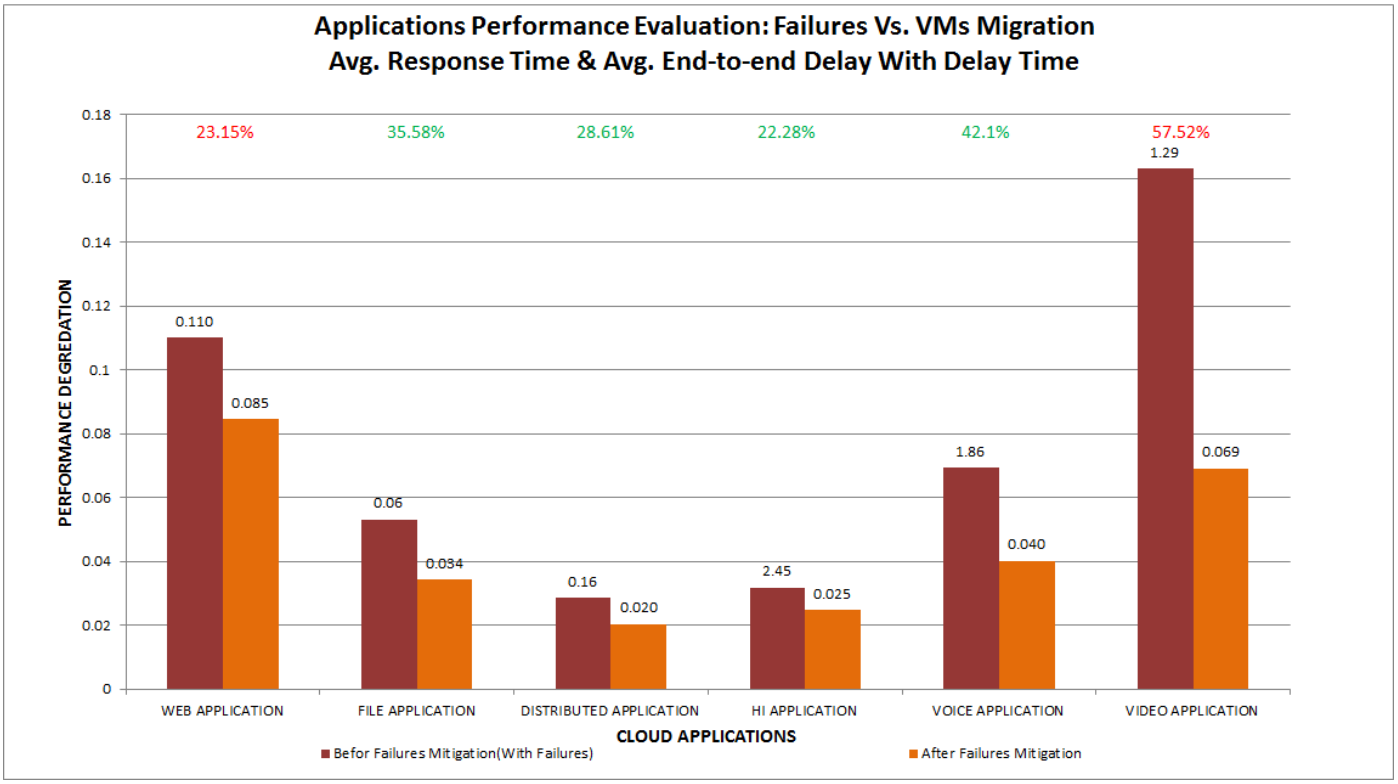


Figure 9.8: Failures Vs. VM Migration with Delay Time Increasing on the Link

From the previous three figures which summarize our results, we find some important outcomes which output after mitigating the failures and are represent some of our results in this master thesis work. The first is that, the average of the response time for web, file and distributed applications is decreasing after failure mitigation with a percentage of 95.957 %. The second is that the average end-to-end delay for the highly interactive, voice and video applications is decreasing after the failure mitigation with a percentage of 92.963 %. The third is that the packet loss for highly interactive, voice and video applications is decreasing after failure mitigation with a percentage of 52.018 %. Finally, From the general point of view, the average of the applications performance degradation is decreasing after solving and mitigating failures with a percentage of 94.461 %, in other words we can say that the applications performance is increasing with a percentage of 94.461 % after failures mitigation. In the previous results, we focused only on the most important metrics for the applications (e.g. response time, end to end delay and packet loss) and for the other metrics (e.g. latency, throughput, jitter,...etc) they also are enhanced after the failure mitigation with approximately the same values. Table (9.2) summarizes the previous findings.

In the next section, we show the results of the second failure mitigation technique we implemented. The second mitigation technique was FEC. We used it to mitigate BER failure.

Table 9.2: Summery of Findings

Application Metric	*Used to Test Application?*	*Failures Introduced?*	*Avg. with failures*	*Avg. After Mitigation*	*Percentage of Decreasing*
Response Time	Web, File and Distributed	Bandwidth Degradation. Delay time increasing on the link.	1.511007	0.036234	95.95722 %
End-to-end Delay	Highly interactive, Voice and Video	Bandwidth Degradation. Delay time increasing on the link. Bit error rate.	5.598944	0.311406	92.96348 %
Packet Loss	Highly interactive, Voice and Video	Bandwidth Degradation. Delay time increasing on the link. Bit error rate.	0.094841	0.047016	52.01798 %
General Performance Degradation	All the applications	All the failures	7.2	0.34769	94.46035 %

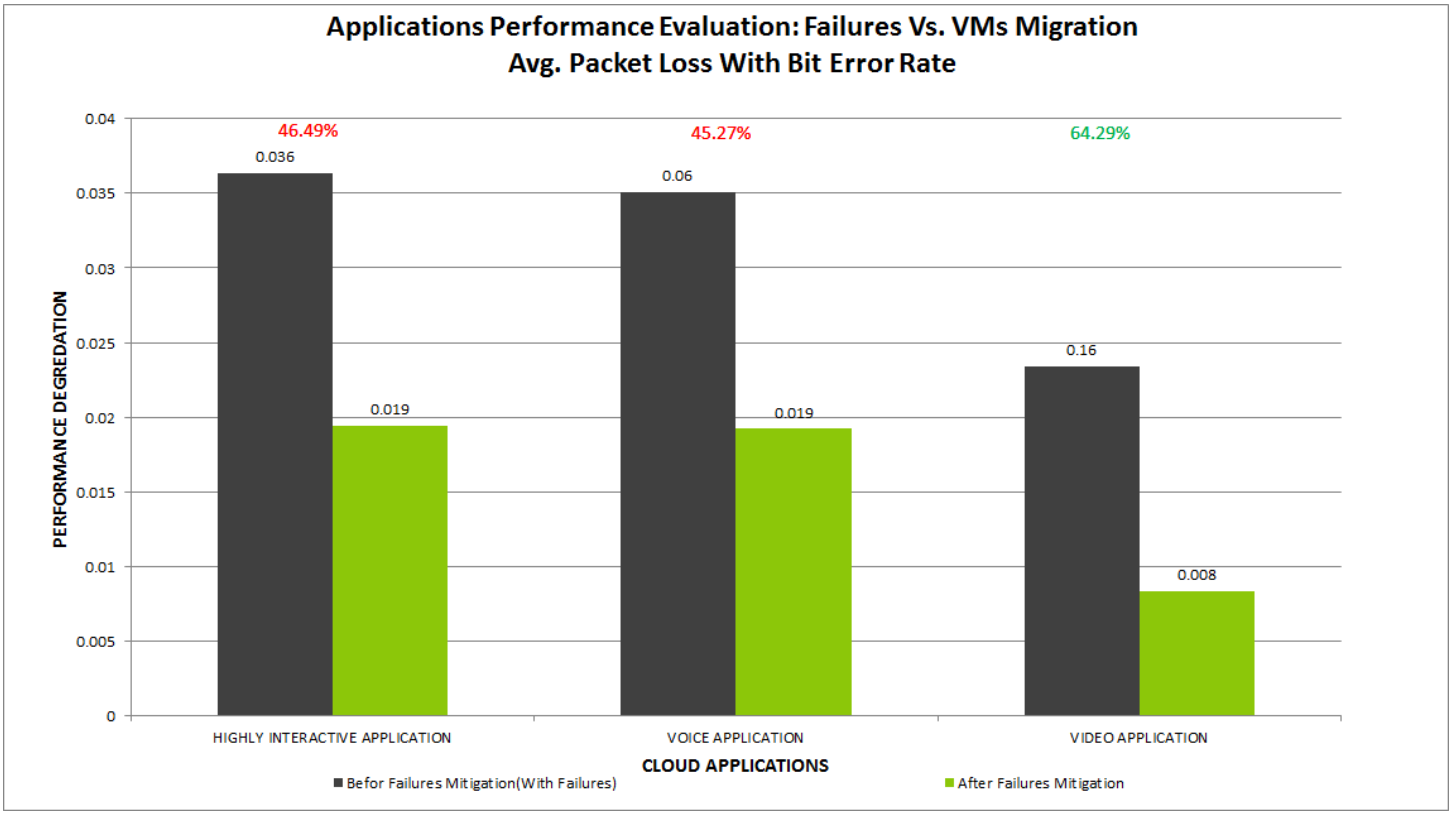


Figure 9.9: Failures Vs. VM Migration with Bit Error Rate

9.4 FEC Results

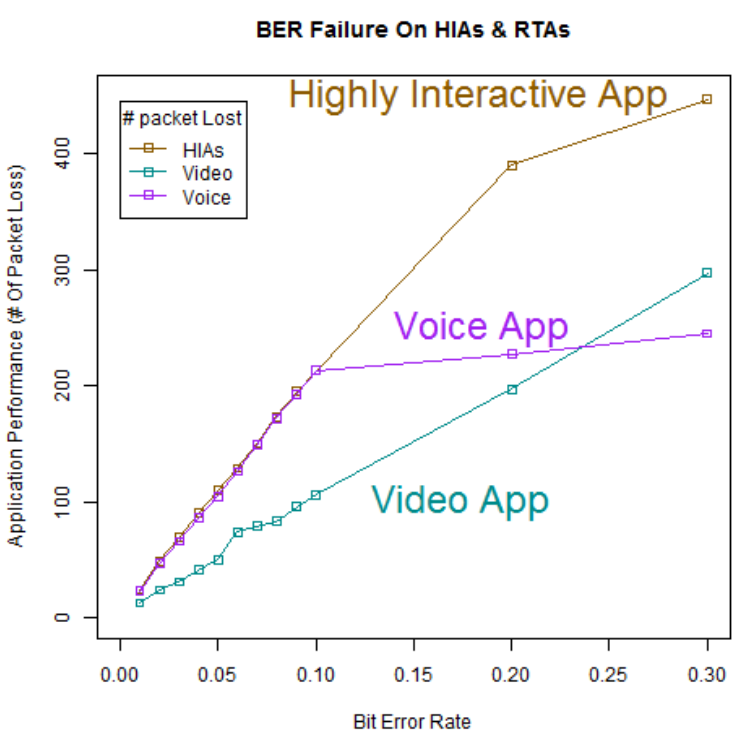
In this section, we show the results of FEC mitigation technique we implemented in the simulation experiments for the two classes of applications for the BER failure. The experimental results for FEC in HIA, voice and video applications are in Subsection 9.4.

FEC in HIA & RTA

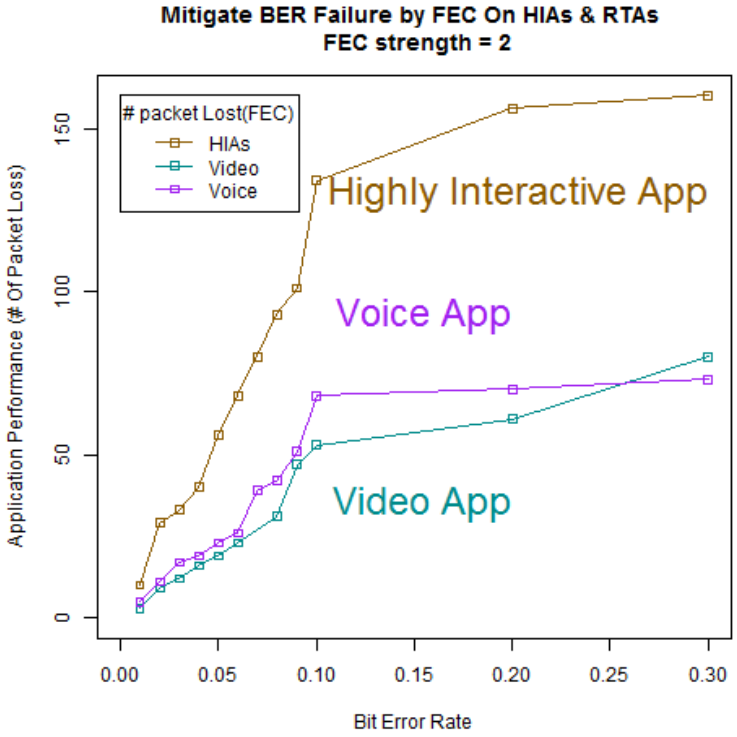
The plots in Fig. 9.10 show the impact of FEC mitigation technique on the performance of HIA, voice and video applications during the bit error rate failure is introduced. In the first Fig. 9.10a, it shows the number of packets lost during BER failure in HIA, voice and video applications and how the number of packet lost increasing by the increasing of the BER and that harm the applications performance. In the second Fig. 9.10b, it shows how the number of the packet lost is decreased after mitigating the BER failure by using FEC technique. And this let the performance degradation decreased again. Even if the BER increased the number of packet lost also increased but not that much like if there is no mitigation to the failures.

With this solution to failures which may happen inside CDCs, we can assure a good performance for service applications and also assure a high QoS that mentioned in SLA document.

Also, Fig. 9.11 shows preciously a comparison between the number of packet loss metric before and after mitigation by FEC. The figure shows that the average number



(a) Before FEC Mitigation



(b) After FEC Mitigation

Figure 9.10: FEC Mitigation Technique for BER Failure

of packet loss in the highly interactive application is decreased after mitigating BER by FEC with a percentage of 52.9 %. For voice application, the average number of packet loss is decreased after mitigating BER by FEC with a percentage of 73.12 %. For video application, the average number of packet loss is decreased after mitigating BER by FEC with a percentage of 64.6 %. Generally, the packet loss metric is decreasing after the failure mitigation by FEC with a percentage of 63.54 %.

In the next section, we introduce a comparison between the two mitigation techniques we have used to mitigate and solve failures. This comparison have been done to show which technique is better, and when we should use which one of the two ways of failure mitigation.

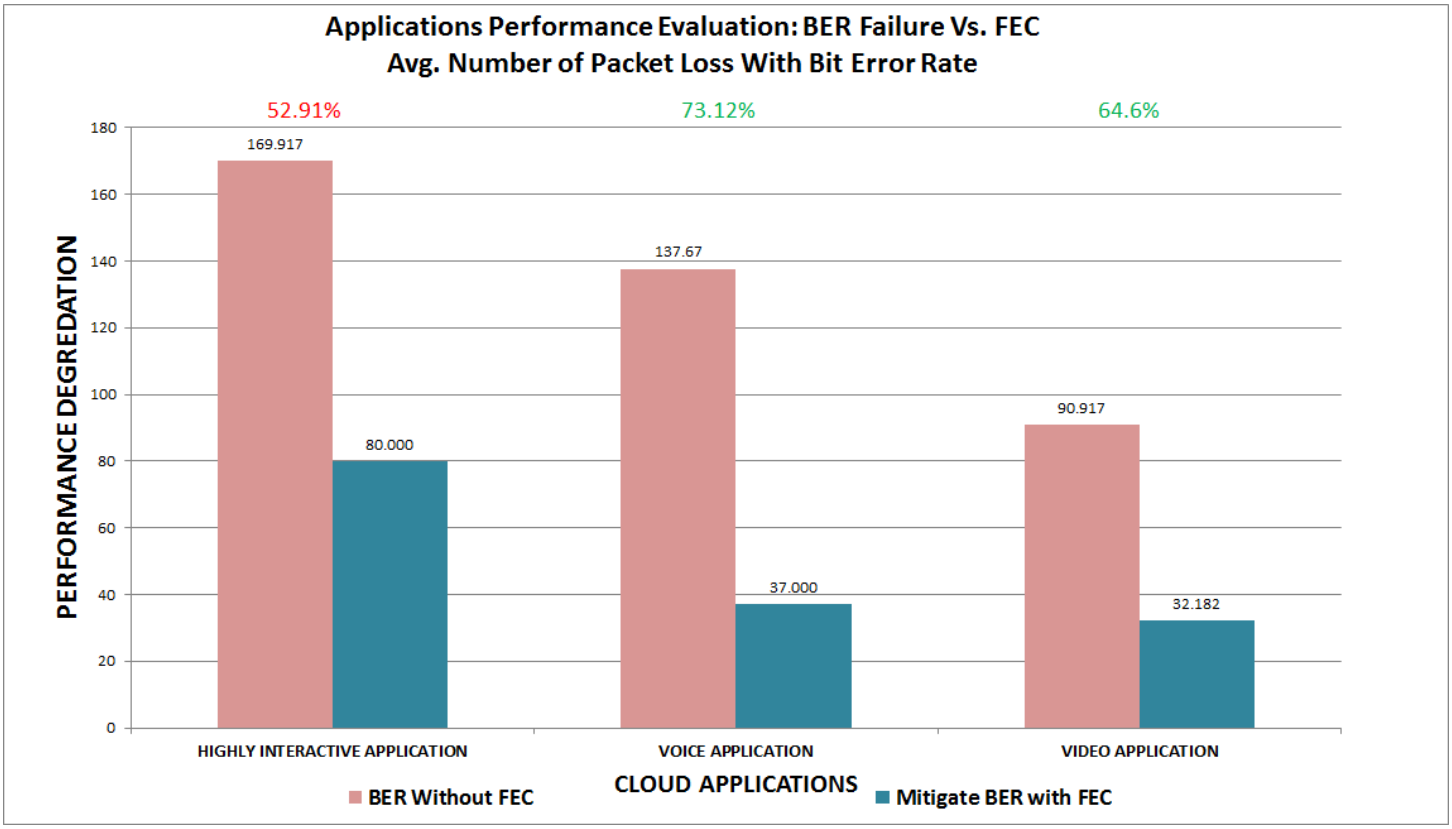


Figure 9.11: BER Failure Vs. FEC Mitigation

9.5 VM migration Vs. FEC

In this section, we do a comparison between the two mitigation techniques we used in our experiments. We used VM migration to mitigate all the failures and used the Forward Error Correction to mitigate the bit error rate failure. Fig. 9.12 shows the comparison between VM migration failure mitigation technique and FEC failure mitigation technique. But we do this comparison in the packet loss metric only and on the BER failure only where it is the common factor between the both mitigation techniques.

From the figure, we can see that mitigating BER failure with FEC is better than mitigating it with VM migration. Where the average of packet loss in the highly interactive application is decreased with FEC mitigation more than using VM migration with a percentage of 9.5%. And for voice application, the average of the packet loss is decreased with FEC mitigation more than using VM migration with a percentage of 56.1%. Finally, for video application, the average of the packet loss is decreased with FEC mitigation more than using VM migration with a percentage of 21.5%. Generally, the applications performance will be increased with using FEC to mitigate BER than using VM migration with a percentage of 29.02 %.

This percentage shows the difference between the packet loss metric values in case mitigate BER failure with VM migration (decreased with 52.018 %) and in case mitigate BER failure with FEC (decreased with 63.54 %) and that is a proof that FEC is better to mitigate BER failure.

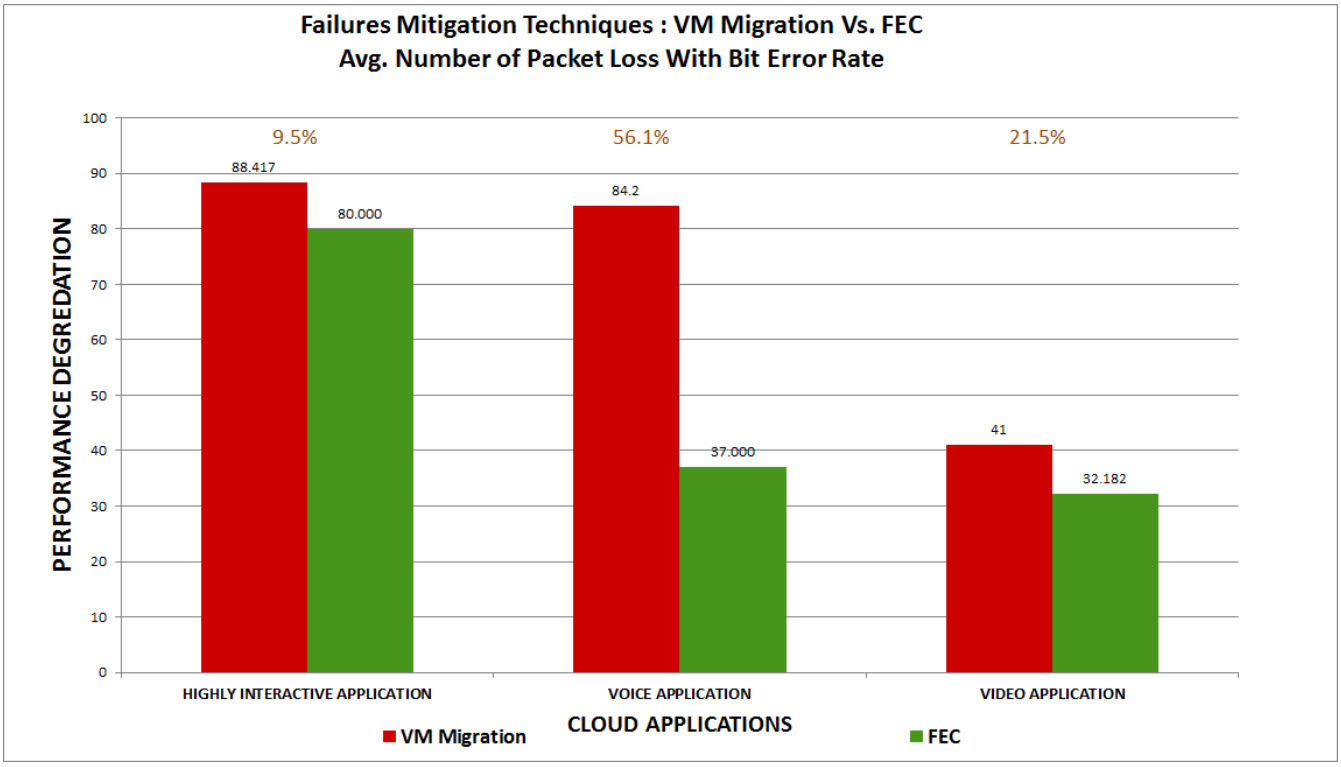


Figure 9.12: VM Migration Vs. FEC Mitigation Techniques

We still need to know when can we use FEC or use VM migration techniques, even if we see in our experiments that FEC is better or VM migration is better? The question should be, which method will is easy to use and take the less cost and also will increase the QoS and performance of applications (services). To answer this question, we need to know what is exactly the cost (time) when we use VM migration and how we will migrate? If we say that minimum and easiest way to migrate will take 2 seconds for instance. So, we can use VM migration as a mitigation technique for failures because this will increase the QoS and performance of applications and this what we want to assure in SLA where after migration, we will use a new link and device instead of the link or device which contain failures. If the migration will take time more than 2 seconds, like 10 or more, we can use FEC in the same link special for applications which can not wait for 10 seconds or more (e.g. HIA, RTA). In this situation, FEC will increase QoS and performance in the same link or device.

With explaining our experimental results, we approximately finish our work in this thesis. And with mitigating DCs failures, we assure a good performance for the applications and services as mentioned in SLA. We also can assure the SLA QoS and the response times metric which should covered in SLA document. With the assurance of services performance and high QoS, we can assure SLA between cloud providers and cloud user.

In the incoming sections, we discuss our work and review the challenges we have faced during this work. Also, we conclude our work and what we intend to do in our future work. In next section, we introduce a brief discussion which discuss totally our work in this thesis as a big view for the idea and how we proceeded.

Chapter X

Discussion

"Discussion is an exchange of knowledge; an argument an exchange of ignorance."

Robert Quillen

In this chapter, we discuss what we did in this master thesis work from scratch till the end. We discuss the general idea of the thesis project in Section 10.1. Then the discussion of the cloud computing applications in Section 10.2. In Section 10.3, we discuss the data center failures. Finally, the discussion of the cloud applications performance assessment and evaluation is in Section 10.4.

10.1 *The General Idea*

We propose a framework of solutions for assurance that the offered by cloud providers service is in line with the conducted SLA requirements. The best way to do that is to assess the performance of cloud applications provided by cloud data center providers. The assessment of cloud applications is based on testing them by introducing failures, to see the impact of failures on the performance of the applications. The performance of cloud applications for sure will suffer from the failures and will be degraded. Cloud application itself will try to survive failures.

If we solve failures that we introduced or at least mitigate those failures, then we guarantee the stability of cloud applications working, and the performance of cloud applications will no longer be degraded. At this point, we could assess the cloud application performance, hosted on cloud data centers and provided to cloud customers by cloud providers (The owners of the cloud data centers). This assessment should be provided to the cloud customers to assure and to verify that there is no violation in the service level agreement between them and cloud data center providers. The SLA assurance proof that the services provided to the cloud data center customers are working properly even if case any failure occurs. We assure a good QoS to applications and a minimum response times for any critical failures as mentioned in SLA document.

To evaluate cloud applications performance evaluation and assessment, we need to find first cloud applications hosted on data center, then secondly find the failures may occur in cloud data center and which have effects on cloud computing applications. After that, we need to test cloud applications found with the failures in the cloud data centers

to provide a study which describes each kind of applications strengths and weaknesses points during failures. The important is the weaknesses points and the problems occur to applications when failures occur. Then we solved or at least mitigate the problems to let cloud applications working with high performance or at least acceptable performance all the time. And left cloud customers unaware of any problems or failures occur in the data center. And keep high QoS to services and high applications performance as mentioned in SLA. Also keep the response times to any critical failures metric less as possible. With that, we made backup to SLA between cloud data centers services providers and cloud customers.

There are many related work that trying to do the same like ours, but they trying to develop and design software like the keep it moving [7] software which designed to reduce SLA violations in the large scale SaaS clouds. There are also some studies for private clouds like the Amazone EC2 SLA [13], which guarantee about 99.95 % of the availability of the services provided in a specific area over a 365 day period. So, what is the difference between our work and those studies? The new trend in our work, we tested generally most of the classes of applications with the most failures may happen in the cloud data center. Our work not related to SaaS layer only, it also related to the IaaS layer of the cloud. In our work, we don't care about software application it self where we assume that the software applications is working properly without failures. And we care about problems and failures may face the applications on the data center and harm or degraded the performance of cloud applications. The goal of our work is to play as a third party to verify the QoS and performance of the applications hosted on the cloud data centers, and provide this SLA assurance to cloud customers. This is the novelty in our research where usually the cloud providers say that they provide high availability,QoS and performance for services but nobody verify that, here in this master thesis we intended to verify and assure SLA (that no violations, and high availability applications) and if there is any violations it should reported to the both sides (Cloud Customers, Cloud Providers). In this case of violations in SLA, cloud provider should take actions to solve the problems (we may provide a solutions) and for the cloud customer, he should change cloud provider or ask him about the services performance and QoS (it should be high !!). Also in our future work we will try to test all cloud applications in a pure physical and real environment of cloud data center servers, switches and routers. Where here, in this thesis work we just test a real scenarios for 4 classes of application, and then test 6 application classes in NS-2 simulator.

10.2 *Cloud Computing Applications*

We classified all the cloud computing applications to eight different classes. Cloud applications provided as SaaS from the data center cloud provider to cloud customers. The classes of applications are web, file, distributed, real time, highly interactive, high performance, massive data analysis and mobile computing applications.

There are many requirements for the cloud computing applications like availability, QoS and scalability [57]. We tested in the real local scenarios only web, distributed, file and real time (chatting) applications, and in the simulator we tested the same four classes in addition to highly interactive application. There are many models for testing each class of applications, like for web application [79, 83–86], for the distributed application [82, 88, 89], for the real time application [80], for mobile application [81] and the

model for testing highly interactive application [87].

In our work, we got a mathematical model for testing four classes of the applications which tested in real scenarios. The models are a power equations which represent the graph line for testing each class of applications. We can use these models to predict the behavior of applications and also the performance which represented in the important metric for each class of applications. There are many metrics for each class of applications, we focus only on the most important ones for each class. The metrics such as latency, throughput, response time, delay time, end-to-end delay, jitter and number of packet loss..., etc.

The important metric for web application is the response time for http get, for file application is the response time of download request, for distributed application is the response time for sending e-mails as in the email application, for the real time application is the end-to-end delay, jitter and packet loss, and for the highly interactive application also the end-to-end delay is the most important metric. We tested all the previous metrics for each class of applications to see how the failures effect on the performance of the applications.

10.3 *Cloud Data Centers Failures*

There are many failures in cloud data centers such as servers failures, infrastructure & cooling failures, power consumption failures, software failures and the network failures. In our work, we focused on network failures where it represent about 60 % of the total failures in the cloud data centers, also the other physical failures like the servers failures which represent about 15 % of the total failures in the cloud data centers. The other failures we not care about where it is not related to our work. The solution for most of failure (e.g. Network, Physical server, Soft failures,...,etc.) is by redundancy, replacement and replications.

For the network failures [4] are also solved by redundancy, restarting and reactivating [49]. The redundancy in the network is three types like protocol, application and device redundancy. We used the device and application redundancy to solve failures introduced in our experimental work. We also used the data redundancy to solve some failures on the same link. To introduce network failures in our experiments, we use the degradation of bandwidth, the increasing of delay time on the link and the bit error rate as a network failures. In the real experiments, we just used bandwidth degradation as a failure, and in the simulator we use the three failures. We used the virtual machine migration and the forward error correction as a mitigation techniques for the failures in the simulator experiments.

10.4 *Cloud applications performance evaluation & assessment*

To assessment the applications during failures, we tested the applications in NS-2 simulator after testing them on real scenarios. The performance of applications represented by metrics, if these metrics are violated, the performance of applications is degraded. We introduced the three types of failures to the applications and see the behavior of the metrics of applications.

By the assessment of the applications performance, we found that web, file and distributed applications have less effect from the failures than the real time application and highly interactive applications. File and distributed applications can work properly even if there are failures, they just work with less performance. Also the chatting application can work with failures. Web, real time (voice, video) and highly interactive applications can not work during failures and they need for mitigation of the failures.

We solved the problem of bandwidth decreasing and the delay time increasing by using the virtual machine migration, and also the bit error rate failure we used the same technique in addition to the forward error correction technique on the channel as a second mitigation technique. In FEC there is no any devices or applications redundancy, the redundancy just in data on the same link and same devices which contain the failures. We used it in case the migration is not in an easy way and take long time to migrate. Also, FEC is better than VM migration if used to mitigate BER and this let the applications performance increased.

By solving failures and keep the performance of applications acceptable, we assured the SLA QoS and services performance, even if there is any failures may occur. All this in addition to assure a minimum value for response times metric which is the time to response to any critical failures as mentioned in SLA document.

In the next chapter, we will review most of the challenges we have faced during this thesis work. Also we will provide briefly how we solve those challenges or deal with them.

Chapter XI

Challenges

"Being challenged in life is inevitable, being defeated is optional"

Roger Crawford

In this chapter, we introduce some of the challenges we faced while developing and testing cloud applications hosted on cloud data centers. In this thesis work, we faced many challenges during our research and development work. We start by introducing the new trend challenge in the first Section 11.1. Then we introduce some challenges related to cloud computing applications in Section 11.2. We then discuss some of the challenges we faced while exploring the cloud data center failures in Section 11.3. We conclude this chapter by describing the challenges in the testing and implementation of our scenarios in Section 11.4.

11.1 New Trend

Our work in this master thesis is a new trend in the cloud computing applications and cloud data centers research and development. Usually, the cloud services providers companies provide a SLA between them and cloud customers. The companies verified that there are no violations to SLA and guarantee that the services availability and performance is about 99.99%. But, when the SLA document and contract release between cloud providers and cloud customers, it should be verified by another third party. Our work in this thesis is represent the work that the third party should do to verify that there is no violations to SLA really. And this is the novelty in our work, where there was nobody do this before. So, this work represent a new trend in the cloud computing data centers and cloud services providers and here it is the challenge.

11.2 Cloud Computing Applications

While trying to research for finding the cloud applications and testing them, we faced two main challenges. The first is that the diversity of cloud applications which is in Subsection11.2.1. The second is the searching for models to test the applications in reality, and this is in Subsection 11.2.2.

11.2.1 Diversity of Cloud Applications

To test cloud applications and get the performance of applications during failures in the cloud data center. We need to identify cloud applications. Because of the speed spreading of the cloud applications, there are many popular applications used by the cloud users. We faced a big challenge while identifying the cloud applications that hosted on cloud data centers due to the huge amount of the numerous cloud applications. To solve this challenge we categorized the cloud applications into eight different classes of applications such as web, file, distributed, real time, highly interactive, high performance, massive data analysis and mobile computing applications.

11.2.2 Model for Testing Cloud Applications

After classifying cloud applications into 8 classes, we need to test them and see their performance in the environment of failures. We were looking for models for testing each class of applications in reality, but unfortunately we didn't find what we want exactly however there are many work related with that. We solve this challenge by building a real environment for testing each class of applications and get a mathematical based models (power equations) for testing those classes.

11.3 Cloud Data Centers

The cloud data center architecture is much complicated and there are many components inside the data center like the cooling systems and the power systems in addition to the network inside the data center. Network is consisting of a group of connected devices such as switches , routers, load balancers and servers,.. etc. We just focus on the network of the data center where it related to our work. There are many failures in the network of the cloud data centers e.g. the device failures, server failures, soft failures,...etc and that is the challenge. Due to the diversity of the failures, it is difficult to introduce all those failures. Also, it is difficult to introduce those failures in the simulator or in the real scenarios for testing applications. So, we solved this challenge by introducing failures similar to the network failures such as the bandwidth degradation, delay time increasing on the link and the bit error rate.

11.4 Testing & Implementation

To test cloud applications in the real scenarios and in the simulator we faced some challenges. The challenges in the real scenarios testing are in Subsection 11.4.1. In Subsection 11.4.2, we review the challenges we faced in testing the applications in the simulator.

11.4.1 Real Testing

To test cloud applications, we need to develop real scenarios for each class of applications, but unfortunately we don't have real devices like servers, switches and routers, and there are no devices available for testing and that is the challenge. We solved this challenge by building the testing environment locally however it is not easy and need many servers to

setup locally. And also we tested 4 of the cloud application classes.

11.4.2 Simulator

To test cloud applications in NS-2 simulator, we need to know many languages and scripting languages like TCL, C++ and Perl. Also we need to simulate the applications protocols in the simulator. But NS-2 simulator not contain all the applications protocols which we should be used in to simulate the applications. The protocols not programmed in the simulator, we can add C++ classes to simulate the applications protocols. To add a new class we have to understand all the classes and the objects in the simulator code, and that is the challenge. We solved this challenge by tying to find already implemented protocols in the simulator which can used to test our applications. And also we tested 6 of the cloud application classes.

In the next chapter, we will finally conclude our work in this master thesis. We will review our contributions, findings and future work.

Chapter XII

Conclusion

"No one can be a great thinker who does not recognize that as a thinker it is his first duty to follow his intellect to whatever conclusions it may lead."

John Stuart Mill

In this master thesis, we propose a framework of solutions for assurance that the offered by cloud providers service is in line with the conducted SLA requirements. The best way to do that is to assess the performance of the cloud applications provided by the cloud data center providers. The assessment of the cloud applications is based on testing the cloud applications by introducing failures, to see the impact of the failures on the performance of the applications. The performance of the cloud applications for sure will suffer from the failures and will be degraded. The cloud application itself will try to survive failures. If we solve the failures that we introduced or at least mitigate those failures, then we guarantee the stability of the cloud applications working, and the performance of the cloud applications will no longer be degraded. At this point, we could assess the cloud application performance, hosted on the cloud data centers and provided to the cloud customers by cloud providers (The owners of the cloud data centers). This assessment should be provided to the cloud customers to assure and to verify that there is no violation in the service level agreement between the cloud data center providers and their customers. The SLA assurance proof that the services provided to the cloud data center customers are working properly even if case any failure occurs. We assure a good QoS to applications and a minimum response times for any critical failures as mentioned in SLA document.

We did that by classifying the cloud applications and identifying the cloud data center failures that effect the performance of the applications. We then introduced the failures found to the cloud applications during working in both real scenarios and a simulator. Then we evaluated and assessed the performance of the applications during the failures. Then we solved the failures by using the VMs migration or FEC and again evaluate and assessment the performance of the applications. By mitigating and solving the problems,we kept the cloud applications working at high performance or at least acceptable performance all the time. And left the cloud customers unaware of any problems or failures occur in the data center. With that, we made backup to the SLA between the cloud data centers services providers and cloud customers. We kept high QoS to services and high applications performance as mentioned in SLA. Also kept the response times to any critical failures metric.

In the first part of this chapter, we summarized our main contributions. The second part of this chapter, we summarized our main findings. The third part is dedicated to open problems that point to future work.

12.1 Summary of Contributions

The three main contributions of this master thesis are:

- ***Performance evaluation for the cloud applications*** : We did the assessment of the performance of the cloud applications while the failures of the cloud data centers were occurring. This assessment let us evaluate the cloud applications performance in many failure environments and let us know if the applications could continue working with failures or if they needed the solutions for the failures to work properly.

- ***Failures Mitigation*** : After introducing the data center failures to the cloud applications while running, we solved and mitigated the failures we introduced. The mitigation was done in simulator experiments by implementing the VM migration as a mitigation technique. Also the FEC mitigation technique was used to mitigate bit error rate failure.

- ***SLA Assurance*** : After the data center failures mitigation, we again evaluated the performance of the applications and guaranteed that the applications worked properly. The guarantee proved that there were no violations to the SLA between the cloud providers and the cloud customers, and this was the SLA assurance in the cloud computing data centers.

Here are the contributions of this master thesis in detail :

1. To identify & categorize the cloud computing applications into different classes.
2. To identify the cloud data centers failures and their impact on the performance of the cloud applications.
3. To develop real scenarios applications to test real failures on them (just 4 classes of applications), and get some plots.
4. To provide a mathematical model (power equation) based for testing some classes of applications.
5. To simulate the classes of applications on NS2 (Just 5 classes of the applications).
6. To introduce failures in the simulation and get some plots.
7. To solve and to mitigate the failures identified in the cloud DCs in the simulation (by using two mitigation techniques, VM migration & FEC), then get some plots.
8. Write a scientific conference paper to document our thesis project and publish it.

12.2 Summary of Findings

There are many findings, we found during the research and development (R & D)phases in this master thesis. There are findings related to the classes of the applications 12.2.1, other findings are related to the most important applications metrics 12.2.2 and finally those related to the failures introduced into the applications 12.2.3.

12.2.1 Cloud Applications

Here are the findings :

1. The eight classes of cloud applications are the web, file, distributed, real time, highly interactive, high performance, massive data analysis and mobile computing applications.

2. Cloud data centers failures are classified such as the network, physical server, VMs availability, soft failures and single point of failures,..etc. In addition to the problems causing each category of failures.

3. The file and distributed applications have little of failure mitigation and if they did need it, it would be very rarely. They also can work properly even if the failures exist. Also the chat application can work while failures are occurring. They do not need a very high bandwidth on the network. The error bit rate has no big impact on their performance.

4. The web applications rarely need failure mitigation and it can not work properly while failures exist. They need a high bandwidth on the network. The error bit rate has little impact on their performance.

5. The highly interactive applications often need failure mitigation and it cannot work properly while failures exist. Also it need a high bandwidth on the network to provide high availability and scalability. They need low delay time in the network links. The bit error rate has little impact on their performance.

6. Voice and video applications usually need failure mitigation and they cannot work properly while failures exist. They do not need a high bandwidth on the network, but need a QoS on the network devices to get high priority for their packets on the network. They need low delay time in the network links. The bit error rate has more effect on the voice than on the video.

12.2.2 Important Metrics

This comprises the findings related to the most important metrics from the cloud applications, and which are summarized in Fig. 9.7, 9.8, 9.9 & 9.11 and the tables (9.1 & 9.2). They may be summarized as follows:

- ***Response Time Metric*** : The average value of this metric is decreased after failures mitigation by 95.957 %.

- ***End-to-end Delay Metric*** : The average value of this metric is decreased after failures mitigation by 92.963 %.

- ***Packet Loss Metric*** : The average value of this metric is decreased after failures mitigation by 52.018 % if BER failure is mitigated with the VM migration, and with by 63.54% if BER failure is mitigated with the FEC mitigation technique.

- ***The applications performance Evaluation*** : The performance of the applications is increased after failure mitigation by 94.46165 %.

12.2.3 Failures Introduced

Finally, there are other findings related to the failures we introduced into the applications (e.g. Bandwidth Degradation, Delay time increasing on link and Bit error rate).

- ***Bandwidth Degradation Failure*** : Mitigated by using VM migration on all the applications classes where it was introduced to all the applications tested. This failure has a high impact on the performance of 66.6 % of applications tested e.g. web, HIA, voice and video. And for file and distributed applications there is less impact on the performance. This failure harms the response, delay and end to end delay time metrics of the applications and also the other metrics like latency, throughput and packet loss but not as much as the most important metrics mentioned above.

- ***Delay Time on Link Failure*** : Mitigated by using VM migration on all the applications classes where it was introduced on all the applications tested. This failure has a high impact on the performance of 50 % of applications tested like web, voice and video applications. And for file, HIA and distributed applications there is less impact on the performance. This failure harms the response, delay and end to end delay time metrics of the applications and also the other metrics like latency, throughput and packet loss but not as much as the most important metrics mentioned earlier.

- ***Bit Error Rate Failure*** : Mitigated by using both mitigation techniques, VM migration and FEC. It was introduced to highly interactive and real time applications classes and not to the others where it would have no impact on their important metrics (performance). This failure has more impact on the performance of HIA and voice application than on video application. It harms the packet loss metric and it has no impact on the other important metrics. The best way to mitigate this failure is by using FEC mitigation technique, where FEC is better than VM migration if migration take long time. By using FEC to mitigate BER the applications performance increased by 29.02 % more than using VM migration to mitigate BER failure.

12.3 Future Work

In this section, we indicate future work, and prospective improvements that we hope to do it. We intend to simulate our experiments in a big data center topology in the GreenCloud[1] simulator. We look forward to continuing to test the remain classes of cloud applications in real scenarios and in simulator. We also look forward to moving all our experiments to a pure real environment of the cloud data centers with real devices (e.g. switches, routers, servers,...etc.).

We look forward to providing a flexible framework for testing cloud applications and providing a good summary for SLA violations if they exist. The novel framework will provide solutions for SLA violations in cloud data centers owned by cloud providers. We also hope to improve this framework to be a third party between cloud providers and cloud users giving SLA assurance in the cloud data centers by improving the quality of the services provided to cloud customers and to solve the problems which degraded the performance of cloud applications whether in the SaaS layer or in the IaaS layer.

[1]The green Cloud Simulator [18]: http://greencloud.gforge.uni.lu/

Bibliography

[1] B. Furht and A. Escalante, *Handbook of Cloud Computing.* Springer Science and Business Media, LLC 2010.

[2] P. Patel, A. Ranabahu, and A. Sheth, "Service Level Agreement in Cloud Computing," *The Ohio Center of Excellence in Knowledge-Enabled Computing (Kno.e.sis)*, 2009.

[3] M. Ahmed, R. Chowdhury, M. Ahmed, and M. H. Rafee, "An advanced survey on cloud computing and state-of-the-art research issues," *In IJCSI International Journal of Computer Science Issues*, vol. Vol. 9, Issue 1, No 1, p. 1: 718, (January 2012).

[4] P. Gill, N. Jain, and N. Nagappan, "Understanding Network Failures in Data Centers: Measurement, Analysis, and Implications," *ACM, SIGCOMM'11, Toronto, Ontario, Canada*, August 15-19, 2011.

[5] School of Engineering Internet Engineering, "NS2 (Network Simulator version 2) Manual," *Information Sciences Information and Communications Technology*, 2010.

[6] P. Meeneghan and D. Delaney, "An Introduction to NS, Nam and OTcl scripting," *National University of Ireland*, April 2004.

[7] A. Roy, R. Ganesan, and S. Sarkar, "Keep It Moving: Proactive workload management for reducing SLA violations in large scale SaaS clouds," 2013 IEEE.

[8] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The Cost of a Cloud: Research Problems in Data Center Networks," *ACM SIGCOMM*, January 2009.

[9] Ranjithprabhu.K and Sasirega.D, "Eliminating single point of failure and data loss in cloud computing," *International Journal of Science and Research (IJSR)*, vol. Volume 3 Issue 4, p. 2319 : 7064, (April 2014).

[10] S. Sankar and S. Gurumurthi, "Soft Failures in Large Datacenters," *IEEE*, (Jul 2013).

[11] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and re-search challenges," *Springer,J Internet Serv Appl*, p. 1: 718, (2010).

[12] G. Reese, M. Benioff, and M. Miller, "Cloud computing tutorial, http://thecloudtutorial.com/related.html."

[13] A. T. Velte, T. J. Velte, and R. Elsenpeter, *Cloud Computing: A Practical Approach.* The McGraw-Hill Companies, 2010.

[14] C. Baun, M. Kunze, J. Nimis, and S. Tai, *Cloud Computing Web-Based Dynamic IT Services.* Springer-Verlag Berlin Heidelberg, 2011.

[15] Salman A. Baset, "Cloud SLAs: Present and Future," *IBM Research, ACM SIGOPS Operating Systems Review*, vol. 46 Issue 2, pp. 57–66, July 2012.

[16] SAP Data Center, "How a data center works, http://www.sapdatacenter.com/."

[17] K. Bilal, S. U. R. Malik, and S. U. Khan, "Trends and Challenges in Cloud Datacenters," *IEEE CLOUD COMPUTING SOCIETY*, 2014.

[18] D. Kliazovich, P. Bouvry, and S. U. Khan, "GreenCloud: A Packetlevel Simulator of Energy-aware Cloud Computing Data Centers," *Journal of Supercomputing*, vol. vol. 62, no. 3, pp. pp. 1263–1283, 2012.

[19] Tcl Developer Xchange site, "Tcl developer xchange, http://www.tcl.tk/."

[20] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Sandpiper:Black-box and gray-box resource management for virtual machines," *Computer Networks*, vol. vol. 53, no. 17, p. 2923–2938, 2009.

[21] Z. Shen, S. Subbiah, X. Gu, and J. Wilkes, "Cloudscale: elastic resource scaling for multi-tenant cloud systems," *in Proc. SOCC. ACM*, p. pp. 5–19, 2011.

[22] G. Khanna, K. Beaty, G. Kar, and A. Kochut, "Application performance management in virtualized server environments," *Proc. NOMS.IEEE*, p. pp. 373–381, 2006.

[23] L. Eyraud-Dubois and H. Larchevêque, "Optimizing resource allocation while handling sla violations in cloud computing platforms," *Proc. IPDPS. IEEE*, 2013.

[24] R. Gupta, S. K. Bose, S. Sundarrajan, M. Chebiyam, and A. Chakrabarti, "A two stage heuristic algorithm for solving the server consolidation problem with item-item and bin-item incompatibility constraints," *Proc. SCC. IEEE*, p. pp. 39–46, 2008.

[25] S. Kandula, R. Mahajan, P. Verkaik, S. Agarwal, J. Padhye, and P. Bahl, "Detailed diagnosis in enterprise networks," *SIGCOMM*, 2010.

[26] V. Padmanabhan, S. Ramabhadran, S. Agarwal, and J. Padhye, "A study of end-to-end web access failures," *CoNEXT*, 2006.

[27] C. Labovitz and A. Ahuja, "Experimental study of internet stability and wide-area backbone failures," *In The Twenty-Ninth Annual International Symposium on Fault-Tolerant Computing*, 1999.

[28] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, Y. Ganjali, and C. Diot, "Characterization of failures in an operational IP backbone network," *IEEE/ACM Transactions on Networking*, 2008.

[29] A. Shaikh, C. Isett, A. Greenberg, M. Roughan, and J. Gottlieb, "A case study of OSPF behavior in a large enterprise network," *ACM IMW*, 2002.

[30] D. Turner, K. Levchenko, A. C. Snoeren, and S. Savage, "Understanding the causes and impact of network failures," *SIGCOMM*, 2010.

[31] D. Watson, F. Jahanian, and C. Labovitz, "Experiences with monitoring OSPF on a regional service provider network," *ICDCS*, 2003.

[32] K. V. Vishwanath and N. Nagappan, "Characterizing cloud computing hardware reliability," *ACM SoCC'10*, 2010.

[33] B. Schroeder, E. Pinheiro, and W.-D. Weber, "DRAM errors in the wild: A large-scale field study," *SIGMETRICS*, 2009.

[34] D. Ford, F. Labelle, F. Popovici, M. Stokely, V. Truong, L. Barroso, C. Grimes, and S. Quinlan, "Availability in globally distributed storage systems," *OSDI*, 2010.

[35] B. Schroeder and G. Gibson, "Disk failures in the real world: What does an MTTF of 1,000,000 hours mean too you?," *FAST*, 2007.

[36] K. V. Vishwanath and N. Nagappan, "Characterizing cloud computing hardware reliability.," *Symposium on Cloud Computing (SOCC)*, 2010.

[37] R. Potharaju and N. Jain, "When the Network Crumbles: An Empirical Study of Cloud Network Failures and their Impact on Services," *ACM SoCC'13*, Oct. 2013.

[38] D. Banerjee, V. Madduri, and M. Srivatsa, "A Framework for Distributed Monitoring and Root Cause Analysis for Large IP Networks," *SRDS '09*, 2009.

[39] S. Kandula, D. Katabi, and J.-P. Vasseur, "Shrink a tool for failure diagnosis in IP networks," *MineNet*, 2005.

[40] Kompella, Yates, Jennifer, G. Albert, and S. Alex, "IP Fault Localization Via Risk Modeling," *NSDI*, 2005.

[41] P. Bahl, R. Chandra, A. Greenberg, S. Kandula, D. A. Maltz, and M. Zhang, "Towards highly reliable enterprise network services via inference of multi-level dependencies," *SIGCOMM*, 2007.

[42] S. Kandula, R. Mahajan, P. Verkaik, S. Agarwal, J. Padhye, and P. Bahl, "Detailed diagnosis in enterprise networks," *SIGCOMM*, 2005.

[43] M. K. Aguilera, J. C. Mogul, J. L.Wiener, P. Reynolds, and A. Muthitacharoen, "Performance debugging for distributed systems of black boxes," *SOSP*, 2003.

[44] M. Y. Chen, A. Accardi, E. Kiciman, J. Lloyd, D. Patterson, A. Fox, and E. Brewer, "Path-based failure and evolution management," *NSDI*, 2004.

[45] P. Reynolds, J. L.Wiener, J. C. Mogul, M. K. Aguilera, and A. Vahdat, "WAP5: black-box performance debugging for wide-area systems," *WWW*, 2006.

[46] M. Isard, "Autopilot: automatic data center management," *SIGOPS Oper. Syst. Rev.*, vol. 41, p. 60–67, April 2007.

[47] Y. Wang, H. Wang, A. Mahimkar, R. Alimi, Y. Zhang, L. Qiu, and Y. R. Yang, "R3: resilient routing reconfiguration," *SIGCOMM*, 2010.

[48] R. B. Kiran, K. Tati, Y. chung Cheng, S. Savage, and G. M. Voelker, "TotalRecall: System Support for Automated Availability Management," *NSDI*, 2004.

[49] X. Wu, D. Turner, C. Chen, D. A. Maltz, X. Yang, L. Yuan, and M. Zhang, "NetPilot: Automating Datacenter Network Failure Mitigation," *ACM SIGCOMM'12*, 2012.

[50] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," *NSDI'05 USENIX Association*, p. pp. 273–286, 2005.

[51] F. Hermenier, X. Lorca, J.-M. Menaud, G. Muller, and J. Lawall, "Entropy: a consolidation manager for clusters," *ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, March 11-13, 2009, Washington, DC, USA.

[52] Verma, P. Ahuja, and A. Neogi, "Power and migration cost aware application placement in virtualized systems," *Technical report, IBM*, 2008.

[53] X. Meng, Y. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," *IEEE INFOCOM*, 2010.

[54] J. Chen, W. Liu, and J. Song, "Network Performance-Aware Virtual Machine Migration in Data Centers," *The Third International Conference on Cloud Computing, GRIDs, and Virtualization*, 2012.

[55] V. Shrivastava, P. Zerfos, K. won Lee, H. Jamjoom, Y.-H. Liu, and S. Banerjee, "Application-aware Virtual Machine Migration in Data Centers," *IEEE INFOCOM*, 2011.

[56] F. P. Tso, G. Hamilton, K. Oikonomou, and D. P. Pezaros, "Implementing Scalable, Network-Aware Virtual Machine Migration for Cloud Data Centers," *IEEE Sixth International Conference on Cloud Computing*, 2013.

[57] M. Ali and M. H.Miraz, "Cloud Computing Applications," *Internaional Conference on Cloud Compuing and eGovernance*, 2013.

[58] P. Dhar, "Cloud computing and its applications in the world of networking," *International Journal of Computer Science*, vol. Vol. 9, Issue 1, No 2, pp. pp. 1694–0814, January 2012.

[59] E. Proko and I. Ninka, "Analyzing and testing web application performance," *International Journal Of Engineering And Science*, vol. Vol.3, Issue 10, pp. pp. 47–50, October 2013.

[60] W. Shang, Z. M. Jiang, H. Hemmati, B. Adams, A. E. Hassan, and P. Martin, "Assisting developers of big data analytics applications when deploying on hadoop clouds," *IEEE,ICSE*, 2013.

[61] A. Valadares and C. V. Lopes, "A Framework for Designing and Evaluating Distributed Real-Time Applications," *IEEE/ACM 18th International Symposium on Distributed Simulation and Real Time Applications*, 2014.

[62] Q. Xu, S. Mehrotra, Z. M. Mao, and J. Li, "PROTEUS: Network Performance Forecast for Real-Time, Interactive Mobile Applications," *ACM, MobiSys'13*, 2013.

[63] D. D. Clark, S. Shenker, and L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism," *ACM, SIGCOMM '92 Conference proceedings on Communications architectures & protocols*, 1992.

[64] H. T. Dinh, C. Lee, D. Niyato, and P. Wangi, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wirel. Commun. Mob. Comput*, 2013.

[65] S. S. Qureshi, T. Ahmad, K. Rafique, and S. ul islam, "MOBILE CLOUD COMPUTING AS FUTURE FOR MOBILE APPLICATIONS - IMPLEMENTATION METHODS AND CHALLENGING ISSUES," *IEEE CCIS*, 2011.

[66] Q. Fu, W. Zhang, and B. Wang, "Energy efficient mobile cloud computing and its applications," *9th IEEE International Conference on Networking, Architecture, and Storage*, 2014.

[67] A. Gupta, F. Gioachin, and P. Faraboschi, "The Who, What, Why and How of High Performance Computing Applications in the Cloud," *Hewlett-Packard Development Company, L.P*, 2013.

[68] W. Shang, Z. M. Jiang, H. Hemmati, B. Adams, A. E. Hassan, and P. Martin, "Performance analysis of high performance computing applications on the amazon web services cloud," *2nd IEEE International Conference on Cloud Computing Technology and Science*, 2010.

[69] S. Bhola and M. Ahamad, "Workload Modeling for Highly Interactive Applications," *ACM, SIGMETRICS '99*, 1999.

[70] M. A. Sharaf, P. K. Chrysanthis, A. Labrinidis, and C. Amza, "Optimizing I/O-Intensive Transactions in Highly Interactive Applications," *ACM, SIGMOD'09*, 2009.

[71] M. Shiraz, A. Gani, S. Member, R. H. Khokhar, and R. Buyya, "A Review on Distributed Application Processing Frameworks in Smart Mobile Devices for Mobile Cloud Computing," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, vol. VOL. 15, NO.3, THIRD QUARTER 2013.

[72] A. Papaioannou and K. Magoutis, "An Architecture for Evaluating Distributed Application Deployments in Multi-Clouds," *IEEE International Conference on Cloud Computing Technology and Science*, 2013.

[73] J. S. Veen, E. Lazovik, M. X. Makkes, and R. J. Meijer, "Deployment Strategies for Distributed Applications on Cloud Computing Infrastructures," *IEEE International Conference onCloud Computing Technology and Science (CloudCom)*, 2013.

[74] S. Bhola and M. Ahamad, "Synthesis of Real Time Distributed Applications for Cloud Computing," *IEEE Conference on Computer Science and Information Systems*, 2014.

[75] X. Etchevers, T. Coupaye, F. Boyer, and N. de Palma, "Self-configuration of distributed applications in the cloud," *IEEE International Conference on Cloud Computing (CLOUD)*, 2011.

[76] G. Salaün, X. Etchevers, N. D. Palma, F. Boyer, and T. Coupaye, "Verification of a Self-configuration Protocol for Distributed Applications in the Cloud," *ACM, SAC'12*, 2012.

[77] W. Chan, L. Mei, and Z. Zhang, "Modeling and testing of cloud applications," *IEEE APSCC*, 2009.

[78] J. Gao, X. Bai, and W.-T. Tsai, "Cloud Testing- Issues, Challenges, Needs and Practice," *Software Engineering : An International Journal (SEIJ*, vol. Vol. 1, No. 1, 2011.

[79] H. Reza, K. Ogaard, and A. Malge, "A MODEL BASED TESTING TECHNIQUE TO TEST WEB APPLICATIONS USING STATECHARTS," *IEEE Fifth International Conference on Information Technology: New Generations*, 2008.

[80] S. Bensalem, M. Bozga, M. Krichen, and S. Tripakis, "Testing conformance of real-time applications by automatic generation of observers," *Elsevier*, 2004.

[81] C. Tao and J. Gao, "Modeling Mobile Application Test Platform and Environment: Testing Criteria and Complexity Analysis," *ACM, JAMAICA '14*, July 2014.

[82] H.-D. Chu and J. E. Dobson, "An Integrated Test Environment For Distributed Applications," 1997.

[83] S. Elbaum, S. Karre, and G. Rothermel, "Improving Web Application Testing with User Session Data," *25th International Conference on Software Engineering*, 2003.

[84] N. Rafique, N. Rashid, S. Awan, and Z. Nayyar, "Model based testing in web applications," *International Journal of Scientific Engineering and Research (IJSER)*, vol. Volume 2 Issue 1, January 2014.

[85] J. Ernits, R. Roo, J. Jacky, and M. Veanes, "Model-Based Testing of Web Applications using NModel," in *Testing of Software and Communication Systems*, Springer, 2009.

[86] A. A. Andrews, JeffOffutt, and R. T. Alexander, "Testing Web Applications by Modeling with FSMs," *Springer-Verlag*, January 2005.

[87] R. Bastide, D. Navarre, and P. Palanque, "A Model-Based Tool for Interactive Prototyping of Highly Interactive Applications," *ACM*, 2002.

[88] D. Gupta, K. V. Vishwanath, and A. Vahdat, "DieCast: Testing Distributed Systems with an Accurate Scale Model," *ACM, 5th USENIX Symposium on Networked Systems Design and Implementation*, 2011.

[89] G. Denaro, A. Polini, and W. Emmerich, "Early Performance Testing of Distributed Software Applications," *ACM SIGSOFT*, 2004.

[90] A. Ahmed and A. S. Sabyasachi, "Cloud Computing Simulators: A Detailed Survey and Future Direction," *IEEE*, 2014.

[91] R. N. Calheiros, R. Ranjan, A. Beloglazov, F. D. Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Wiley Online Library*, August 2010.

[92] A. Núñez, J. L. Vázquez-Poletti, A. C. Caminero, G. G. Castañé, J. Carretero, and I. M. Llorente, "iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator," *Journal of Grid Computing*, vol. Volume 10, Issue 13, pp. pp. 185–209, March 2012.

[93] R. Wickremasinghe, R. N. Calheiros, and R. Buyya, "CloudAnalyst: A CloudSim-based Visual Modeller for analysing Cloud Computing Environments and Applications," *24th IEEE International Conference on Advanced Information Networking and Applications*, 2010.

[94] S. Lim, B. Sharma, G. Nam, K. Kim, and C. R. Das, "MDCSim: A Multi-tier Data Center Simulation Platform," *Cluster Computing and Workshops*, 2009.

[95] S. K. Garg and R. Buyya, "NetworkCloudSim: modelling parallel applications in cloud simulations," *Utility and Cloud Computing (UCC), Fourth IEEE International Conference*, 2011.

[96] R. N. Calheiros, M. A. S. Netto, C. A. F. D. Rose, and R. Buyya, "EMUSIM: an integrated emulation and simulation environment for modeling, evaluation, and validation of performance of cloud computing applications," *Software-Practice and Experience*, 2012.

[97] S. Ostermann, K. Plankensteiner, and R. Prodan, "GroudSim: An Event-Based Simulation Framework for Computational Grids and Clouds," *Parallel Processing Workshops Lecture Notes in Computer Science*, 2011.

[98] M. Tighe, G. Keller, M. Bauer, and H. .Lutfiyya, "DCSim: A Data Centre Simulation Tool for Evaluating Dynamic Virtualized Resource Management," *8th international conference and 2012 workshop on systems virtualization management (svm) Network and service management (cnsm)*, 2012.

[99] J. Jung and H. Kim, "MR-CloudSim: Designing and implementing MapReduce computing model on CloudSim," *International Conference on ICT Convergence (ICTC)*, 2012.

[100] M. Shiraz, A. Gani, R. H. Khokhar, and E. Ahmed, "An Extendable Simulation Framework for Modeling Application Processing Potentials of Smart Mobile Devices for Mobile Cloud Computing," *10th International Conference on Frontiers of Information Technology*, 2012.

[101] S. Sotiriadis, N. Bessis, N. Antonopoulos, and A. Anjum, "SimIC: Designing a new Inter-Cloud Simulation platform for integrating largescale resource management," *IEEE 27th International Conference on Advanced Information Networking and Applications*, 2013.

[102] T. Kumar, R. E. Cledat, and S. Pande, "Dynamic Tuning of Feature Set in Highly Variant Interactive Applications," *ACM, EMSOFT'10*, 2010.