2010

Time delay and its effect in a virtual lab created using cloud computing

Juan Nunez

Time Delay and its Effect in a Virtual Lab Created using Cloud Computing

By

Juan Nunez

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Networking and Systems Administration

Rochester Institute of Technology

B. Thomas Golisano College
of
Computing and Information Sciences

April 2010

Rochester Institute of Technology

B. Thomas Golisano College
of
Computing and Information Sciences

Master of Science in
Networking and Systems Administration

Thesis Approval Form

Student Name: **Juan Nunez**

Thesis Title: **Time Delay and its Effect in a Virtual Lab Created using Cloud Computing**

Thesis Committee

Name	Signature	Date
Prof. Charles Border Chair		
Prof. Yin Pan Committee Member		
Prof. Arlene Estevez Committee Member		

Thesis Reproduction Permission Form

Rochester Institute of Technology

B. Thomas Golisano College
of
Computing and Information Sciences

Master of Science in
Networking and Systems Administration

Time Delay and its Effect in a Virtual Lab Created using Cloud Computing

I, Juan Nunez, hereby grant permission to the Wallace Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part. Any reproduction must not be for commercial use or profit.

Date: April 2010_ Signature of Author: ___________________________

Abstract

The emergence of Cloud Computing, as a model of virtualized physical resources and virtualized infrastructure, offers the opportunity of outsourcing the implementation of a Virtual Lab Manager. Virtual Lab Management has come to be considered the Holy Grail in the deployment and administration of Labs created in a Virtual Environment. With the advent of Cloud Computing new opportunities are developing that promise to cover much of the future in Virtual Labs. Designing network and information labs with real equipment and tools does not make sense from a cost benefit standpoint, as hardware gets obsolete in a short gap of time, therefore replacing real labs with labs in a Virtual environment this days is a must for teaching in information, security and network classes. Choosing an adequate Virtual Lab Environment solves the problem of creating an adequate academic environment where teachers can serve as effective guides for students which will have a lot of freedom and first hand on experience in the learning subject under consideration. A Virtual Lab Manager in a Cloud Computing environment reduces cost even further, but creates some doubts about the time delays inherent in such a technology. After choosing to use the one created by VMLogix for Amazonaws ec2, it was decided to answer a question in this paper: being Virtual Labs a real time application, how it is affected by time delays and bandwidth when accessed from remote places? The same criteria used for video on demand, voice-over-IP or on line business system as used in networks are going to be applied in the presented work although the much interactivity in a Virtual Lab of any kind .

Categories and Subject Descriptors

General Terms: Management, Monitoring

Keywords: virtual machine, cloud computing, time delay. Network

Table of Contents

Table of Figures

List of Tables

Lists of Lists

Scripts

1. Introduction

The existence and use of Virtual Machines can be traced back as far as 1960 when IBM began to use virtual machines in its mainframes. But explosion in the use of this tool in the last years is not only explained by the fact that they "maximize hardware utilization, decrease hardware costs, reduce power consumption and simplify system management and security" (Virtuautopia), and although each one of these advantages could by themselves justify the need to use Virtual Machines, the explosion in use is really explained by the vast amount of applications that have emerged around this technology

One such application is the Virtual Lab. Such term is understood as an environment created in a distributed network space to access virtual machines and its applications and managed centrally as differentiated by a more general concept that accept as such any results given by electronic means in a different way that those obtained from real physical labs (UNESCO). Virtual Machines technologies allow the virtualization of hardware interfaces to run different guest Virtual Environment in a physical box, providing at the same time security and stability no matter how demanding the applications may be. The use of these technologies for educational purposes is, thus, very promising.

Having personally observed and experienced in other more advanced university how this tool could transform the way people gain access to modern technologies (Border), some of my colleagues and myself took the decision to implement or use a Virtual Lab, as above defined, to be used in courses related to network design, information security and database management and in this process take some observations, design experiments and make conclusions as a thesis that could be used as guide in future developments.

Beginning to use Virtual Machines as an educational tool is not an easy task, but the clear advantages they represent when used in applications as virtual labs, far surpasses all other considerations. To design, develop and implement a virtual lab starting from a null experience requires dedicating a lot of time deciding how to get virtual machines working at the lower possible cost using all available resources at the PUCMM. Once this problem is solved designing the first labs is not so difficult.

Another problem related with developing Virtual Labs is the decision about management. Rigby and Dark could give some inspiring thoughts when it comes to choose a Virtual Lab. Whether to use outsourcing, because in this case the institution concentrates on lab design and less on maintenance, or building remote lab environment “in-house” piggy backing on existing resources and management. Using in this way the existing network and authentication facilities is a possibility in this last mentioned approach.

The final aspect to consider is what Virtual Machine Manager to use. This choice conditions in a great deal the lab manager to select. VMWare is the dominating solution existing today in the market, and was a first choice, but it was found out that installation and management of this product is not the best solution for a first try.

The emergence of the computing cloud in the academic environment came as a fast solution and removes many uncertainties in the development of the Virtual Lab. Getting in contact with Amazon EC2 cloud computing and VMLogix Lab Manager was another great surprise as these products represent a solution that outsources the construction and the management of the Virtual Lab.

To understand the concept of Cloud Computing a definition from Vaquero comes very handy: “Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization". Cloud computing means a lot of computing resources but there are a lot of new problems to explore.

One of the problems encountered with cloud computing is getting instances to start. Lagar et al. put it this way: “Instantiating new VMs is a slow operation (typically taking “minutes” [EC2])". This slow start is a factor to be considered when using these labs. Besides the moment the machine starts it is unaware of any running application state.

Can these delays be acceptable in the long run? It was decided to treat virtual labs in this environment as any real time application, like voice application in a network, and as

such consider the effect of overall time delay in the acceptability of this working environment.

The work to be presented can be summarized with this question: is there any way to measure time delays and bandwidth limits while working in a virtual lab constructed in a cloud computing environment that would make, beyond those limits, impossible to develop normal work ?. Up to what point can be considered time delay and bandwidth limit as user friendly when accessing a virtual lab constructed in the clouds?

One element of this time delay is network delay. Network delay, as a known fact, is composed of propagation, serialization, and queuing delays. Of these components the one that affect most a Virtual Lab is queuing delay, the one depending on network congestion and routers queue length, and it can be reasonably accepted that propagation and serialization speeds are sufficiently large as not to affect access to Virtual labs. Another element to consider, not related to the network delay, is the processing delay giving mainly by the virtual machine performance.

To better study and understand the viability of this work it was decided to analyze how the time delay could affect access in real time the virtual machines. Much effort was concentrated in evaluating real time access to the virtual machines considering as independent variables the existing network delay and the bandwidth to access the virtual lab. It was not taken into consideration other factors, such as "system load, real operating system overhead, real scheduling" (M. D. Canon) and it was concentrated in testing real time access and constraints imposed by time delay and bandwidth limits.

2. Problem Statement

The presence of virtual machines has brought many advantages in educational environments like flexibility, cost reduction and ubiquity, especially if there is access to a Virtual Lab, because by eliminating the distance barrier, it has created a very special real time application in the educational field. But as in all real time applications the uncertainty exists of what QoS (quality of service) will be offered in a network that relies in TCP protocol. But other factors are equally important in the QoS required: bandwidth, load on the servers working on the virtual machines, packet losses, jitter, routing changes, and, very specially, time delay.

Using Virtual Machines in a Cloud Computing environment for educational purposes should not only be seen as a valuable tool, but should also be considered confronted, by its own nature, with different constraints.

As a consequence of bandwidth, virtual machine performance or queue congestion, virtual machine deployment as any real time application can be affected. The aim of this work is to evaluate Quality of Service in virtual machine lab in a Cloud Computing environment, as a function of time delay and bandwidth variations.

When students access such a virtual lab in different places and time of day, under unfavorable conditions, they will be working under different environments and their ability to complete the task at hand could be impaired. An observable fact is that people using internet at different time of the day, or different days in the weak have different level of bandwidth and therefore experiment different levels of time delay.

It was decided to do some research on the effect that this changing environment, will have on satisfaction using Virtual Machines. Time delay was measured in a quantitative way using known monitoring tools in a machine that was to be used as filter between the virtual machines and the machines accessing the latter. Additionally, the bandwidth was controlled with adequate tools

A result of this research is to predict and answer the question: what is the maximum level of time delay and the minimum bandwidth that users would accept to access the virtual lab without any disturbing consequences?

And the practical importance of the answer is given by the fact that any user can determine with simple and easy tools if any disturbance can affect the performance of the virtual lab.

3. Literature Review

Much study has being done on issues related to network delay. Another story is finding papers relating network delay with level of satisfaction in the use Virtual Lab. Needless to say there is no literature to find in issues related to delays in accessing Cloud Computing, except when describing the known and accepted fact that deploying virtual machines takes a great deal of time.

Eli Brosh et al. did some research about the delay performance of TCP because of "the gap between the perceived shortcomings of TCP and its wide adoption in real-world implementations". They considered in their work the two real-time media applications most used in internet: VOIP and video streaming, but they assert that like any other real time application, as Virtual Lab, that works under TCP should be, most of the time delay friendly as those two pinpointed applications. What Eli Brosh et alt consider with respect with VOIP and video streaming can be also be applied to Virtual Labs as application limited – the sending rate is giving by the application and assuming that the underlying network do not have a greedy flow.

Closely related to performance or time delays is the subject of bottlenecks in networks. While bottlenecks are reduced to management and performance overhead the other subject are more general especially when virtualization is involved. In their work Huang, Liu, Abali and Panda take note of these problems and decided to reduce virtualization overhead to get a performance close to the one in the real machine ("A Case for High Performance Computing with Virtual Machines"). The problem with network communication with the virtual machine is not considered in this work as such. It is rather

integrated in the I/O virtualization overhead. They suggest however that using VM to work in a static computing environment do not affect management efficiency. As Virtual Labs are normally created in a static environment it is going to be assumed that few dynamic changes, coming from within virtual labs, are going to affect any measurements made.

No matter how many virtual machines form part of a virtual lab it is expected that bottlenecks are not going to be created by their interactions. Wolinsky et al. discuss the construction of isolated network inside a wide area Overlays of virtual Workstations to reduce overlay in management of wide area distributed computing. This is possible due to the advances in "virtual computing and the revelation that compute-intensive tasks run well on system virtual machines (VMs), the ability to develop, deploy, and manage distributed systems has been ameliorated"("On the Design of Virtual Machine Sandboxes"). Low administrative overheads, and, therefore, high performance, with very low execution time overhead allows the development of such isolated networks.   It will not yet be considered the distribution of computing load between virtual machines. But the idea could be of interest in future development.

Armitage et al. being concerned with the design of Virtual labs to be used in courses of Network Design and while worried by management tasks in Virtual Environment which they solved by the design of simple scripts and   solved the problem of connectivity recurring to light weight free software applications ("Remotely Accessible Sandboxed Environment").   In this work the manageability of task in virtual machines is solved by the Lab Manager, but using simple scripts to measure the variables in the experiments to be made and the importance of free software tools is vital in simulating different situations

The idea of connectivity as exposed by Armitage could be of some importance in this work. If the tools to be used in connecting to the virtual machines do not represent any disturbing factor in the bandwidth, then the only other consideration to take into account for performance should be i/o operations and overhead management.

The erratic behavior of some applications is just a case more to consider the importance of having tools to measure efficiency of applications. Mennon et al. consider the need of developing tools to measure performance within Virtual machines using virtual connections. Their research is concentrated in network applications within a Virtual Machine environment. In their studies they even considered "unforeseen interactions between an application and the VMM and the strange performance anomalies that can happen ("Diagnosing Performance Overheads in the Xen Virtual Machine Environment").

Soror et al. Studied and worked on the configuration of the VM considering how to allocate physical resources to the VM. Through the physical allocation of resources the system will be having a better degree of efficiency as whole and applications will not be longer competing against each other for those resources ("Automatic Virtual Machine Configuration for Database Workloads") .But when the resources is just bandwidth, how will applications compete against each other ? How will the users of the VM stand against the use of the applications of the available bandwidth?

The impact of performance of a VMM working with multiple guests each one with different and concurrent applications and with special intensive processor requirements as well as bandwidth intensive and latency sensitive requirements was considered by Diego Ongaro et al. In a study where they examined those different applications for different VM scheduling they limited their research on the impact of I/O scheduling on the overall performance ("Scheduling I/O in Virtual Machine Monitors").

Aravind Menon et al. state the fact that there few tools to debug performance problems in VM environment. In their work they found situations not easily explained, in particular when the applications under observation were related to network use ("Diagnosing Performance Overheads in the Xen Virtual Machine Environment"). In all cases the interaction with the Virtual Machine Monitor helps explain all the odd behaviors.

Dong-Jae Kang et al. observed the relation between Virtual Machines operated concurrently and I/O requirements and priorities. The fact that the I/O of a virtual

machine is treated in the same way as the I/O of a process of a normal system is a guarantee that proper I/O bandwidth will not be allocated. The overall improvement on average utilization of I/O resources is no guarantee that these same resources are granted in fair way ("Proportional Disk I/O Bandwidth Management for Server Virtualization Environment"). The same reasoning can be brought about network allocation.

The study of high performance network virtualization is analyzed by Guangdeng Liao et al. The advantages of isolation, manageability and resources ownership are weighed against degradation of performance of network intensive applications. They pay special attention to the overhead of network I/O virtualization, the extra driver domain to process I/O requests and the extra scheduler in the virtual machine monitor (MVM) for scheduling domains as they affect performance of network intensive applications ("Software techniques to improve virtualized I/O performance").

The problems related with manageability are also considered by Wei Huang et al., especially for large scale systems, a problem not worth much attention before. Among the manageability efforts maintenance, reconfiguration, fault tolerance, and administration should be included ("Virtual machine aware communication libraries"). The importance of manageability has to be considered as an important requirement necessary to achieve high performance as well as high productivity.

To improve network traffic and, therefore, services qualitatively and quantitatively Qiang Li1 et al. propose to that a monitoring and analysis of it as a requisite when it is desirable no to under-utilize services. They consider that each service in a network has different computational needs and the workload assigned to it will vary and while one service with a peak workload in a given moment may be causing that another service may be sitting idle, but worst there may be no capacity to transfer resources when a service needs them ("VM-based Architecture for Network Monitoring and Analysis").

Irfan Ahmad et alt in analyzing Vmware ESX Server found some amazing discoveries in regard of I/O storage subsystem performance for an architecture, in the host machine, designed for high speed I/O. In particular using direct-attached disk, Raid

arrays and a storage area network (SAN), they found that the behavior of the VM machines closely matches that of the host server very nearly: higher throughput in the host server shows higher throughput in the virtual machines ("An Analysis of Disk Performance in VMware ESX Server Virtual Machines").

4. Methodology

As the goal of this thesis is to establish what values in time delay and bandwidth measured when accessing a virtual lab constructed using the VMLogix Lab Manager are to be considered acceptable from the users viewpoint, creating adequate labs, to be used in experiments in a controlled environment, lead our main effort in the work that is been presented.

Five distinct experiments were designed with the idea to measure, under controlled environment, the independent variables of the work at hand: time delay and bandwidth.

During the development of the experiments many type of tools were used. Most of them have in common their simplicity and their open source origin. Tools are needed to create the controlled environment needed in the experiments, to measure time delay and bandwidth and to process the captured data. In this latter case some of the tools will be simple ruby or shell scripts to process data.

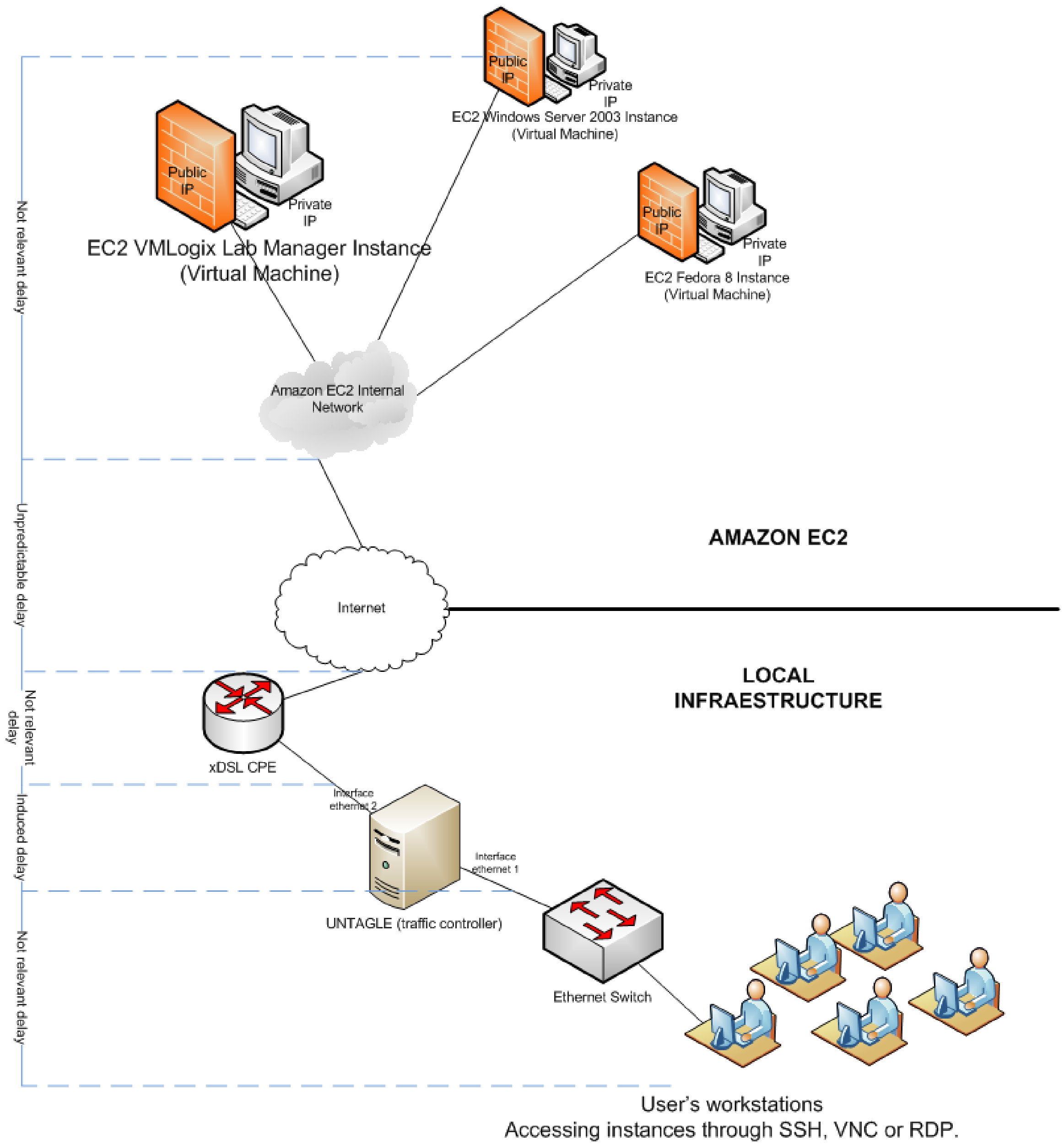
The experiments all needed the use of a special machine that is going to be called the filter or control machine -usually situated close to a router- where accessing the Virtual Lab is affected by manipulating the independent variables to be measured. They all needed one or more test machines from where access to the Virtual Lab was started. Most of the experiments measured the independent variables, time and delay and bandwidth, starting from one or more of the machines.

In the first four experiments the filter machine was an Untangle Machine as this Linux distribution has an easy way to access the tools to be used.

In the first experiment group of five (5) students were the participants, and were chosen by the fact they had previous experience accessing virtual labs. They worked during the

same time period while in the filter machine the dependent variables were manipulated.

Fig 1 Working Environment



In a second experiment real machines and virtual machines were connected to the filter machine. The reason for choosing those machines was due to the fact that the interaction with the virtual lab eliminated some disturbing factors encountered in the first

experiment. The disturbing factors encountered in the first experiment were diverse as discovering virae in some of the machines used as not controlling other users in the same network. It was decided to continue using only virtual machines from the third experiment which was otherwise similar to the second experiment as bandwidth and time delay were made to affect the test machines.

Another aspect considered was the time delay imposed in the packets going through the filter machine. But in this case some discrete values in a wide range were used. The recommendations from the International Telecommunication Union (ITU) for network delays in voice applications were the starting point for any considerations (Varshney, Snow, McGivern and Howard).

The levels of time delays used in the experiments were classified as low, medium and high level time delays. Values up to 200 ms were considered as low level time delays. From 200 ms up to 500 ms were medium time delays. And values from 500ms were considered in the high level range.

In the fourth experiment the Untangle machine was no longer directly connected to the router, but was directly connected to the real machine where time delay was controlled. The Untangle machine was limited to control the test machines bandwidths, while time delay was setup in the router machine.

In the first four experiments the filter machine was always the Untangle Machine. With this machine different level of bandwidth were established for the machines accessing the virtual labs. All the allowed bandwidths were measured as a fraction of the maximum bandwidth of the router.

In the fifth experiment the Untangle machine was no longer used and an Ubuntu machine was taken as test machine and another linux machines was used as controlling machine in the measurements made.

To measure the time delay for the connection from the user's machines to the Virtual Lab Manager a simple script was made to run in the filter machine to affect at regular time

intervals the time delay being set up.

Characteristic of this experiment was the use of scripts that started and changed parameters for the tc command to better control time delay and bandwidth using pre-established values.

Controlled and measurable bandwidths were used during the experiments, from the maximum available in the experiments to lower values, like the ones used in a dialog connection, in a discrete range of values to be considered. While both time delays and bandwidth were used in experiments to observe the effect of both variables.

To guarantee that graphical and text consoles will be accessed and to be tested, an application lab having at least one Window and one Linux machines was chosen. Access to the Windows graphical environment was tested as well as the access to the console of the Linux machine.

5.1 Subjective Variable

The applications chosen were run in all the experiments created, using real networks and virtual local networks, the surveys conducted and measurement taken during the whole experiment were done using those chosen applications. The surveys and the values measures produced qualitative results that provided us feedback of a qualitative kind.

The qualitative feedback was the basis for a subjective and most important variable: the degree of discomfort that users felt under the experiments. The qualitative aspect is defined the moment it is noticed some slowness in access to the virtual machines in the clouds. But, how much slow must the access be to confirm that one or more test persons do not feel at ease working with the labs in all or some of the experiments?

As the first experiment was the only one where some test persons participated, and all of them had previous experience using VM, and their individual perception was a good measure of the discomfort felt when bandwidth and/or time delay were not what they were used to. Changing time delay in the control machine without they being informed

was the main line of action followed to get a picture of the relation between the independent variables and the subjective discomfort value.

A range of values, going from 1 to 10, were established to associate discomfort with a quantitative measure. Values 1 to 5 were associated with fast to less fast, but normal, access to the virtual machines in the clouds. The value 6 and up were mapped to slowness, or impossibility of browsing, or doing any work any work in the virtual machines.

The value 8 was given when one or more test persons felt discomfort using the keyboard in a Windows or Linux Console. Never, it was felt or was not reproduced the environment, the value 10 was used to mark impossibility of using the virtual lab.

An important characteristic of this s-delay variable is related to the fact that it is not related to just one human sense alone, like auditory or visual sense, but the value that the participants can give is also determined by the use of electro-mechanical tools such as the computer keyboard or the mouse. As such the participants can use objective criteria to establish the degree of discomfort they felt.

A weakness in the last four experiments was given not only by the small number of participants, that never were more than three individuals, and besides these were more than professional using the VMLogix Lab Manager when it was time for the last experiment, but also experts in anticipating changes in the environment, although there was not, in no one of the experiment, an established period of time to determine when time delay, or bandwidth was going to be affected.

5. Tools used

To have some quantitative analysis as part of the research, monitoring tools were used to measured time delays and bandwidth. And to make changes in these variables some controlling tools were taken into use.

One simple tool used to check the measured values of both variables time delay and bandwidth will be speed.test available at http://www.speedtest.net. That tool was used

mainly in the controlling machine, but sometimes was also used in the test machines too.

Round-trip time delays were measured using the old and reliable ping tool. To that end a script was created to “ping” a machine in the route to the virtual lab machine, as the virtual machines themselves could not be reached by the icmp protocol. The script “pings” the target machine a fixed amount of time during a time period to be established when the script is called. This script will be run in some test machines and always in the controlling machine.

To establish which one is the target machine to be “pinged” the traceroute command was used having as a parameter the fixed ip address of the Lab Manager.

As mentioned in the description of the experiments, in some cases an Untagle Machine was used as filter node to the router to change bandwidth, in a graphical way, and establish time delays for the test machines

The tool tc (traffic control), an essential part of the iproute2 package of Linux, was essential to configure qdiscs (queue disciplines) and configure packet classifications into qdiscs. This valuable tool is forming part of the later Linux Kernels and can, therefore be found on any Linux distribution. **netem** (network emulator), another tool, is a parameter of tc, was used to establish specific quantitative time delays in one or more machines that communicate using the controlling machine. htb will be the classful qdisc used to control bandwidth. Used together with netem it was possible to control both time delay and bandwidth.

The last tool to be mentioned is the iperf tool which was also used to measure delay and bandwidth between two machines. Iperf is developed by the National Laboratory for Applied Network Research (NLANR). Iperf can also be used to report on jitter and datagram loss.

6. Experiences with VMLogix Lab Manager for Amazonaws ec2

Using Amazonaws together with VMLogix LabManager for the first time was not a difficult task. The learning curve for the administrators is a fast one, although some

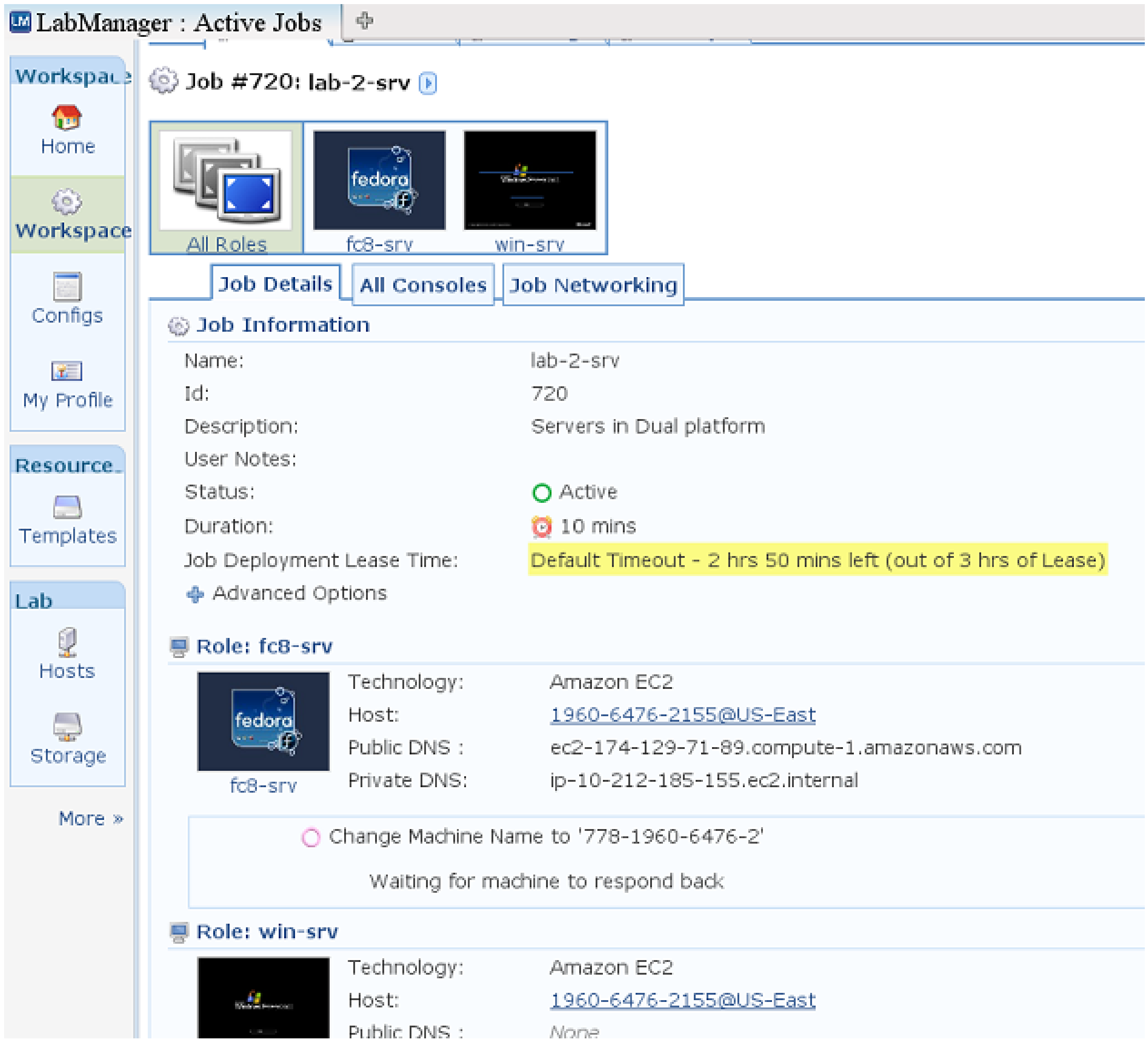
setbacks were experienced. The core of using Amazonaws can be reduced to managing existing templates, importing available ones or creating images to be used as templates. With templates configurations are created and these serve as virtual labs.

VMLogix Lab Manager facilitates the tasks of managing the resources, but other tools can be used such as the “AWS Management Console” and the ec2-tools. These tools should work in a coordinated way, but sometimes some surprises are to be expected. Images created with any of the tools were not ready to be imported and used in configurations. That was, and still is, one of the odd behaviors to be learned about.

The other big surprise came when it was discovered that while the concept of IPZones is used in VMLogix Lab Manager its Cloud Edition cannot use it as AWS uses the concept of EC2 Security Groups, and, therefore, creating isolated virtual machines to be used in security labs is still a task at hand. But creating otherwise complex tasks such as Virtual PBX Stations has shown to be simple routine with the tools at hand.

To do the work at hand, labs were designed to be used by students at PUCMM academic center. To that end users were added to be managed by VMLogix_LM and some normal access to “real virtual” labs was first tried by all parts involved.

FIGURE 2 USING CLOUD COMPUTING

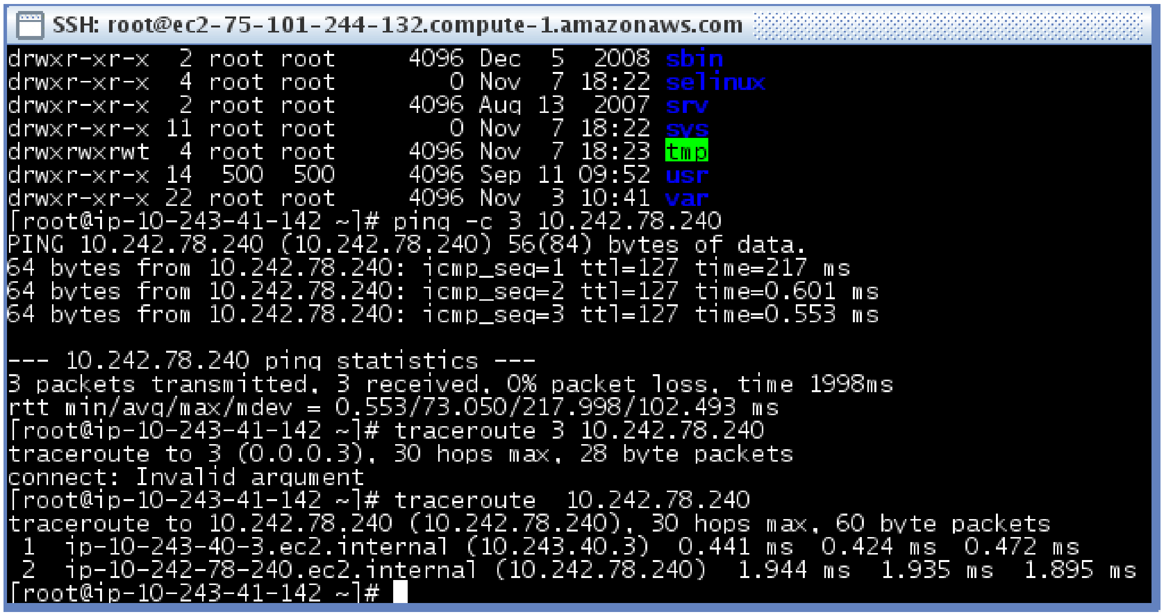


(Time delay is an issue when doing work labs in the cloud. It starts the moment any configuration or image is deployed.)

Two of these labs were used in this proof of concept in more than one experiment. In common they had the integration of two machines: one Linux Machine and one Windows Machine. The former will be the reference when measuring delay friendliness in a console environment, the latter when testing the same aspect in a desktop environment. It

should be pointed out that the virtual machines were not isolated one from another, pinging between them could proved it, but they did not form an unique internal network as figure 2 shows while tracing the network route. Each machine belongs to network and is attached to a router, but the routers can reach each other.

Figure 3 Communication between virtual machines



In other works it can be assumed that there is no difference in delays worth naming in accessing any of the machines in a lab that uses more than one machine.

7. Experiments

It should be pointed out that deploying a configuration is a time consuming task, it can take from 6 to 15 minutes. But there is nothing that can be done about it. Furthermore, the labs that were deployed had a time limit of 3 hours, and it did happen that some of the experiments were interrupted at the end of time interval

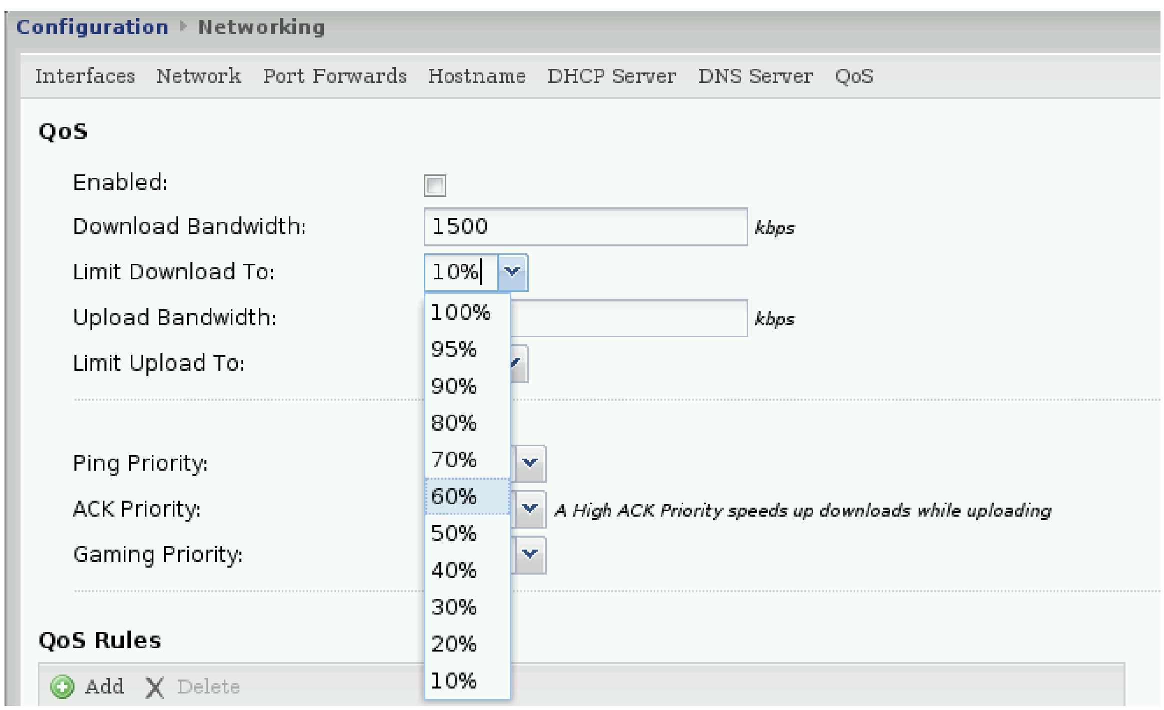
To prove the thesis under discussion several kind of tests, five in all, were conducted. All of them had in common the existence of a machine, the filter or control machine, where

selective changes in bandwidth and time delay affecting all other machines were conducted. The other machines in experiments are going to be called source or test machines. The Virtual Lab was accessed from all the machines but measures were taking in some of them.

As already mentioned, bandwidth and time delays were affected in a controlled manner. In some machines test to measure packet delays to a chosen external machine was conducted in a similar manner to the test done in the control machine to compare the values measured.

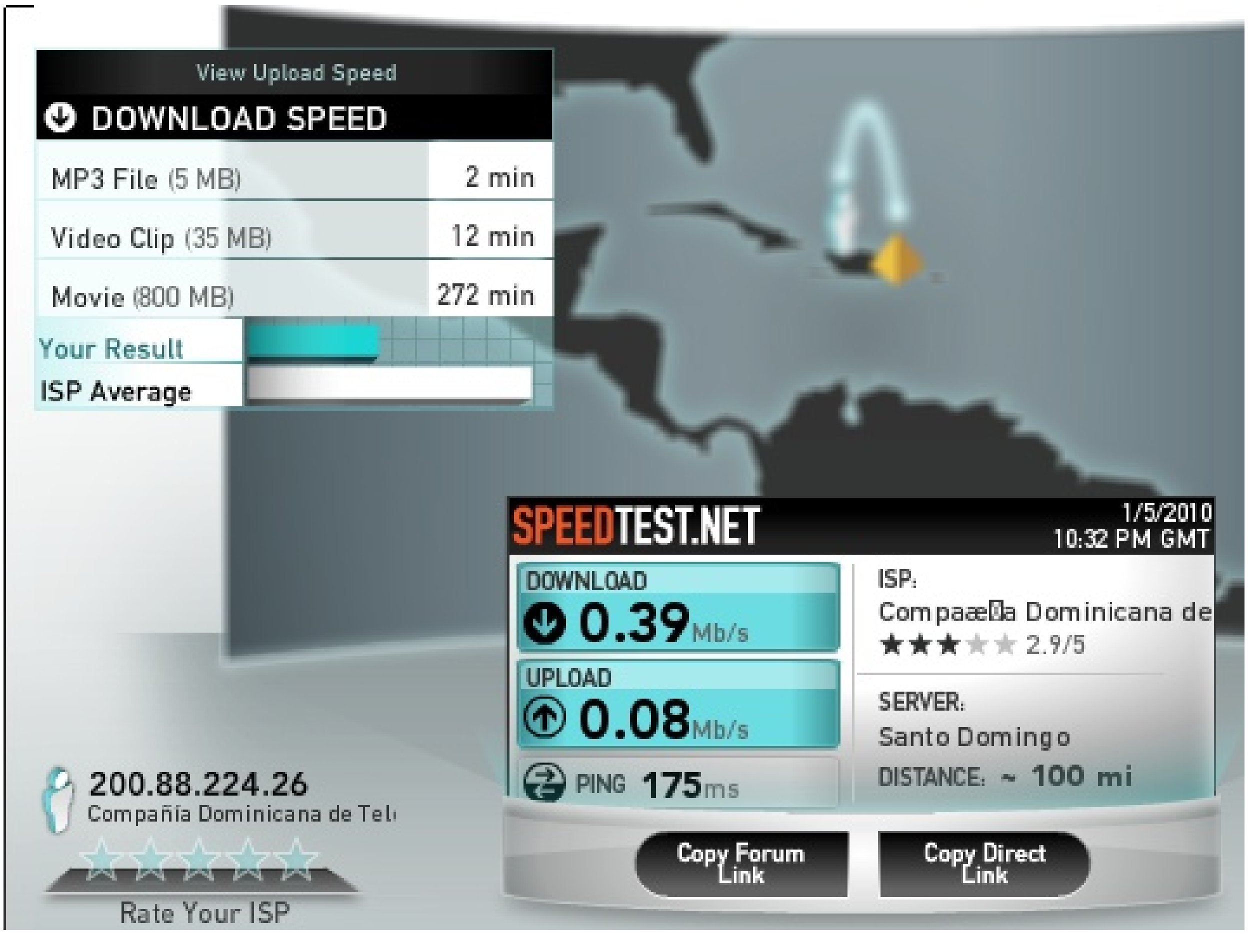
In the first experiment a selected group of 5 students which already have conducted a lab were brought together to a facility at the PUCMM campus. They were to work at the same time with the same lab, using the same network, and were connected through an Untangle machine that was to be used to regulate the QoS for all the participants in random manner. This was the Network Control Experiment.

FIGURE 4 - USING QOS CONTROL



At the same time www.speedtest.net was used to measure the effective bandwidth to the router as well the ping connection time to the same.

FIG 5 MEASURING PING TIME



The second experiment was concentrated on creating controlled delay time between the machine accessing the lab and the virtual machines. To that end a single machine was connected to an Untangle Machine where QOS was controlled and controlled delays were set up. The delays were established using tools presented in all Linux distributions: tc (traffic control) and netem (Network Emulator)

The third experiment was similar to the second one, but in this case only one machine was a “real” one, the other ones were virtual machines. The test machines were two Window machines and one Ubuntu distribution. The real machine was a Fedora 12 machine. Time delays were set on the test machines under stable and fixed bandwidth. Once again www.speedtest.net was used. Untangle was the controlling machine. And tc with netem was used to control time delay.

In the fourth experiment a network of virtual machines was created to be controlled by a single Untangle machine that was connected to the real machine. In this way the control machine served as router to the other machines in the experiment, and at the same time could be verified that the delays affecting the test machines was not affecting the control machine. All the experiments done so far were recreated in this test.

In the fifth and last experiment only two test machines were used, and they were two Linux machines. The control machine was a real Fedora 12 machine. In this experiment no Untangle machine was used as the independent variables, time delay and bandwidth, were going to be controlled using tc together with netem and htb as a classful qdisc. And this tool was used in one of the two linux machines.

In this last experiment was easier to have “blind” machines as part of the methodology used. A blind machine was no affected by any induced disturbance while the participant did not have any knowledge about this reality.

All experiments were designed creating an environment as stable as it could be arranged so that results measured could be compared between them and, at the same time, check if the results are independent of the environment created by the experiment in itself.

7.1 Experiment 1. Network Control Experiment: QoS control

In this first experiment where a selected group of students work in a heterogeneous data center using a common router and a filter machine, an Untangle machine, to control the QoS in the connection to the outside world.

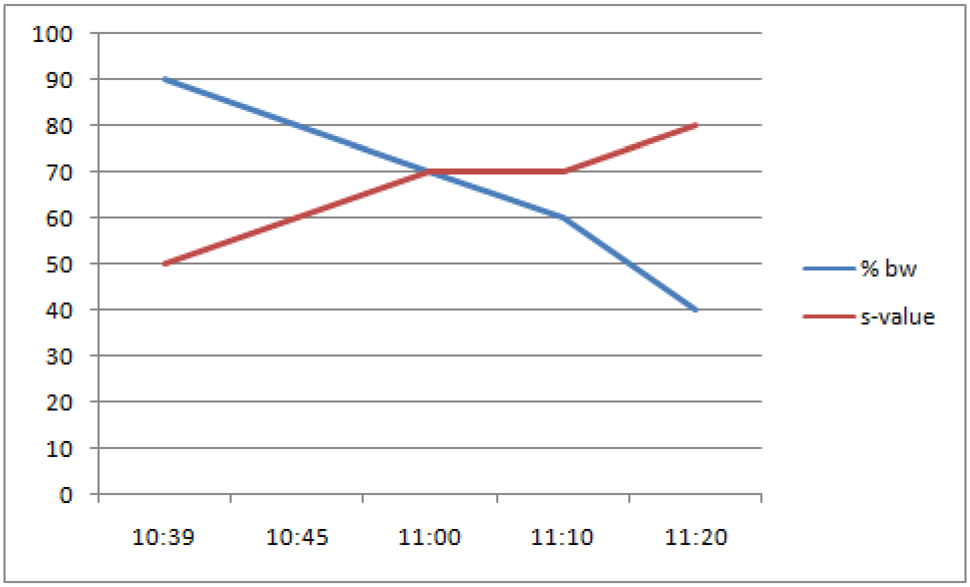
After working for more than one hour and without their knowledge some restrictions were imposed on the participants. Every fifteen minutes at first and later every ten minutes the % of bandwidth was gradually reduced tills the subject started to feel some discomfort. The results are summarized in table 1. This unique experiment was a democratic one as only after all participants, being in part unaware of the bandwidth manipulation, started to complain when the working conditions were unacceptable.

TABLE 1 - BANDWIDTH CONTROL FOR A NETWORK

10/24/09					
% Bw	Bandwidth	% Bw	Bandwidth	ping time	
download	download	upload	upload		s-value
	Mbs		Mbs	ms	
90	0.33	90	0.08	229	5
80	0.19	80	0.07	234	6
70	0.13	70	0.07	222	7
60	0.12	60	0.03	245	7
40	0.07	40	0.03	213	8

This table, is expressed graphically in figure 4 using a scale value of 10 to express the s-salue (subjective value)

Figure 6 Bandwidth versus s-value



All the participants expressed their discomfort when bandwidth was reduced to 60% of the original bandwidth: 512/128 kbps. S-values (subjective values) of 7 and up express the discomfort felt by the participants when they noticed that they should pay more attention to operations as the access was getting slower.

7.2 Experiment 2. Test Machine Experiments: detailed controls

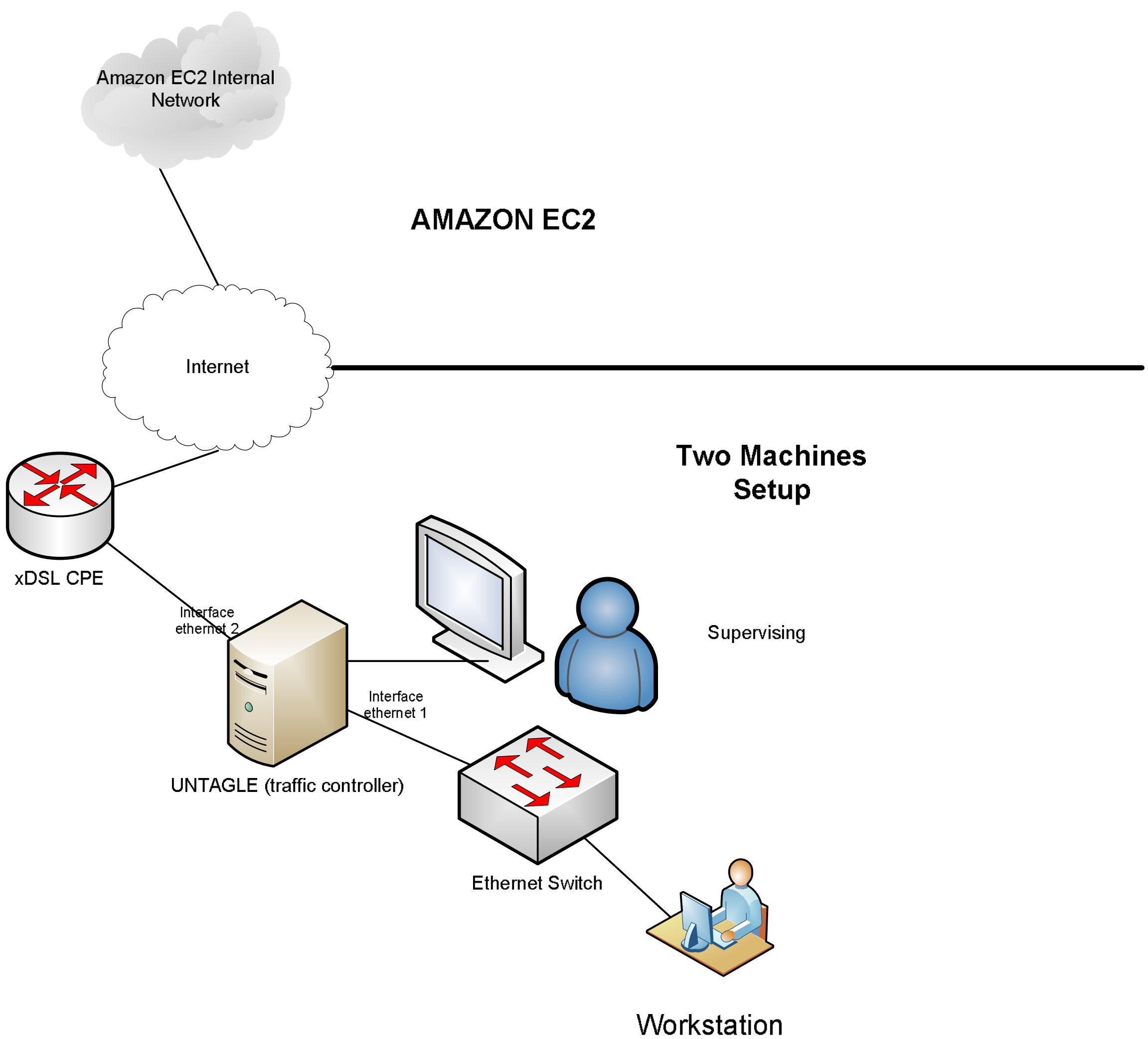
To continue with the experiments and data recollection it was decided to use just one test machine and one control, or filter, machine, which was, once again an Untangle machine. This was the Test Machine Experiment and was divided into five parts:

- Time Delay control experiment (experiment 2a)
- QoS control experiment (experiment 2b)
- Using test.net (experiment 2c)
- QoS and Time Delay Control experiment (experiment 2d)

- Changing QoS values in test machine (experiment 2e)

The test machine has ip address: 10.0.10.40/24 and gateway 10.0.10.55 in the filter. The interface is eth1. The test machine was a Fedora 12 machine.

FIGURE 7. TEST AND FILTER MACHINE



The filter had two ip addresses: 10.0.10.55/24 with interface eth1 and 10.0.0.55/24 with interface eth0 and gateway 10.0.0.1 in the router. Only packets going throughout eth1 are

going to experience delay. The command route -n is self-explaining.

List 1 Using the route command

route -n

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
10.0.0.0	*0.0.0.0*	*255.255.255. 0*	*U*	*1*	*0*	*0*	*eth1*
10.0.10.0	*0.0.0.0*	*255.255.255.0*	*U*	*0*	*0*	*0*	*eth3*
192.168.122.0	*0.0.0.0*	*255.255.255.0*	*U*	*0*	*0*	*0*	*virbr0*

A shell script (script 1) was created to be used in the filter machine and in the source machine to control and measure the difference in time between ping action to a machine close to the Virtual manager. The machine chosen was determined by a traceroute command

Figure 8 Using the Traceroute command

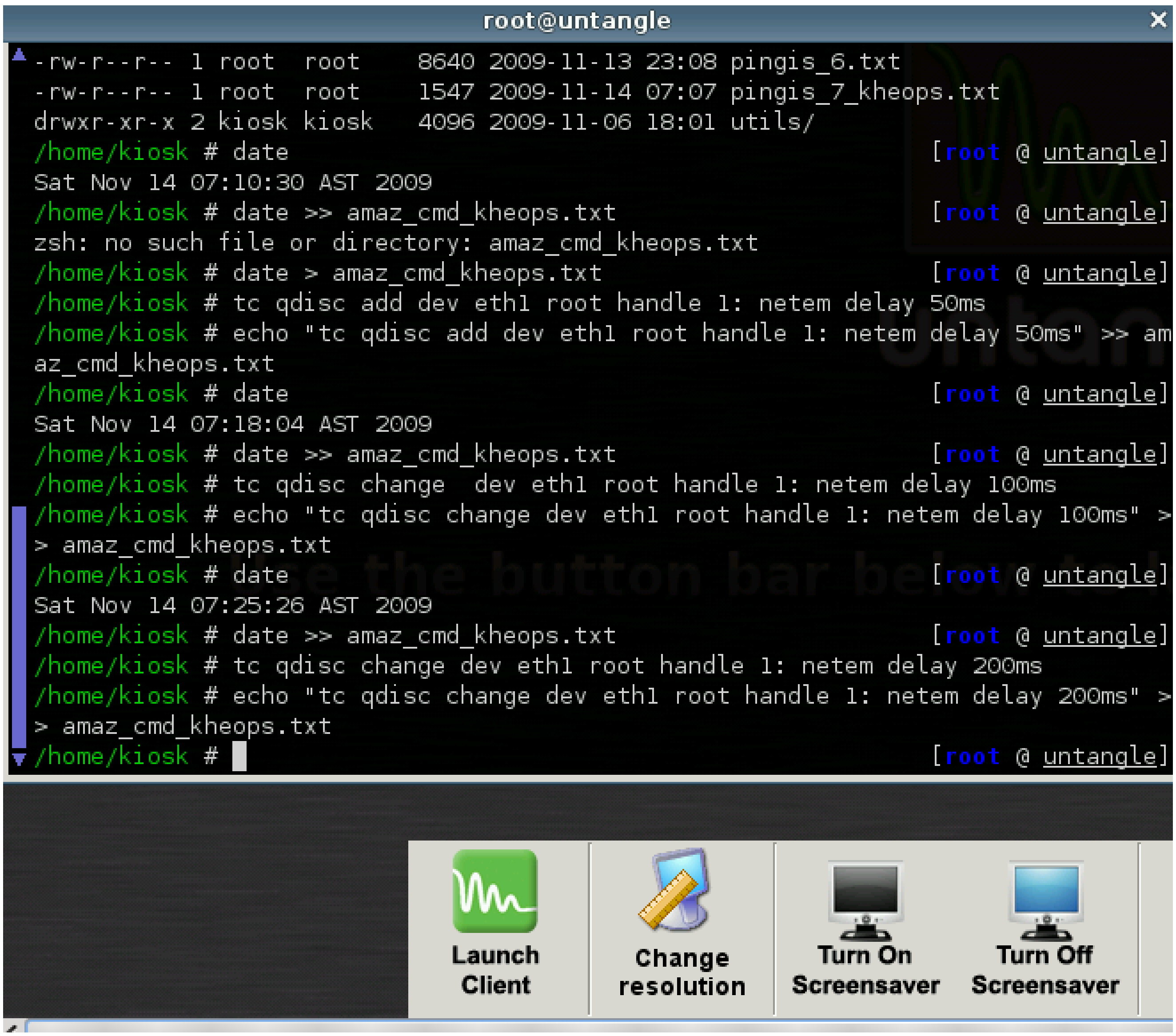
```
27  * * *
28  * * *
29  * * *
30  * * *
[felipe@bizet data]$ traceroute  ec2-174-129-125-107.compute-1.amazonaws.com
traceroute to ec2-174-129-125-107.compute-1.amazonaws.com (174.129.125.107), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)  25.261 ms  24.768 ms  24.283 ms
 2  172.16.32.1 (172.16.32.1)  68.185 ms  75.933 ms  76.565 ms
 3  192.168.102.1 (192.168.102.1)  84.100 ms  88.011 ms  100.010 ms
 4  172.23.24.81 (172.23.24.81)  104.037 ms  107.854 ms  112.051 ms
 5  157.238.179.42 (157.238.179.42)  171.992 ms  180.064 ms  180.713 ms
 6  157.238.179.41 (157.238.179.41)  187.888 ms  167.625 ms  175.350 ms
 7  ae-1.r20.miamfl02.us.bb.gin.ntt.net (129.250.4.161)  108.102 ms  110.299 ms  174.351 ms
 8  as-2.r21.asbnva01.us.bb.gin.ntt.net (129.250.2.184)  206.723 ms  210.258 ms  214.273 ms
 9  * * *
10  * * *
11  72.21.197.32 (72.21.197.32)  223.882 ms  228.029 ms  259.938 ms
12  72.21.222.145 (72.21.222.145)  260.597 ms  261.262 ms  216.397 ms
13  * * *
14  * * *
```

As for the icmp protocol for machines in Amazonaws clouds were blocked the closest machine ip to the target machine was selected as a common reference point for source machine and filter machine. In this example the machine with ip 129.250.2.184 was

chosen.

It was assumed in these experiments that a constant route between source machine and target virtual machine during any given session always exists, or better yet, it will be assumed that the machine close to the target was, during the working session, always the same. Both assertions showed not to hold all the time, but it does not change the validity of the experiment.

FIG 9 THE FILTER MACHINE



The commands **nt** and **netem** were used every certain time period in the filter machine, increasing the delay every time. The time and command used were stored in a text file

7.2.1 Experiment 2a. Test Machine Experiment: time delay control

The goal of this experiment was to establish time delays over the packets going out of the test machine. Time delays were set up over a time period of half an hour and values included were100, 250 and 500 ms. To that end the tc command was the tool to make the changes and time was registered together with the command and delay parameters. List 3 shows what was done.

List 2 Time delay control for test machine (2a)

Sat Nov 7 11:50:18 AST 2009	tc qdisc add dev eth1 root handle 1: netem delay 100ms
Sat Nov 7 12:07:00 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 250ms
Sat Nov 7 12:17:43 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 500ms
Sat Nov 7 12:22:13 AST 2009	tc qdisc del dev eth1 root handle 1:

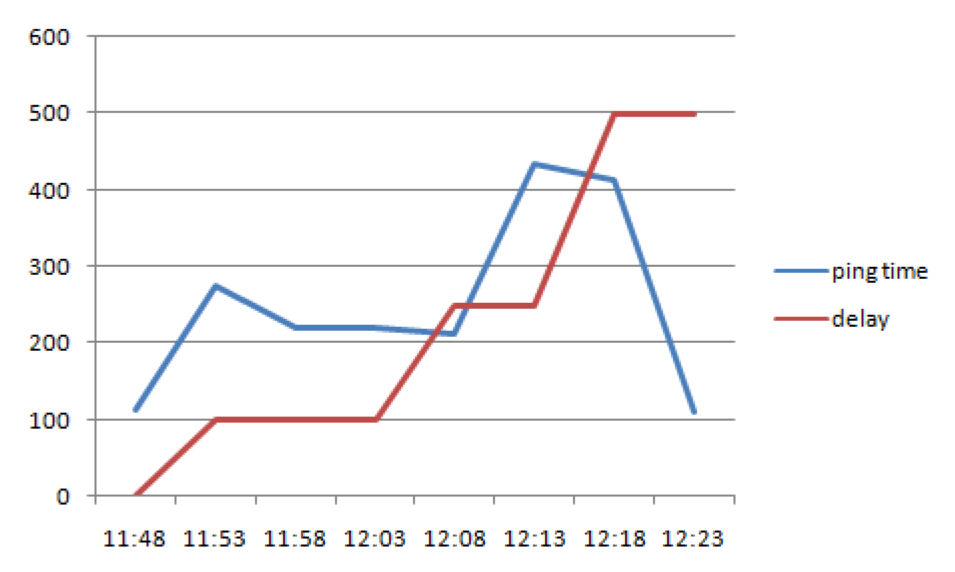
At the same time the script exe_ping.sh was used to capture round trip delays in the test and the filter machine, and table 2 was constructed with the data captured. The media for the values captured is the product of the script dat_proc.rb. Although the expected round trip delay time for the source (or test) machine did not increase the expected values the experiment was conducted to an end without any interruption.

Table 2 Measuring time delay control

Test	Time	ip address	ping time	delay	flt	time	ip address	ping time
			ms	ms				ms
Src	11:48	4.68.17.144	113.10	0	flt	11:47	4.68.17.144	110.35
Src	11:53	4.68.17.144	275.43	100	flt	11:52	4.68.17.144	112.86
Src	11:58	4.68.17.144	221.38	100	flt	11:57	4.68.17.144	133.70
Src	12:03	4.68.17.144	219.19	100	flt	12:02	4.68.17.144	204.67
Src	12:08	4.68.17.144	211.41	250	flt	12:07	4.68.17.144	162.80
Src	12:13	4.68.17.144	434.78	250	flt	12:12	4.68.17.144	114.34
Src	12:18	4.68.17.144	414.27	500	flt	12:17	4.68.17.144	128.63
Src	12:23	4.68.17.144	110.50	500	flt	12:22	4.68.17.144	227.65

After the 250ms delay it was noticeable the effect in graphical interfaces and with the greater delay value (500 ms) the effect was noticeable in text consoles.

Figure 10. Effect of delay in ping time



7.2.2 Experiment 2b Test Machine Experiment: Fixed upload control (95%)

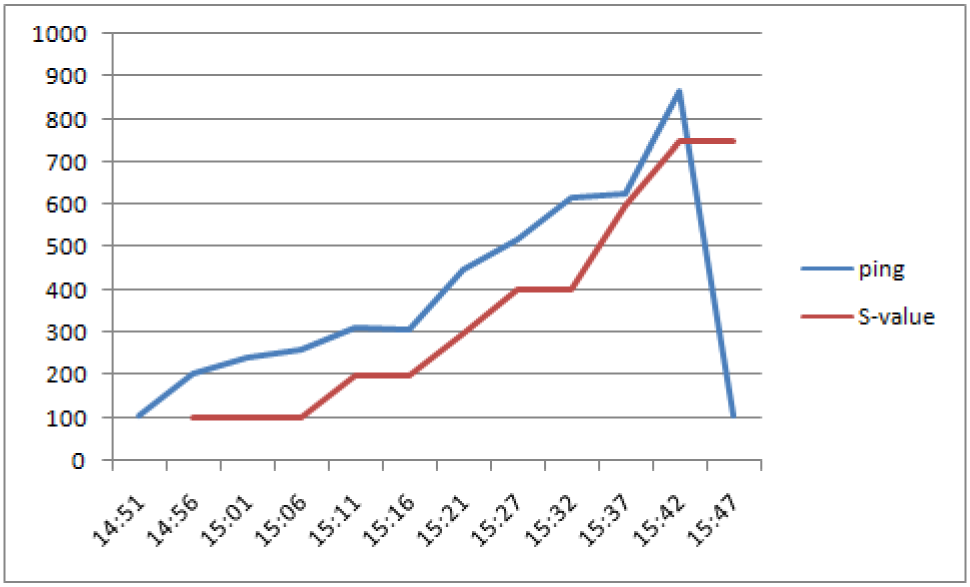
This second test in the second experiment is similar to the first one as the same tools there, especially tc and netem were used to induce time delays changing from 100 ms up to 750 ms under almost an hour time period. The time delays induced for the test machine are detailed in list 4. And like in test 1 in this experiment with two machines time delays were recorder in table 3 and processed under the same time period using the same tools.

7.2.3 Experiment 2c. Test Machine Experiment: Using test.net

This third test is similar to the first experiment especially as the tool test.net found inhttp://ww.speedtest.net was used, but this time we have only Untangle as the filter – or control -machine and the test machine. Additionally capture and processing of round trip delays in both machines using the scripts exe_ping.sh and dat_proc.rb. Table 4 was thus produced under a ninety time period. And table 5 was constructed with the data produced by test.net under the same time period.

Mixing the tables 4 and 5 the figure 11 is produced to show the linear relation between discomfort values (s-values) and the delays measured in the test machine. A scale value of 20 was used represent the almost linear relation between discomfort and delay. It is worth mentioning that the bandwidth in the router is only 1500/384 Kbps, a little less than in the first experiment.

Figure 11 Discomfort versus delay time



7.2.4 Experiment 2d Fixed bandwidth control, variable time delay.

But when there was an equal control imposed on both download and upload bandwidth, there were some differences when accessing the virtual lab. First a fixed bandwidth limitation to 60% download and 95% upload time delays in a range from 20ms to 500ms were set up. The list of tc and parameters used is reflected in list 5.

To table 6 constructed with round trip delay data was added the time delay induced. The effect in round trip delay for the test machine is noticeable but does not follow a clear linear relation as expected in a disturb free environment.

Once again with the data produced by test.net under the same time period table 7 was constructed. It was observed, this time, that discomfort was felt after an induced 200 ms time delay.

7.2.5 Experiment 2e Changing QoS values in test machine

In this test no changes were made in time delays for the test machine. The test was limited to changing bandwidth. Table 8 present time delays captured and processed under an eighty minutes time period. And table 9 is the data captured using test.net

Discomfort were noticed for any bandwidth, both in graphical interfaces and text consoles, after bandwidth was reduced in 70% or more in upload and download equally. But those differences could be explained by network own congestion at some of the moments during the time this experiment was taking place. There was a little difference of less than 7 ms between ping time obtained for test and filter machine

7.3 Experiment 3 QoS and time delay control

This phase in the experiments is a continuation of experiment 2 and consisted in taking fixed values for QoS in each test and using tc and netem tools to change time delays for the test machine.

The experiment was divided in four parts.

- Fixed 40% QoS (test 3a)
- Fixed 60% QoS (test 3b)
- Fixed 50% QoS (test 3c)

In the first test bandwidth was fixed in 40% of the download bandwidth and 40% for upload bandwidth. The first of the experiments suggested that 60% of the download bandwidth was a critical value as was observed in the first part of the experiment. Delay times changed in the filter machine from 50 ms to 750 ms.

In the second test bandwidth was fixed in 60% of the download bandwidth and the same percentage for both download and download bandwidth. Delay times changed in the filter machine from 50 ms to 700 ms.

In the third test bandwidth was fixed in 50% of the download bandwidth and the same percentage for both download and download bandwidth. Delay times changed in the filter machine from 100 ms to 800 ms.

7.3.1 Experiment 3a QoS Control: 40% bandwidth control

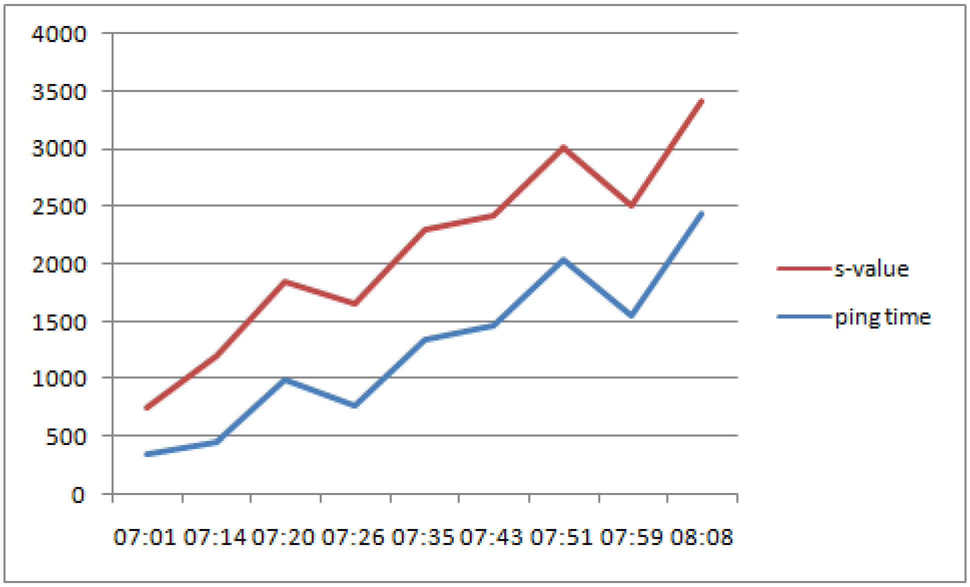
Time delay was controlled using the tools **tc** and **netem**. The time table of their used was registered and notes were taken whenever some change in access to the virtual machines was noticeable. Once again from 200ms and up some discomfort was noticeable, and from 500ms the effect in text consoles was equally important.

In this first test a fixed bandwidth limitation to 40% download and 40% upload time delays in a range from 50ms to 700ms were used as reflected in list 6. With the bash shell script comparative round trip delays were taken from both the test machine and the filter machine every 5 minutes and for every time delay. (Table 10)

For 100 ms time delays and up set up in the filter machine noticeable discomfort was affecting work in the Virtual Machines (especially when graphical environment was present).Using www.speedtest.netTable 11 was created and the discomfort is associated here with a bandwidth of 0.17 MBps.

For a scale factor of 1000 for the s-value figure 12 was drawn to illustrate the relation, once again, between time delay measured in the router and discomfort using the s-values.

Figure 12 Discomfort versus ping time delay 40% bandwidth



But this ping time delay is a function of the delay time induced in the test machine.

7.3.2 Experiment 3b QoS Control: 60% bandwidth control

In a second experiment a fixed bandwidth limitation to 60% download and 60% upload time delays in a range from 100ms to 700ms were set up as reflected in list 7.

Once again using the bash shell script comparative round trip delays were taken from both the test machine and the filter machine every 5 minutes and for every time delay and Table 12 was thus generated. Table 13 takes in to account results from www.speedtest.net and the delay times from table 12. Discomfort can be felt for time delay of 200ms and up, and for minimum bandwidth of 0.47MBps.

7.3.3 Experiment 3c QoS Control: 50% bandwidth control

In a third experiment a fixed bandwidth limitation to 50% download and 50% upload time delays in a range from 50ms to 700ms were set up. Using the same tools were created similar tables. List 8 is made with the time delay variations made under an hour time period.

Under the same time interval time delays were measured and the results are shown in table 14. And finally the test.net was used to construct table 15 with the values measured. Discomfort can be felt for time delay of 100ms and up, and for minimum bandwidth of 0.49MBps.

7.4 Experiment 4 Controlling the controller

The experiments in this period were conducted using , once again a network, but this time of virtual machines, and one real machine, and its aim was to establish the same filters used in the first experiments. The Untangle machine, the filter machine, was the one to control bandwidth, but this time was not directly connected to the router, but was connected instead to the real machine. The Untangle machine was controlled in time delays by the real machine, acting as simple router, and with it all other virtual machines.

The experiment was divided in three parts:

i. bandwidth control

ii. 40% bandwidth and time delay control

iii. 60% bandwidth and time delay control

iv. time delay control

In the first part, where the only variable affected was bandwidth, using the filter machine for the machines "behind" it bandwidth was reduced from a 100% value to a minimum of 10% of the incoming and outgoing data under a two hours period.

In the second test, the router machine was used to establish delays for packets coming in and out of the test machine in the network while a 40% of the bandwidth was established for it. Time delays ranging from 50 ms up to 700 ms were used.

In the third test the router machine was used to establish delays for packets coming in and out of the test machine in the network while a 60% of the bandwidth was established

for it. Time delays ranging from 50 ms up to 700 ms were used.

In the fourth test with no changes in bandwidth access was established under changing time delay conditions spanning values from 100 ms up to 800 ms.

7.4.1 Experiment 4a Bandwidth Control

With the help of www.speedtest.net table 16 with values measured in the routing machine was constructed while access test were made. Although no major changes were noticed in the table, except, for the bandwidth changes made in the filter machine, the discomfort noticed was also written down.

When bandwidth for the tested machines was 60% of the original 0.5 Mbps, or below that value some discomfort could be clearly felt. Time delay was measured and the result is what table 17 shows.

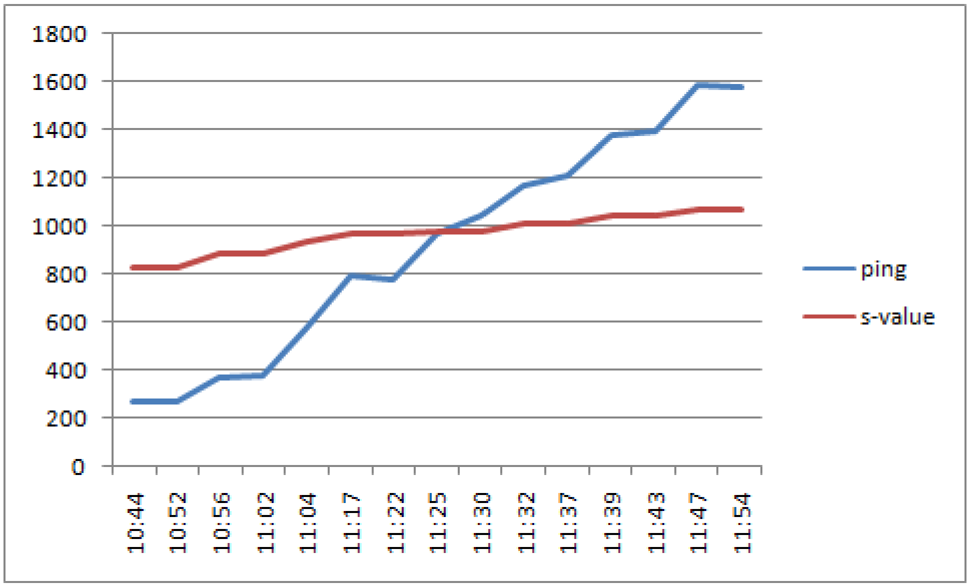
7.4.2 Experiment 4b 40% bandwidth and time delay control

Under an hour time period changes were made in the real machine, using tc and netem from a minimum time delay of 50 ms to a maximum of 700 ms having bandwidth operating at 40% of the router bandwidth. The commands used to induce time delay changes are reported in list 9.

The result from measuring external bandwidth and time delays is exposed in table 18, where it can be observed that ping time changed from the normal 165ms to a value increased with the double of the delay imposed. While data obtained from test.net was used to construct table 19.

All the experiment result can be summarized in figure 13 where discomfort (s-value with a scale value of 110 was used.

Figure 13 S-values versus ping time for experiment 4b



This part of the experiment was repeated as discomfort accessing the virtual machines was felt as soon values greater than 100 ms were used to delay of packets send to and from the test machines.

7.4.3 Experiment 4c 60% bandwidth and time delay control

In this part of the experiment a bandwidth constraint of 60% of the router access was imposed in the test machines (05 mbp/0.375mbp) with time delays variations from 0 to 800 ms as reported in table 20.

Using tc at chosen time point, time delays were imposed in incoming and outgoing packets as is reported in list 10, while table 21 reports measurements made in round trip delay. There was no change in results obtained as compared with the other first parts of this experiment.

7.4.4 Experiment 4d Time delay control

In this test using the same network configuration, no changes were made in bandwidth, but just in time delays that went from 100 ms up to 600 ms. A simple command was executed at variables time points under some 80 minutes time interval to make changes in delay times. A table with the commands used and the time when they were executed is presented in list 11.

With no bandwidth restrictions, once again could be verified, in table 22, that ping time increased with the double of the value of the delay imposed. Table 23 was the result of measuring ping time using test.net under the test.

The level of discomfort measured was the same felt when delays over 100 ms were created. The moment 300 ms time delay was imposed there was some discomfort when accessing Windows and Linux consoles.

7.5 Experiment 5 Numeric tools control.

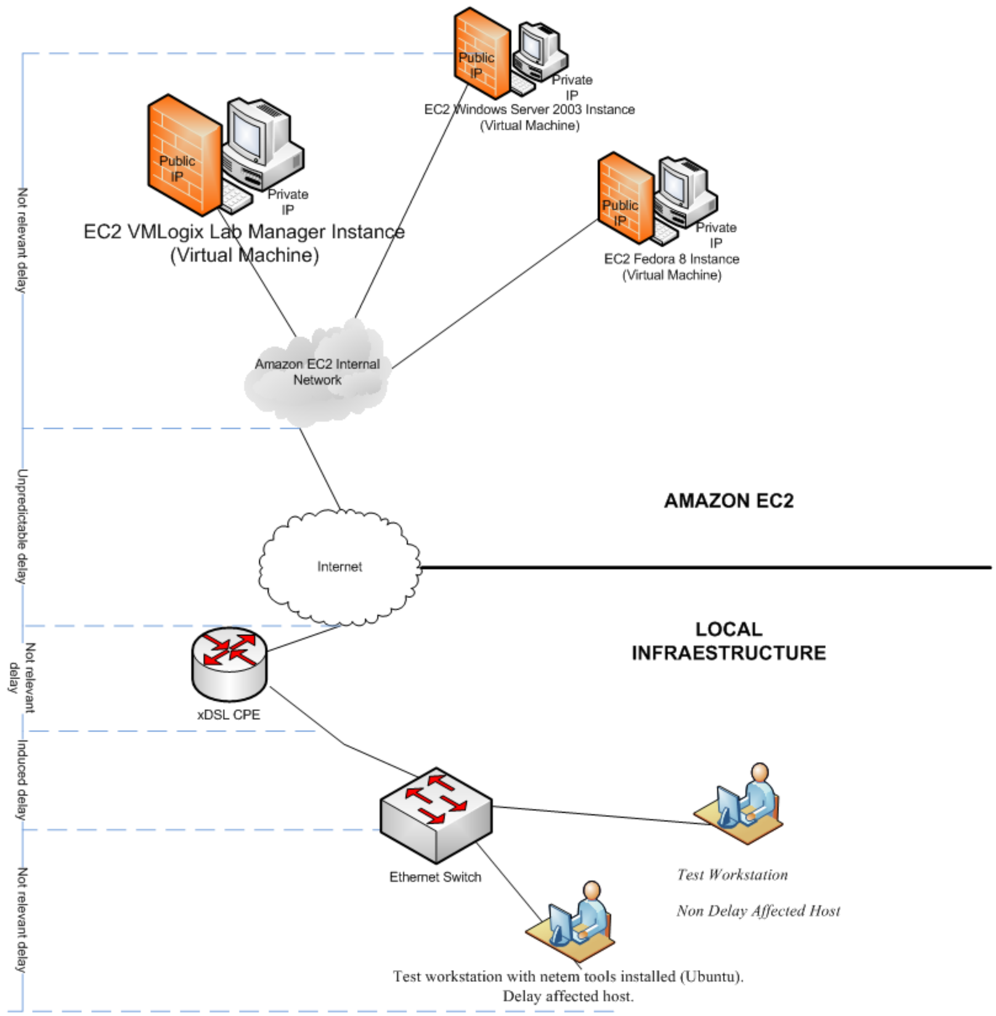
The last of the experiments executed was a variation of the other ones, but this time use of simple scripts that create the environment in the execution of netem is the fundamental part and there was no recourse to Untangle machine.

Five scripts were created to initialize, to induce change in time delay, to induce change in bandwidth, to induce change in both time delay and bandwidth and to destroy a queue discipline.

The fundamental script was the one that initialized, and created the root for the queue discipline and established interface to use, initial delay and bandwidth to use: tc_ini.sh (script 3). The second script was the one to allow change in time delay for packets in a queue discipline: tc_delay.sh (script 4). The third script was created to change bandwidth in a queue discipline: tc_bw.sh (script 5). The fourth t script was the one that controlled

changes in both time delay and bandwidth: tc_both.sh (script 6). And finally was the script that eliminated any of the queue disciplines created: tc_destroy.sh (script 7).

Figure 14 Numeric Tools Experiment



This experiment was developed in two stages. In the first part only changes in time delays were considered. In the second part changes in time delays and bandwidths were affected

7.5.1 Experiment 5a Time delay Control

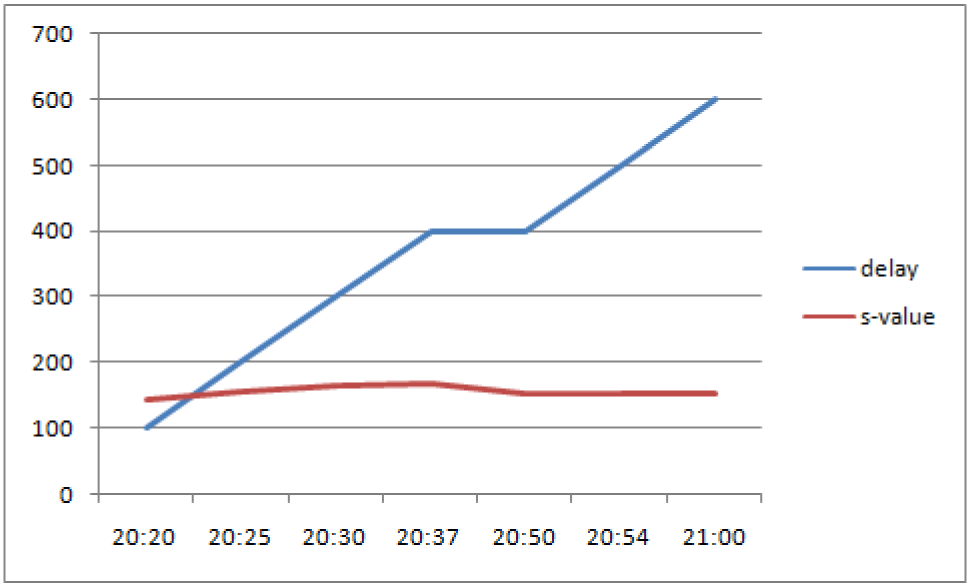
As one done in other experiments with values collected by executing www.speedtest.net table 24 was build to present changes in time delays obtained when there were changes in in/out network traffic.

Twice under the experiment changes in the target machine for the delay imposed were made, as external congestion changes in network were reflected in route changes, and this is reflected in list 12 of commands used. As list 12 shows, time delays were changed from 100 ms up to 700 ms under almost an hour time period. Bandwidth for packets going out from the test machine was constant.

As in all other experiments the media in time delays given by the ping command executed every 5 minutes were collected in the source and the filter machine (table 25 and table 26).

Using a scale value of 20 for s-values figure 14 was drawn from the s-values taken from table 24 and the delays were the ones on list 12.

Figure 15 Experiment 5a: discomfort versus time delay



In this experiment discomfort was experienced the moment the access to the virtual machines is subjected to an additional 100 ms time delay, but some time during the practice it felt better to work under a time delay of 500 ms rather than with a time delay of 400 ms especially when there changes in the network path. Network congestion could be the source of this anomaly.

7.5.2 Experiment 5b Time delay and bandwidth Control

As a result of this two hours experiment the usual list (list 13) for collecting time delays was created taking into consideration the time delay input as well as then bandwidth caused by executing different scripts. In the other case changes were made to affect the test machine. The initial command was: sh script/tc_ini.sh eth1 100 256 "10.0.10.65". Table 27 reports round trip delays measured, while, how and when the scripts were used during the experiment is told in the following list 14.

In one case changes were made in the filter machine. And was usual in the other experiments the ping tool was executed in both machines every five minutes to measure the round trip time delay between the target machine, close the lab manager machine, and either of the before mentioned machines (table 28)

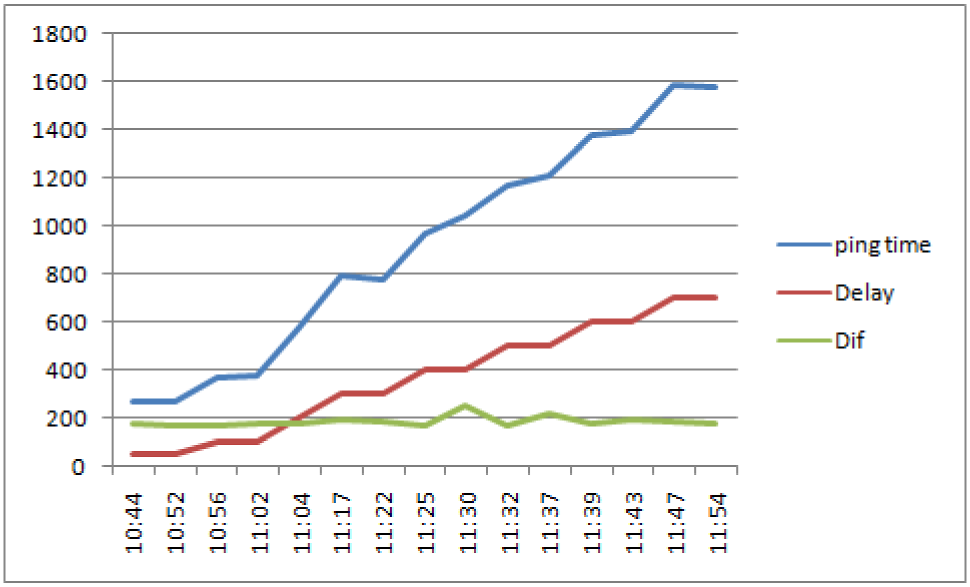
8. Analysis and Results

As stated from the beginning, the work presented in this report is about how a real time application in classical internet architecture does not secure that it "will receive the QoS required in order to function properly" (Erol Gelenbe). Nevertheless the fact that the increase in real time applications goes on, it is left for users of such services to understand and know the limitations that can affect the use of the new technologies and what level of time delay and bandwidth can be acceptable when using the new tools.

In all the experiments executed there were objective quantitative and qualitative data to consider as well as subjective discomfort and its relation to the former values. In an ideal experiment none of the values to be measured should have being affected for other factor than the ones we were introducing in the experiments. But such was not, always, the case. Although it is known the "effect on network performance due to degradation of traffic variability and as a result of routing changes (Himabindu Pucha), it was decided to ignore those effects as they were not affecting in a permanent and consistent way our hypothesis and measurements.

Taking for example table 19 in experiment 4a where time delays were induced in a test machine from values ranging from 50 ms to 700 ms, the effect on the measured ping time was constant as given by dif = (ping time) – 2 * delay, for a value close to 180 ms.

FIGURE 16 TIME DELAY IN TEST MACHINE



What in the report is called pinging time, round-trip time, was produced from two distinct sources: from http://www.speedtest.net and from the shell script (script 1). An though the values thus obtained, when not affected by induced time delays, as was always the case with the filter machine, should be close to each other, there were some differences that can be explained not only by network congestion but by the methodologies used. The shell script was used to produce results every 5 minutes with only 5 samples taken.

When time delay was an independent variable in any of the experiments executed values produced by the tool tc (traffic control) of 200 ms (d1) and up always produced some (subjective) discomfort when working in graphical environment. Values from 500 ms (d2) and up produced discomfort when working in a text console environment.

As pinging time, from the router to a machine close to the Lab Manager, was observed to change from 105ms (p1) to 280ms (p2), it can be stated that a one way delay time of 372ms (t1) to the Lab Manager will produce discomfort in a graphical environment , and 672 ms (t2) will produce discomfort in a text console environment. The values (t1, t2) mentioned are the result of: t1 = (p1 + p2)/2 + d1 and similar for text console

Although results are well defined when working with time delays, there is no clear border when working with bandwidth reduction control. Taking into consideration download bandwidth the network practice fixed an upper limit of 0.13MBps and with the test machine experiment the upper limit can be fixed in 0.43MBps

When both QoS and time delays were considered values of download bandwidth ranging from 0.42 MBps to 0.50 MBps for induced time delays of 400 ms to 200 ms were the ones producing discomfort to access the Lab Manager.

9 Summary

As expected, time delay does affect the use of a virtual lab in the clouds. Acceptable time delay values for web access to virtual labs in the clouds using VMLogix Lab Manager can clearly be established in a range going up to 260 ms for configurations using graphical interfaces

Assuming a natural time delay from the source machine used to the closest one to the virtual machine of 60ms (half the time obtained from a normal ping to that machine) it can be asserted that after 260ms time delay an user is going to have some trouble working with virtual labs in a graphical environment. And equally important trouble is going to be had when the delays are over 560ms in a text console environment.

There were some assumptions made in the experiments: there was no important or any change in the networking route from the test machine to the Virtual LabManager. But it happened,

The second experiment, and the other that followed, were made using just a test machine and a filter machine as it gives better and more controlled results. It is recommended to synchronizing application of time delays with the measurement of results given by the script used. In this way a more fine grained measurement can be made.

It is felt that anomalies given by peak time delays having its origin in random network congestion should have being taken into account but it was there was no methodology or

plan made to examine this phenomenon. So, some future works going through a more detailed and in time more fine-grained experiments are in place as well as considering not only the protocols ssh and rdp, as was the case in this work, but the other used protocols (vnc, nx) to access Virtual Machines through VMLogix Lab Manager.

A more precise relation between discomfort as dependent and subjective variable as related to both time delay and bandwidth could not be reached, so more detailed work has to be done in this area too.

10 References

Ahmed A. Soror, Umar Farooq Minhas,Ashraf Aboulnaga, Kenneth Salem, Peter Kokosielis and Sunil Kamath. "Automatic Virtual Machine Configuration for Database Workloads." Transactions on Database Systems. New York, NY, USA: ACM, 2008.

Aravind Menon, Jose Renato Santos, Yoshio Turner, G. (John) Janakiraman and Willy Zwaenepoe. "Diagnosing Performance Overheads in the Xen Virtual Machine Environment." ACM/Usenix International Conference On Virtual Execution Environments. New York, NY, USA: ACM, 2005.

Border, Charles. The Development and Deployment of a Multi-User, Remote Access . Covington, Kentucky, USA, 7 March 2007.

Dark, Steven Rigby and Melissa. "Designing a Flexible, Multipurpose Remote Lab for the IT Curriculum." Proceedings of the 7th conference on Information technology education SIGITE'06. New York, NY, USA: ACM, 2006.

David Isaac Wolinsky, Abhishek Agrawal, P. Oscar Boykin, Justin R. Davis, Arijit Ganguly, Vladimir Paramygin, Y. Peter Sheng and Renato J. Figueiredo. "On the Design of Virtual Machine Sandboxes for Distributed Computing in Wide-area Overlays of Virtual Workstations." Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing. Washington, DC, USA: IEEE Computer Societ, 2006.

Diego Ongaro, Alan L. Cox and Scott Rixner. "Scheduling I/O in Virtual Machine Monitors." ACM/Usenix International Conference On Virtual Execution Environments. New York, NY, USA: ACM, 2008.

Dong-Jae Kang, Chei-Yol Kim, Kang-Ho Kim and Sung-In Jung. "Bandwidth Management for Server Virtualization Environment." Proceedings of the 2008 International Conference on Computer Science and Information Technology. IEEE Computer Society, 2008.

Eli Brosh, Salman Abdul Baset, Dan Rubenstein and Henning Schulzrinne. "The Delay-Friendliness of TCP." ACM SIGMETRICS international conference on Measurement and modeling of computer systems. New York, NY, USA: ACM, 2008.

Erol Gelenbe, Georgia Sakellari and Maurizio D'arienzo. "Admission of QoS aware users in a smart network." ACM Transactions on Autonomous and Adaptive Systems (TAAS). New York, NY, USA: ACM, 2008.

Guangdeng Liao, Danhua Guo, Laxmi Bhuyan and Steve R King. "Software techniques to improve virtualized I/O performance on multi-core systems." Symposium On Architecture For Networking And Communications Systems. New York, NY, USA: ACM, 2008.

H. Andrés Lagar-Cavilla, Joseph A. Whitney, Adin Scannell, Philip Patchin, Stephen M. Rumble, Eyal de Lara, Michael Brudno and M. Satyanarayanan. "SnowFlock: Rapid Virtual Machine Cloning for Cloud Computing." Proceedings of the 4th ACM European conference on Computer systems. ew York, NY, USA: ACM, 2009.

Himabindu Pucha, Ying Zhang, Z.Morley Mao and Chalie Hu. "Understanding network delay changes caused by routing events." Joint International Conference on Measurement and Modeling of Computer Systems. New York, NY, USA: ACM, 2007. 73 - 84 .

Irfan Ahmad, Jennifer M. Anderson, Anne M. Holler, Rajit Kambo and Vikram Makhija. " An Analysis of Disk Performance in VMware ESX Server Virtual Machines." New York, NY, USA: ACM, 2006.

Luis M. Vaquero, Luis Rodero-Merino , Juan Caceres and Maik Lindner. "A Break in the Clouds: Towards a Cloud Definition." ACM SIGCOMM Computer Communication Review (2008).

M. D. Canon, D. H. Fritz, J. H. Howard, T. D. Howell and M. F. Mitoma. "A Virtual Machine Emulator for Performance Evaluation." Proceedings of the seventh ACM symposium on Operating systems principles . New York, NY, USA: ACM, 1979.

Qiang Li, Qinfen Hao, Limin Xiao and Zhoujun Li. "VM-based Architecture for Network Monitoring and Analysis." Proceedings of the 2008 The 9th International Conference for Young Computer Scientists. IEEE Computer Society, 2008.

UNESCOVirtual Laboratory Toolkit, Why Virtual Laboratories?http://virtuallab.tu-freiberg.de/

Upkar Varshney, Andy Snow, Matt McGivern and Christi Howard. Voice over IP. New York, NY, USA: ACM, 2002.

VirtuautopiaAbout_Xen_Virtualization_Essentialshttp://www.virtuatopia.com/index.php/About_Xen_Virtualization_Essentials

Wei Huang, Jiuxing Liu, Bulent Abali and Dhabaleswar K. Panda. "A Case for High Performance Computing with Virtual Machines." Proceedings of the 20th annual international conference on Supercomputing. New York, NY, USA: ACM, 2006.

Wei Huang, Matthew J. Koop, Qi Gao and Dhabaleswar K. Panda. "Virtual machine aware communication libraries for high performance computing." Conference on High Performance Networking and Computing. New York, NY, USA: ACM, New York, NY, USA.

William D. Armitage, Alessio Gaspar and Matthew Rideout. "Remotely Accessible Sandboxed Environment with Application to a Laboratory Course in Networking." Proceedings of the 8th ACM SIGITE conference on Information technology education. New York, NY, USA: ACM, 2007.

11 Appendices

11.1 Accessing Cloud computing: image gallery

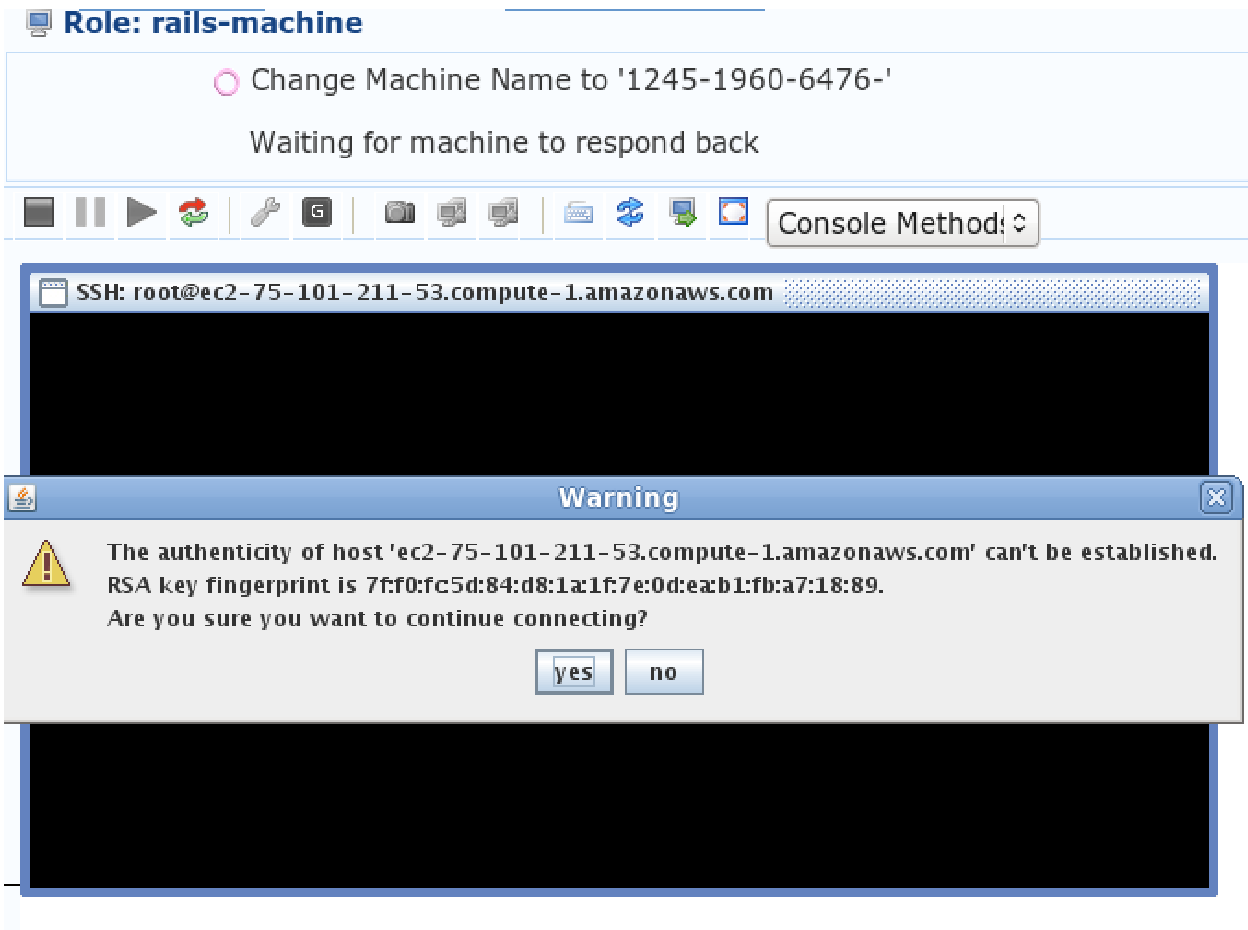


FIG 17 AUTHENTICATING FOR LINUX MACHINE

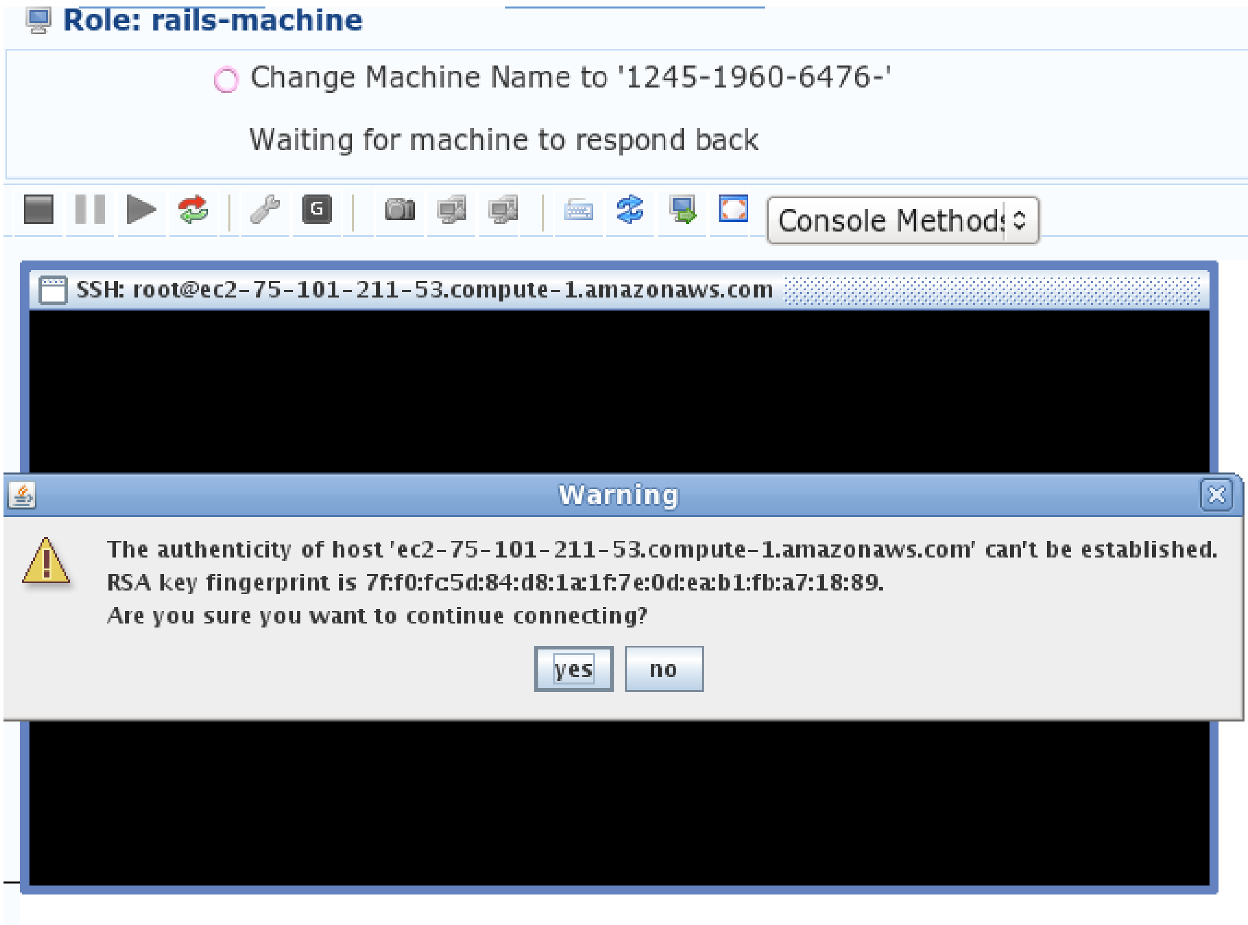


FIGURE 18 DEPLOYING THE LINUX CONSOLE

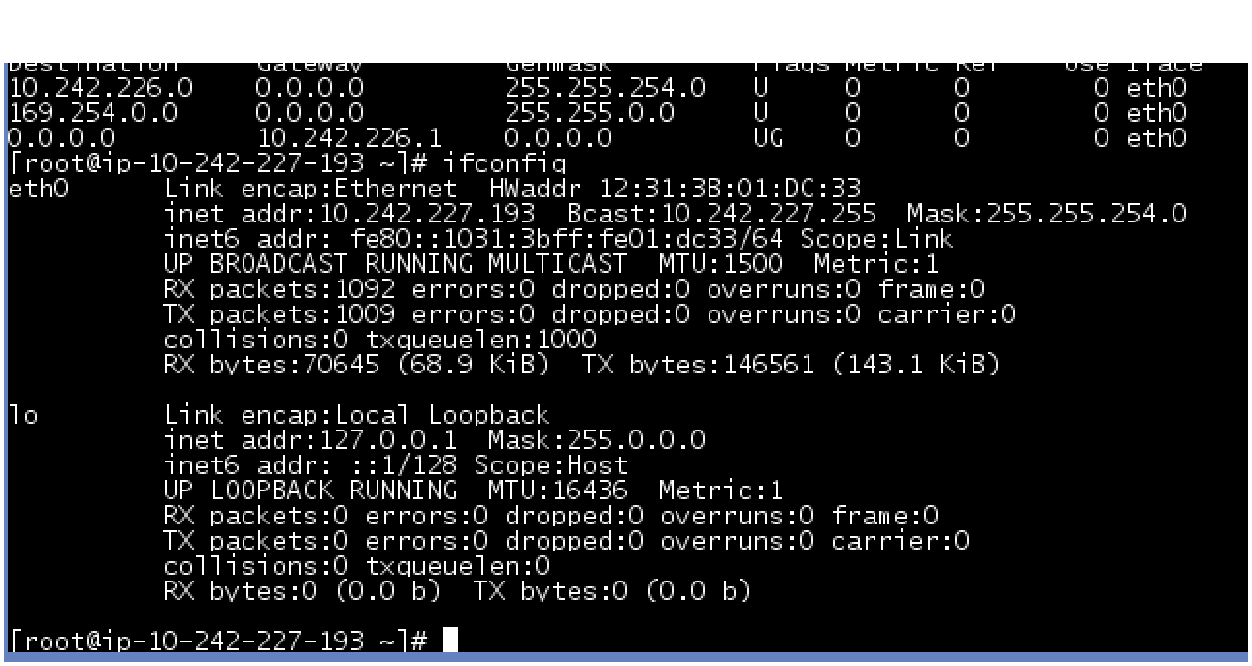
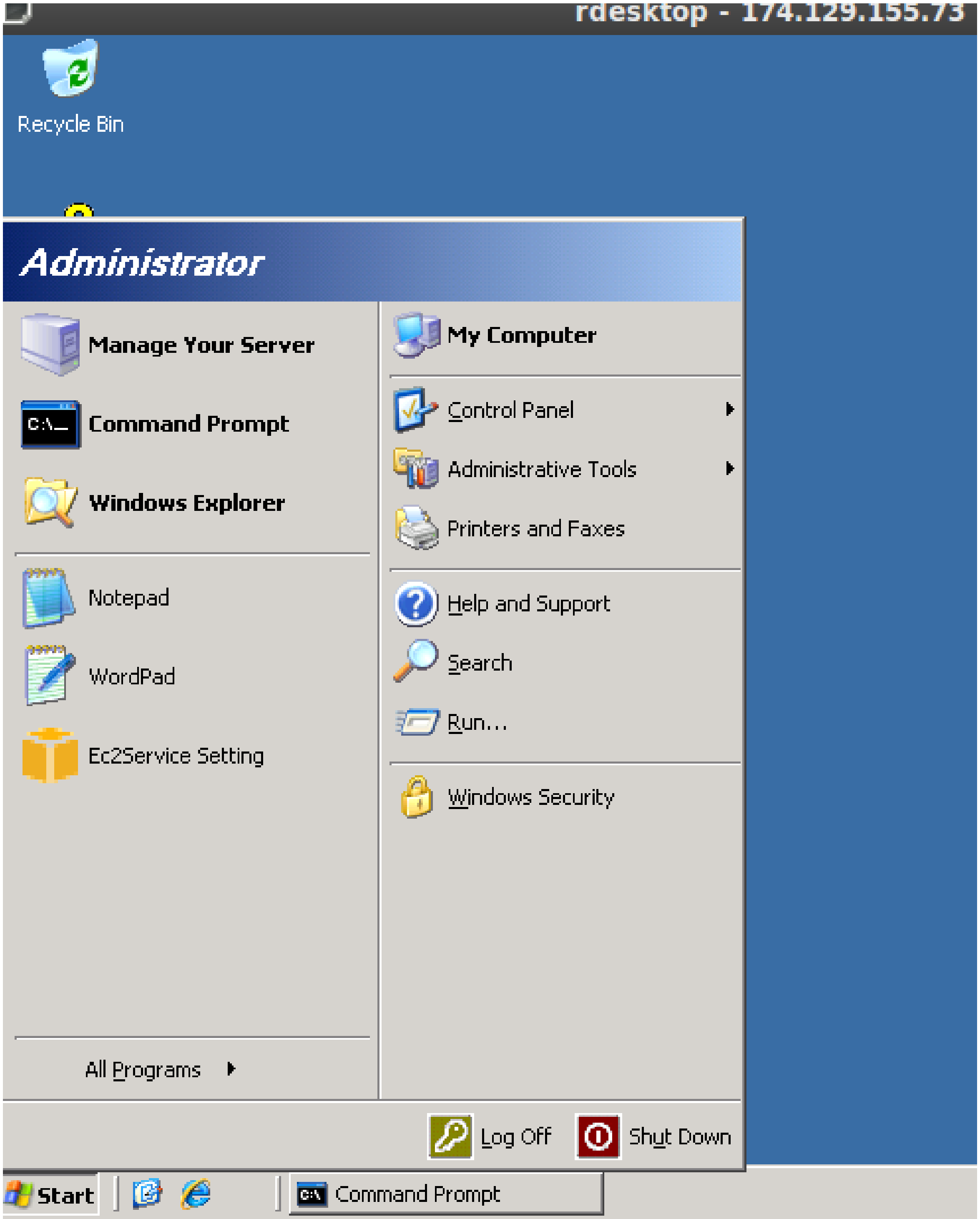


FIG 19 DEPLOYING A WINDOWS MACHINE



11.2 Scripts

Script 1 exe_ping.sh: Capturing round trip delay data

```
#! /bin/bash

# In this script round trip delay to a given target is measured

# There are four parameters:

#    $1 --> target

#    $2 --> file to save output from script

#    $3 --> number of ping counts

#    $4 --> sleep time in minutes

if [ $# -lt "4" ]

then

echo 2>&1 "Missing arguments "

exit 1

fi

while true

do

date>> "$2"

ping -c $3    "$1"    >>    "$2"

sleep "$4"m

done
```

Script 2 dat_proc.rb: Processing round trip delay data

```
#! /usr/bin/env ruby

# ruby script used to process data from the files

#    where round trip delays - from ping action - were saved

#    Only lines with media results are considered

fuen = ARGV.size > 0 ? ARGV[0] : "flt"

reg1 = /(\d\d:\d\d:\d\d)/

reg2 =/PING\s+(\d+\.\d+\.\d+\.\d+)/

reg3 = /\/(\d+\.\d+)\//

st = "#{fuen}|"

while lin = $stdin.gets

st += $1 if lin =~ reg1

st += "|" + $1 + "|" if lin =~ reg2

if    lin =~ reg3

st += $1

puts st

st = "#{fuen}|"

end

end
```

Script 3 tc_ini.sh Initializing a queue discipline

```
#! /bin/bash

# script to control delay and bandwidth using tc

# (traffic control) with netem to control delay and

# htb to control bandwidth cleaning up qdisc

# creating root qdisc

if [ $# -lt 4 ]

then

echo "Use $0 interface delay bandwith source"

exit 1

fi

inter=$1      # interface used

inidel=$2    # initial delay

inibw=$3    # initial bandwidth

src=$4      #    source    ip

tc qdisc del dev $inter root

tc qdisc add dev $inter handle 1: root htb

tc class add dev $inter parent 1: classid 1:1 htb rate ${inibw}Kbps burst 12Kb

tc qdisc add dev $inter parent 1:1 handle 10: netem delay ${inidel}ms

tc filter add dev $inter parent 1: protocol ip prio 1 u32 match ip src ${src}/32 flowid 1:1
```

Script 4 tc_delay.sh : inducing delay in a queue discpiline

```
#! /bin/bash

# script to change delay

# using tc (traffic control) with

# netem to control delay

# htb to control bandwidth

if [ $# -lt 2 ]

then

echo "Use $0 inter delay "

exit 1

fi

inter=$1      # interface used

inidel=$2    # new delay value

tc qdisc change dev $inter parent 1:1 handle 10: netem delay ${inidel}ms
```

Script 5 tc_bw.sh: inducing bandwidth change in a queue discipline

```
#! /bin/bash

# script to change    bandwidth

# using tc (traffic control)

# netem to control delay and

# htb to control bandwidth

if [ $# -lt 2 ]

then

echo "Use $0 interface bandwith "

exit 1

fi

inter=$1    # interface used

inibw=$2    # new    bandwidth value

tc class add dev $inter parent 1: classid 1:1 htb rate ${inibw}Kbps burst 12Kb
```

Script 6 tc_both.sh: inducing bandwidth and time delay change in a queue discipline

```
#! /bin/bash

# script to change delay and bandwidth

# using tc (traffic control) with

# netem to control delay and

# htb to control bandwidth

# traffic discipline is already created

if [ $# -lt 3 ]

then

echo "Use $0 interface delay bandwith "

exit 1

fi

inter=$1    # interface used

inidel=$2    # new delay

inibw=$3    #      new bandwidth

tc class change dev $inter parent 1: classid 1:1 htb rate ${inibw}Kbps burst 12Kb

tc qdisc change dev $inter parent 1:1 handle 10: netem delay ${inidel}ms
```

Script 7 tc_destroy.sh: Eliminating a queue discipline

```
#! /bin/bash

# script to remove the root qdisc

if [ $# -lt 1 ]

then

echo "Use $0 interface "

exit 1

fi

inter=$1      # interface used

tc qdisc del dev $inter root
```

11.3 Tables and list in Experiments

Experiment 2b

List 3 Time delay control for test machine (2b)

Sat Nov 7 14:55:17 AST 2009	*tc qdisc add dev eth1 root handle 1: netem delay 100ms*
Sat Nov 7 15:08:38 AST 2009	*tc qdisc change dev eth1 root handle 1: netem delay 200ms*
Sat Nov 7 15:16:49 AST 2009	*tc qdisc change dev eth1 root handle 1: netem delay 300ms*
Sat Nov 7 15:26:28 AST 2009	*tc qdisc change dev eth1 root handle 1: netem delay 400ms*
Sat Nov 7 15:31:42 AST 2009	*tc qdisc change dev eth1 root handle 1: netem delay 500ms*
Sat Nov 7 15:38:56 AST 2009	*tc qdisc change dev eth1 root handle 1: netem delay 600ms*
Sat Nov 7 15:41:40 AST 2009	*tc qdisc change dev eth1 root handle 1: netem delay 750ms*

Table 3 Measuring time delay: details

test	time	ip address	ping time	Delay	flt	time	ip address	ping time
			ms	Ms				Ms
src	14:51	129.250.2.184	107.0		flt	14:52	129.250.2.184	108.1
src	14:56	129.250.2.184	206.8	100	flt	14:56	129.250.2.184	107.9
src	15:01	129.250.2.184	242.1		flt	15:01	129.250.2.184	108.9
src	15:06	129.250.2.184	262.7		flt	15:06	129.250.2.184	157.2
src	15:11	129.250.2.184	312.9	200	flt	15:11	129.250.2.184	108.2
src	15:16	129.250.2.184	307.7		flt	15:16	129.250.2.184	211.6
src	15:21	129.250.2.184	449.8	300	flt	15:21	129.250.2.184	106.4
src	15:27	129.250.2.184	517.7	400	flt	15:26	129.250.2.184	108.2
src	15:32	129.250.2.184	616.1		flt	15:31	129.250.2.184	105.0
src	15:37	129.250.2.184	623.2	600	flt	15:36	129.250.2.184	106.8
src	15:42	129.250.2.184	863.6	750	flt	15:41	129.250.2.184	104.4
src	15:47	129.250.2.184	106.3		flt	15:46	129.250.2.184	105.3

Experiment 2c

Table 4 Measuring time delay: changing bandwidth fixed upload

src	time	ip addres	ping time	delay	flt	time	ip addres	ping time	s-value
			ms	ms				ms	
src	14:51	129.250.2.184	107		flt	14:52	129.250.2.184	108	5
src	14:56	129.250.2.184	206.8	100	flt	14:56	129.250.2.184	108	6
src	15:01	129.250.2.184	242.1	100	flt	15:01	129.250.2.184	109	6
src	15:06	129.250.2.184	262.7	100	flt	15:06	129.250.2.184	157	6
src	15:11	129.250.2.184	312.9	200	flt	15:11	129.250.2.184	108	7
src	15:16	129.250.2.184	307.7	200	flt	15:16	129.250.2.184	212	7
src	15:21	129.250.2.184	449.8	300	flt	15:21	129.250.2.184	106	8
src	15:27	129.250.2.184	517.7	400	flt	15:26	129.250.2.184	108	8
src	15:32	129.250.2.184	616.1	400	flt	15:31	129.250.2.184	105	8
src	15:37	129.250.2.184	623.2	600	flt	15:36	129.250.2.184	107	8
src	15:42	129.250.2.184	863.6	750	flt	15:41	129.250.2.184	104	9
src	15:47	129.250.2.184	106.3	750	flt	15:46	129.250.2.184	105	9

Table 5 QoS Control for test machine fixed upload

11/07/2009						
	% Bw	Bandwidth	% Bw	Bandwidth	ping time	S-value
Time		Download		upload		
		mbs		mbs	ms	
18:50	80	0.53	95	0.13	169	6
19:10	70	0.49	95	0.13	168	6
19:20	60	0.56	95	0.11	173	7
19:25	50	0.4	95	0.12	167	7
19:55	40	0.4	95	0.11	170	8
20:05	30	0.4	95	0.06	168	9
20:10	20	0.42	95	0.13	167	8
20:15	10	0.4	95	0.13	171	8

Experiment 2d

List 4 Fixed QoS with time delay control

Sat Nov 7 20:42:43 AST 2009	tc qdisc add dev eth1 root handle 1: netem delay 20ms
Sat Nov 7 20:44:09 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 100ms
Sat Nov 7 20:47:39 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 200ms
Sat Nov 7 20:56:22 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 300ms
Sat Nov 7 21:06:51 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 400ms
Sat Nov 7 21:13:11 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 500ms

Table 6 Measuring time delays (60%/95% bandwidth)

test	time	ip address	ping time	delay	flt	time	ip address	ping time
			ms	ms				ms
src	20:40	129.250.2.184	503.8		flt	20:39	129.250.2.184	124.2
src	20:45	129.250.2.184	231.8	20	flt	20:44	129.250.2.184	123.2
src	20:50	129.250.2.184	321.9	200	flt	20:49	129.250.2.184	109.4
src	20:55	129.250.2.184	539.4		flt	20:54	129.250.2.184	125.5
src	21:00	129.250.2.184	421.8	300	flt	20:59	129.250.2.184	131.5
src	21:05	129.250.2.184	434.9	400	flt	21:04	129.250.2.184	125.7
src	21:10	129.250.2.184	534.6		flt	21:09	129.250.2.184	123.6
src	21:15	129.250.2.184	634.4	500	flt	21:14	129.250.2.184	589.3
src	21:20	129.250.2.184	1246.5		flt	21:19	129.250.2.184	122.1
src	21:25	129.250.2.184	133.4		flt	21:24	129.250.2.184	121.5

Table 7 Fixed QoS Control for test machine *(60%/95% bandwidth)*

11/07/2009							
	% Bw	Bw	% Bw	Bw	ping time	delay	s-value
Time	Download	download	Upload	Upload			
		Mbs		Mbs	Ms	ms	
20:43	60	0.4	95	0.05	203	20	5
20:50	60	0.53	95	0.11	365	100	5
20:54	60	0.39	95	0.11	568	200	6
20:58	60	0.5	95	0.05	767	300	7
21:05	60	0.38	95	0.06	965	400	8
21:13	60	0.4	95	0.09	1166	500	9
21:20	60	0.34	95	0.07	1665	750	9

Experiment 2e

Table 8 Measuring time delay in QoS Control

test	Time	IP address	Ping time			Time		Ping time
			Ms		Flt		IP address	ms
src	21:52	129.250.2.184	116.2		Flt	21:52	129.250.2.184	124.0
src	21:57	129.250.2.184	108.2		Flt	21:57	129.250.2.184	124.8
src	22:02	129.250.2.184	117.7		Flt	22:02	129.250.2.184	116.3
src	22:07	129.250.2.184	170.0		Flt	22:07	129.250.2.184	321.9
src	22:12	129.250.2.184	122.6		Flt	22:12	129.250.2.184	109.7
src	22:17	129.250.2.184	692.7		Flt	22:17	129.250.2.184	576.3
src	22:22	129.250.2.184	110.3		Flt	22:22	129.250.2.184	110.8
src	22:27	129.250.2.184	180.4		Flt	22:27	129.250.2.184	192.7
src	22:33	129.250.2.184	118.7		Flt	22:32	129.250.2.184	117.8
src	22:38	129.250.2.184	110.1		Flt	22:37	129.250.2.184	163.9
src	22:43	129.250.2.184	229.5		Flt	22:42	129.250.2.184	431.2
src	22:48	129.250.2.184	107.1		Flt	22:47	129.250.2.184	114.1
src	22:53	129.250.2.184	110.5		Flt	22:53	129.250.2.184	116.4
src	22:58	129.250.2.184	334.3		Flt	22:58	129.250.2.184	144.5
src	23:03	129.250.2.184	116.4		Flt	23:03	129.250.2.184	117.7
src	23:08	129.250.2.184	108.0		Flt	23:08	129.250.2.184	106.8

Table 9 Roundtrip values in QoS Control for test machine

11/13/09					
	% Bandwidth	Bandwidth	% Bandwidth	Bandwidth	ping time
Time	Download	Download	Upload	upload	
		Mbs		Mbs	Ms
21:54	100	0.51	100	0.12	283
22:10	90	0.51	90	0.13	282
22:17	80	0.39	80	0.12	281
22:27	70	0.43	70	0.13	283
22::36	60	0.45	60	0.12	281
22:42	50	0.44	50	0.12	278
22:48	40	0.51	40	0.12	282
22:54	30	0.44	30	0.09	283
23:00	20	0.54	20	0.07	282
23:06	10	0.26	10	0.03	281

Experiment 3a

List 6 QoS and time delay control (40% bandwidth)

Sat Nov 14 07:10:54 AST 2009	*tc qdisc add dev eth1 root handle 1: netem delay 50ms*
Sat Nov 14 07:18:13 AST 2009	*tc qdisc change dev eth1 root handle 1: netem delay 100ms*
Sat Nov 14 07:25:38 AST 2009	*tc qdisc change dev eth1 root handle 1: netem delay 200ms*
Sat Nov 14 07:34:33 AST 2009	*tc qdisc change dev eth1 root handle 1: netem delay 300ms*
Sat Nov 14 07:43:11 AST 2009	*tc qdisc change dev eth1 root handle 1: netem delay 400ms*
Sat Nov 14 07:50:06 AST 2009	*tc qdisc change dev eth1 root handle 1: netem delay 500ms*
Sat Nov 14 07:59:59 AST 2009	*tc qdisc change dev eth1 root handle 1: netem delay 600ms*
Sat Nov 14 08:08:34 AST 2009	*tc qdisc change dev eth1 root handle 1: netem delay 700ms*

Table 10 Measuring time delays (40% bandwidth)

test	Time	IP address	Ping time		flt	Time	IP address	Ping time
Src	06:57:12	4.68.17.16	156.295		flt	06:57:14	4.68.17.16	150.228
Src	07:02:16	4.68.17.16	169.586		flt	07:02:18	4.68.17.16	152.780
Src	07:07:21	4.68.17.16	148.071		flt	07:07:22	4.68.17.16	193.392
Src	07:12:25	4.68.17.16	641.685		flt	07:12:26	4.68.17.16	871.691
Src	07:17:29	4.68.17.16	878.213		flt	07:17:32	4.68.17.16	771.476
Src	07:22:34	4.68.17.16	854.691		flt	07:22:36	4.68.17.16	545.106
Src	07:27:40	4.68.17.16	939.337		flt	07:27:41	4.68.17.16	712.706
Src	07:32:45	4.68.17.16	961.979		flt	07:32:46	4.68.17.16	695.446
Src	07:37:52	4.68.17.16	990.248		flt	07:37:51	4.68.17.16	764.338
Src	07:42:57	4.68.17.16	1020.00		flt	07:42:56	4.68.17.16	377.164
Src	07:48:02	4.68.17.16	1056.78		flt	07:48:01	4.68.17.16	575.072
Src	07:53:07	4.68.17.16	1233.60		flt	07:53:07	4.68.17.16	802.565
Src	07:58:13	4.68.17.16	1261.52		flt	07:58:11	4.68.17.16	718.470
Src	08:03:17	4.68.17.16	1437.10		flt	08:03:16	4.68.17.16	1028.86
Src	08:08:23	4.68.17.16	1492.86		flt	08:08:21	4.68.17.16	739.566

Table 11 Fixed QoS Control for test machine (40% bandwidth)

11/14/09							
	% Bw	Bandwidth	% Bw	Bandwidth	ping time	Delay	s-value
Time	download	Download	Upload	Upload			
		Mbs		Mbs	ms	Ms	
07:01	40	0.47	40	0.12	348	0	4
07:14	40	0.43	40	0.12	447	50	7.5
07:20	40	0.14	40	0.08	993	100	8.5
07:26	40	0.17	40	0.06	760	200	9
07:35	40	0.13	40	0.06	1351	300	9.5
07:43	40	0.18	40	0.06	1468	400	9.6
07:51	40	0.1	40	0.05	2037	500	9.8
07:59	40	0.11	40	0.04	1546	600	9.7
08:08	40	0.35	40	0.05	2442	700	9.8

Experiment 3b

List 6 QoS and time delay control (60% bandwidth)

Sat Nov 14 08:24:08 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 100ms
Sat Nov 14 08:32:27 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 200ms
Sat Nov 14 08:40:56 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 300ms
Sat Nov 14 08:50:52 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 400ms
Sat Nov 14 08:58:33 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 500ms
Sat Nov 14 09:06:51 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 600ms
Sat Nov 14 09:15:50 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 700ms

Table 12 Measuring time delays (60% bandwidth)

test	Time	IP address	Ping time		flt	Time	IP address	Ping time
Src	08:17:58	4.68.17.16	149.406		flt	08:17:46	4.68.17.16	150.513
Src	08:23:02	4.68.17.16	199.272		flt	08:22:50	4.68.17.16	187.601
Src	08:28:06	4.68.17.16	276.337		flt	08:27:54	4.68.17.16	265.921
Src	08:33:10	4.68.17.16	994.920		flt	08:32:58	4.68.17.16	156.507
Src	08:38:16	4.68.17.16	356.751		flt	08:38:02	4.68.17.16	148.503
Src	08:43:20	4.68.17.16	448.926		flt	08:43:06	4.68.17.16	162.459
Src	08:48:24	4.68.17.16	449.212		flt	08:48:11	4.68.17.16	147.765
Src	08:53:29	4.68.17.16	551.938		flt	08:53:15	4.68.17.16	147.335
Src	08:58:33	4.68.17.16	648.529		flt	08:58:19	4.68.17.16	182.534
Src	09:03:38	4.68.17.16	654.481		flt	09:03:23	4.68.17.16	152.922
Src	09:08:43	4.68.17.16	764.691		flt	09:08:27	4.68.17.16	398.738
Src	09:13:47	4.68.17.16	772.954		flt	09:13:32	4.68.17.16	155.829
Src	09:18:52	4.68.17.16	849.858		flt	09:18:36	4.68.17.16	145.934

Table 13 Fixed QoS Control for test machine (60% bandwidth)

11/14/09							
	% Bw	Bandwidth	% Bw	Bandwidth	Ping time	Delay	S_value
Time	download	Download	upload	Upload			
		Mbs		Mbs	ms	Ms	
08:16	60	0.59	60	0.12	446	50	6
08:24	60	0.54	60	0.12	543	100	6
08:32	60	0.47	60	0.11	748	200	7.2
08:41	60	0.37	60	0.1	944	300	8.4
08:51	60	0.5	60	0.09	1147	400	8.7
08:59	60	0.37	60	0.08	1350	500	9
09:07	60	0.55	60	0.07	1548	600	9.5
09:16	60	0.49	60	0.06	1748	700	9.4

Experiment 3c

List 7 QoS and time delay control (50% bandwidth)

Sat Nov 14 09:28:57 AST 2009	tc qdisc add dev eth1 root handle 1: netem delay 50ms
Sat Nov 14 09:42:34 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 100ms
Sat Nov 14 09:46:58 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 200ms
Sat Nov 14 09:51:44 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 300ms
Sat Nov 14 09:56:46 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 400ms
Sat Nov 14 10:01:41 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 500ms
Sat Nov 14 10:25:21 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 600ms
Sat Nov 14 10:32:23 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 700ms

Table 14 **Measuring time delays (50% bandwidth)**

test	Time	IP address	Ping time		flt	Time	IP address	Ping time
src	09:27	4.68.17.208			flt	09:27	4.68.17.208	
src	09:32	4.68.17.208			flt	09:32	4.68.17.208	
src	09:35	4.68.17.16	198.7		flt	09:35	4.68.17.16	149.4
src	09:40	4.68.17.16	242.0		flt	09:40	4.68.17.16	175.0
src	09:45	4.68.17.16	260.6		flt	09:45	4.68.17.16	147.5
src	09:51	4.68.17.16	346.9		flt	09:50	4.68.17.16	152.5
src	09:56	4.68.17.16	452.1		flt	09:56	4.68.17.16	455.9
src	10:01	4.68.17.16	660.6		flt	10:01	4.68.17.16	146.9
src	10:06	4.68.17.16	652.4		flt	10:06	4.68.17.16	152.5
src	10:11	4.68.17.16	651.9		flt	10:11	4.68.17.16	149.1

Table 15 Fixed QoS Control for test machine (50% bandwidth)

11/14/09							
	% Bw	Bandwidth	% Bw	Bw	ping time	delay	s-value
Time	download	download	upload	upload			
		Mbs		Mbs	ms	ms	
09:30	50	0.41	50	0.12	446	50	7
09:42	50	0.49	50	0.12	547	100	7.8
09:47	50	0.42	50	0.11	745	200	8
09:52	50	0.56	50	0.1	946	300	8.2
09:57	50	0.5	50	0.09	1142	400	9.1
10:02	50	0.44	50	0.08	1342	500	9.2
10:25	50	0.48	50	0.07	1543	600	9.5
10:33	50	0.41	50	0.06	1748	700	9.7

Experiment 3d

Table 16 Bandwidth Control for test machine

12/27/09				
	Bandwidth	Bandwidth		
Time	download	upload	ping time	%bandwidth
08:00	0.44	0.12	166	100%
08:30	0.42	0.13	229	100%
08:50	0.39	0.09	168	100%
08:55	0.42	0.12	166	100%
09:15	0.37	0.12	203	100%
09:22	0.32	0.08	169	80%
09:29	0.28	0.12	163	80%
09:34	0.44	0.12	184	70%
09:36	0.40	0.10	192	70%
09:43	0.37	0.08	165	60%
09:48	0.33	0.10	167	60%
09:49	0.35	0.08	167	50%
10:00	0.33	0.10	183	50%
10:04	0.35	0.11	172	40%
10:11	0.37	0.09	168	40%
10:15	0.32	0.12	178	30%
10:22	0.27	0.08	212	30%
10:26	0.31	0.12	227	20%
10:32	0.35	0.10	166	20%
10:33	0.23	0.09	173	10%
10:38	0.24	0.12	166	10%

Table 17 Ping time Control for test machine

test	Time	IP address	Ping time		flt	Time	IP address	Ping time
			ms					Ms
src	08:45	129.250.2.184	145.6		flt	08:44	129.250.2.184	152.5
src	08:50	129.250.2.184	305.2		flt	08:50	129.250.2.184	309.5
src	08:55	129.250.2.184	340.0		flt	08:55	129.250.2.184	237.7
src	09:00	129.250.2.184	269.2		flt	09:00	129.250.2.184	457.6
src	09:05	129.250.2.184	211.0		flt	09:05	129.250.2.184	218.4
src	09:10	129.250.2.184	236.4		flt	09:10	129.250.2.184	208.6
src	09:15	129.250.2.184	327.8		flt	09:15	129.250.2.184	174.7
src	09:21	129.250.2.184	603.7		flt	09:21	129.250.2.184	244.7
src	09:26	129.250.2.184	254.0		flt	09:26	129.250.2.184	225.9
src	09:31	129.250.2.184	271.3		flt	09:31	129.250.2.184	235.7
src	09:36	129.250.2.184	465.9		flt	09:36	129.250.2.184	253.9
src	09:41	129.250.2.184	336.4		flt	09:41	129.250.2.184	301.6
src	09:46	129.250.2.184	188.0		flt	09:46	129.250.2.184	195.7
src	09:52	129.250.2.184	223.2		flt	09:51	129.250.2.184	180.6
src	09:57	129.250.2.184	271.1		flt	09:57	129.250.2.184	252.0
src	10:02	129.250.2.184	170.4		flt	10:02	129.250.2.184	219.7
src	10:07	129.250.2.184	279.6		flt	10:07	129.250.2.184	256.9
src	10:12	129.250.2.184	200.4		flt	10:12	129.250.2.184	255.3
src	10:17	129.250.2.184	234.8		flt	10:17	129.250.2.184	229.6
src	10:22	129.250.2.184	247.8		flt	10:22	129.250.2.184	226.8
src	10:28	129.250.2.184	206.2		flt	10:28	129.250.2.184	256.4
src	10:33	129.250.2.184	300.6		flt	10:33	129.250.2.184	192.4
src	10:38	129.250.2.184	165.6		flt	10:38	129.250.2.184	274.1

Experiment 4a

List 8 Time delay control

Sun Dec 27 10:55:22 AST 2009	tc qdisc add dev eth1 root handle 1: netem delay 50ms:
Sun Dec 27 11:03:39 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 100ms
Sun Dec 27 11:16:27 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 200ms
Sun Dec 27 11:24:46 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 400ms
Sun Dec 27 11:32:06 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 500ms
Sun Dec 27 11:38:31 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 600ms
Sun Dec 27 11:47:07 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 700ms

Table 18 Ping time control for test machine

test	Time	IP address	Ping time		flt	Time	IP address	Ping time
			ms					ms
src	10:43	129.250.2.184	262.1		flt	10:43	129.250.2.184	277.7
src	10:48	129.250.2.184	369.7		flt	10:48	129.250.2.184	277.6
src	10:53	129.250.2.184	211.2		flt	10:53	129.250.2.184	203.3
src	10:59	129.250.2.184	295.1		flt	10:58	129.250.2.184	176.1
src	11:04	129.250.2.184	459.2		flt	11:04	129.250.2.184	226.5
src	11:09	129.250.2.184	441.3		flt	11:09	129.250.2.184	169.2
src	11:14	129.250.2.184	402.7		flt	11:14	129.250.2.184	228.5
src	11:19	129.250.2.184	473.6		flt	11:19	129.250.2.184	276.0
src	11:24	129.250.2.184	477.2		flt	11:24	129.250.2.184	173.7
src	11:30	129.250.2.184	576.2		flt	11:29	129.250.2.184	270.6
src	11:35	129.250.2.184	738.7		flt	11:35	129.250.2.184	211.6
src	11:40	129.250.2.184	775.2		flt	11:40	129.250.2.184	242.6
src	11:45	129.250.2.184	768.3		flt	11:45	129.250.2.184	380.9
Src	11:50	129.250.2.184	934.4		flt	11:50	129.250.2.184	180.0

src	11:55	129.250.2.184	898.9		flt	11:55	129.250.2.184	284.8

Table 19 Time delay Control for test machine

12/27/09					
	Bandwidth	Bandwidth	ping time	Delay	s_value
Time	download	upload	ms	Ms	
10:44	0.34	0.1	271	50	7.5
10:52	0.34	0.08	268	50	7.5
10:56	0.34	0.11	365	100	8
11:02	0.25	0.1	377	100	8
11:04	0.38	0.11	573	200	8.5
11:17	0.28	0.08	789	300	8.8
11:22	0.39	0.1	778	300	8.8
11:25	0.39	0.1	968	400	8.9
11:30	0.41	0.06	1046	400	8.9
11:32	0.31	0.07	1169	500	9.2
11:37	0.29	0.08	1210	500	9.2
11:39	0.29	0.07	1374	600	9.5
11:43	0.3	0.07	1391	600	9.5
11:47	0.29	0.06	1583	700	9.7
11:54	0.39	0.07	1571	700	9.7

Experiment 4b

Table 20 Time delay Control for test machine for Fixed Bandwidth

12/27/09	60%	60%		
	Bandwidth	Bandwidth	ping time	Delay
Time	download	upload	ms	Ms
14:40	0.34	0.12	247	0
15:02	0.43	0.09	239	0
15:06	0.31	0.10	284	50
15:10	0.29	0.12	320	50
15:13	0.35	0.11	391	100
15:16	0.32	0.07	490	100
15:19	0.42	0.12	368	100
15:20	0.37	0.11	565	200
15:23	0.28	0.09	653	200
15:29	0.35	0.10	801	300
15:33	0.40	0.10	779	300
15:39	0.32	0.10	1014	400
15:45	0.31	0.06	964	400
15:50	0.34	0.09	1264	500
15:57	0.29	0.09	1163	500
16:00	0.27	0.07	1408	600
16:04	0.32	0.08	1374	600
16:09	0.29	0.07	1564	700
16:13	0.39	0.06	1677	700
16:18	0.30	0.06	1764	800
16:20	0.33	0.06	1763	800

List 9 Time delay and bandwidth control

Sun Dec 27 15:05:14 AST 2009	tc qdisc add dev eth1 root handle 1: netem delay 50ms
Sun Dec 27 15:11:19 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 100ms
Sun Dec 27 15:19:11 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 200ms
Sun Dec 27 15:28:13 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 300ms
Sun Dec 27 15:38:28 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 400ms
Sun Dec 27 15:48:40 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 500ms
Sun Dec 27 15:58:34 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 600ms
Sun Dec 27 16:08:17 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 700ms

Table 21 Time delay Control for test machine for Fixed Bandwidth

test	Time	IP address	Ping time		flt	Time	IP address	Ping time
			ms					ms
src	14:42	4.68.17.80	223.0		flt	12:55	4.68.17.80	451.9
src	14:47	4.68.17.80	200.5		flt	13:00	4.68.17.80	233.5
src	14:52	4.68.17.80	237.9		flt	13:06	4.68.17.80	275.7
src	14:57	4.68.17.80	194.3		flt	13:11	4.68.17.80	245.5
src	15:03	4.68.17.80	292.8		flt	13:16	4.68.17.80	565.6
src	15:08	4.68.17.80	250.6		flt	15:06	4.68.17.80	728.9
src	15:13	4.68.17.80	898.2		flt	15:11	4.68.17.80	168.1
src	15:18	4.68.17.80	680.5		flt	15:16	4.68.17.80	191.4
src	15:23	4.68.17.80	416.1		flt	15:21	4.68.17.80	212.7
src	15:28	4.68.17.80	384.3		flt	15:26	4.68.17.80	199.6
src	15:34	4.68.17.80	1006.4		flt	15:32	4.68.17.80	223.2
src	15:39	4.68.17.80	458.8		flt	15:37	4.68.17.80	221.4
src	15:44	4.68.17.80	609.8		flt	15:42	4.68.17.80	260.5
src	15:49	4.68.17.80	580.7		flt	15:47	4.68.17.80	165.7
src	15:54	4.68.17.80	748.2		flt	15:52	4.68.17.80	264.7
src	15:59	4.68.17.80	902.4		flt	15:57	4.68.17.80	312.6
src	16:05	4.68.17.80	1303.1		flt	16:03	4.68.17.80	213.6
src	16:10	4.68.17.80	1251.2		flt	16:08	4.68.17.80	274.4
src	16:15	4.68.17.80	915.2		flt	16:13	4.68.17.80	741.4

Experiment 4c

List 10- Time delay control

Sun Dec 27 16:27:47 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 100ms
Sun Dec 27 16:35:27 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 200ms
Sun Dec 27 16:48:37 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 300ms
Sun Dec 27 17:01:54 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 400ms
Sun Dec 27 17:12:47 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 500ms
Sun Dec 27 17:21:03 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 600ms
Sun Dec 27 17:29:29 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 700ms
Sun Dec 27 17:35:54 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 800ms
Sun Dec 27 17:42:04 AST 2009	tc qdisc change dev eth1 root handle 1: netem delay 600ms

Table 22 Time delay Control for test machine for Fixed Bandwidth

test	Time	IP address	Ping time		flt	Time	IP address	Ping time
			ms					ms
src	16:20	4.68.17.80	1062.3		flt	16:18	4.68.17.80	759.4
src	16:25	4.68.17.80	972.9		flt	16:23	4.68.17.80	166.0
src	16:30	4.68.17.80	357.0		flt	16:28	4.68.17.80	510.0
src	16:36	4.68.17.80	364.1		flt	16:33	4.68.17.80	230.6
src	16:41	4.68.17.80	445.4		flt	16:39	4.68.17.80	241.3
src	16:46	4.68.17.80	463.6		flt	16:44	4.68.17.80	373.5
src	16:51	4.68.17.80	553.4		flt	16:49	4.68.17.80	238.0
src	16:56	4.68.17.80	512.4		flt	16:54	4.68.17.80	296.0
src	17:01	4.68.17.80	489.2		flt	16:59	4.68.17.80	601.3
src	17:07	4.68.17.80	621.9		flt	17:04	4.68.17.80	304.8
src	17:12	4.68.17.80	605.5		flt	17:10	4.68.17.80	215.1
src	17:17	4.68.17.80	699.3		flt	17:15	4.68.17.80	236.5
src	17:22	4.68.17.80	1310.0		flt	17:20	4.68.17.80	179.7
src	17:27	4.68.17.80	1454.1		flt	17:25	4.68.17.80	712.2
src	17:32	4.68.17.80	864.5		flt	17:30	4.68.17.80	728.4
src	17:38	4.68.17.80	1033.2		flt	17:35	4.68.17.80	172.2
src	17:43	4.68.17.80	799.8		flt	17:41	4.68.17.80	682.2

Table 23 Time delay Control for test machine

12/27/09				
	Bandwidth	Bandwidth	ping time	Delay
Time	Download	upload	Ms	Ms
16:29	0.33	0.09	369	100
16:33	0.37	0.11	436	100
16:38	0.50	0.09	609	200
16:44	0.50	0.10	571	200
16:50	0.29	0.09	783	300
17:01	0.31	0.04	768	300
17:03	0.35	0.09	990	400
17:10	0.33	0.09	965	400
17:14	0.38	0.08	1171	500
17:18	0.27	0.08	1169	500
17:22	0.34	0.05	2081	600
17:27	0.22	0.04	2326	600
17:30	0.35	0.07	1569	700
17:34	0.43	0.06	1584	700
17:37	0.37	0.05	1819	800
17:42	0.52	0.06	1771	800
17:45	0.31	0.07	1634	600

Experiment 5a

Table 24 Time delay changes in outgoing packets

01/04/2010						
	Bw	Bandwidth	ping time	delay	Bw	s-value
Time	download	upload	ms	ms	Mbps	
20:20	0.41	0.12	165	100	0.25	7.2
20:25	0.27	0.09	163	200	0.25	7.8
20:30	0.24	0.06	202	300	0.25	8.2
20:37	0.47	0.08	273	400	0.25	8.4
20:50	0.31	0.07	165	400	0.25	7.6
20:54	0.33	0.01	166	500	0.25	7.6
21:00	0.31	0.12	165	600	0.25	7.6

List 11 Induced time delay changes for 256 kbs bandwidth

Mon Jan 4 20:22:12 AST 2010	sh tc_ini.sh 100 256 29.250.4.161
Mon Jan 4 20:26:11 AST 2010	sh tc_delay.sh 200
Mon Jan 4 20:30:03 AST 2010	sh tc_delay.sh 300
Mon Jan 4 20:36:10 AST 2010	sh tc_delay.sh 400
Mon Jan 4 20:47:25 AST 2010	sh tc_ini.sh 400 256 " 29.250.2.184"
Mon Jan 4 20:52:02 AST 2010	sh tc_delay.sh 500
Mon Jan 4 21:00:08 AST 2010	sh tc_delay.sh 600
Mon Jan 4 21:05:36 AST 2010	sh tc_ini.sh 700 256 " 29.250.2.184"

Table 25 Ping time measurement for route 1

01/04/10					01/04/10			
Machine	Time	IP address	Ping time		Machine	Time	IP address	Ping time
Src	20:35	129.250.2.99	224.09		flt	20:35	129.250.2.99	299.91
Src	20:40	129.250.2.99	207.83		flt	20:40	129.250.2.99	281.26
Src	20:45	129.250.2.99	383.3		flt	20:45	129.250.2.99	340.31
Src	20:51	129.250.2.99	512.68		flt	20:50	129.250.2.99	420.37

Table 26 Ping time measurement for route 2

01/04/10					01/04/10			
Machine	Time	IP address	Ping time		Machine	Time	IP address	Ping time
Src	20:51	129.250.2.184	279.63		flt	20:51	129.250.2.184	503.75
Src	20:56	129.250.2.184	884.19		flt	20:56	129.250.2.184	711.19
Src	21:01	129.250.2.184	256.49		flt	21:01	129.250.2.184	240.27
Src	21:06	129.250.2.184	293.34		flt	21:06	129.250.2.184	741.21
Src	21:12	129.250.2.184	174.74		flt	21:11	129.250.2.184	267.62
					flt	21:16	129.250.2.184	149.49
					flt	21:21	129.250.2.184	149.7
					flt	21:27	129.250.2.184	145.01

Experiment 5b

List 12 Time delay and bandwidth changes

Tue Jan 5 17:59:17 BOT 2010	sh script/tc_delay.sh eth0 200
Tue Jan 5 18:05:50 BOT 2010	sh script/tc_delay.sh eth0 300
Tue Jan 5 18:10:45 BOT 2010	sh script/tc_delay.sh eth0 400
Tue Jan 5 18:19:14 BOT 2010	sh script/tc_delay.sh eth0 500
Tue Jan 5 18:23:42 BOT 2010	sh script/tc_delay.sh eth0 600
Tue Jan 5 18:29:57 BOT 2010	sh script/tc_delay.sh eth0 700
Tue Jan 5 18:36:39 BOT 2010	sh script/tc_delay.sh eth0 800
Tue Jan 5 18:45:35 BOT 2010	sh script/tc_both.sh eth0 200 160
Tue Jan 5 18:51:58 BOT 2010	sh script/tc_delay.sh eth0 400
Tue Jan 5 18:58:33 BOT 2010	sh script/tc_delay.sh eth0 600
Tue Jan 5 19:04:18 BOT 2010	sh script/tc_delay.sh eth0 800
Tue Jan 5 19:10:07 BOT 2010	sh script/tc_both.sh eth0 200 96
Tue Jan 5 19:15:23 BOT 2010	sh script/tc_delay.sh eth0 400
Tue Jan 5 19:19:49 BOT 2010	sh script/tc_delay.sh eth0 600
Tue Jan 5 19:25:05 BOT 2010	sh script/tc_delay.sh eth0 800
Tue Jan 5 19:29:05 BOT 2010	sh script/tc_both.sh eth0 200 64
Tue Jan 5 19:36:10 BOT 2010	sh script/tc_delay.sh eth0 400
Tue Jan 5 19:40:12 BOT 2010	sh script/tc_delay.sh eth0 600
Tue Jan 5 19:43:58 BOT 2010	sh script/tc_delay.sh eth0 800

Table 27 Round trip delay measurement in control machine

01/05/10					
	Bandwidth	Bandwidth	ping time	delay	Bandwidth
Time	Download	upload	Ms	ms	Mbs
17:42	0.37	0.11	165	100	0.25
17:52	0.35	0.09	168	200	0.25
18:00	0.30	0.06	168	200	0.25
18:06	0.39	0.05	170	300	0.25
18:15	0.52	0.06	172	400	0.25
18:20	0.31	0.07	166	500	0.25
18:24	0.37	0.11	165	600	0.25
18:30	0.39	0.08	175	700	0.25
18:38	0.40	0.07	165	800	0.25
18:46	0.42	0.09	166	200	0.16
18:54	0.45	0.10	167	400	0.16
18:59	0.45	0.07	162	600	0.16
19:04	0.35	0.12	181	800	0.16
19:10	0.39	0.11	190	200	0.09
19:15	0.35	0.10	165	400	0.09
19:20	0.34	0.11	168	600	0.09
19:25	0.34	0.12	189	800	0.09
19:30	0.48	0.09	166	200	0.06
19:36	0.36	0.12	164	400	0.06
19:41	0.34	0.08	166	600	0.06
19:45	0.39	0.09	165	800	0.06

List 13 Changing time delay and bandwidth

Tue Jan 5 17:56:33 AST 2010	sh script/tc_delay.sh eth1 200
Tue Jan 5 17:58:52 AST 2010	
Tue Jan 5 18:04:52 AST 2010	sh script/tc_delay.sh eth1 300
Tue Jan 5 18:10:19 AST 2010	sh script/tc_delay.sh eth1 400
Tue Jan 5 18:18:46 AST 2010	sh script/tc_delay.sh eth1 500
Tue Jan 5 18:23:20 AST 2010	sh script/tc_delay.sh eth1 600
Tue Jan 5 18:29:05 AST 2010	sh script/tc_delay.sh eth1 700
Tue Jan 5 18:36:19 AST 2010	sh script/tc_delay.sh eth1 800
Tue Jan 5 18:43:05 AST 2010	sh script/tc_both.sh eth1 200 160
Tue Jan 5 18:51:32 AST 2010	sh script/tc_delay.sh eth1 400
Tue Jan 5 18:58:06 AST 2010	sh script/tc_delay.sh eth1 600
Tue Jan 5 19:03:56 AST 2010	sh script/tc_delay.sh eth1 800
Tue Jan 5 19:09:30 AST 2010	sh script/tc_both.sh eth1 200 98
Tue Jan 5 19:14:51 AST 2010	sh script/tc_delay.sh eth1 400
Tue Jan 5 19:19:29 AST 2010	sh script/tc_delay.sh eth1 600
Tue Jan 5 19:24:43 AST 2010	sh script/tc_delay.sh eth1 800
Tue Jan 5 19:28:26 AST 2010	sh script/tc_both.sh eth1 200 64
Tue Jan 5 19:35:41 AST 2010	sh script/tc_delay.sh eth1 400
Tue Jan 5 19:39:50 AST 2010	sh script/tc_delay.sh eth1 600
Tue Jan 5 19:43:30 AST 2010	sh script/tc_delay.sh eth1 800

Table 28 Round trip delay measurement in test machine

01/05/10					01/05/10			
Machine	Time	IP address	Ping time		Machine	Time	IP address	Ping time
src	17:34	129.250.2.99	181					
src	17:39	129.250.2.99	205					
src	17:44	129.250.2.99	347					
src	17:49	129.250.2.99	498					
src	17:54	129.250.2.99	486		flt	17:53	129.250.2.99	167
src	18:00	129.250.2.99	436		flt	17:58	129.250.2.99	524
src	18:03	129.250.2.184	384		flt	18:03	129.250.2.184	471
src	18:09	129.250.2.184	653		flt	18:08	129.250.2.184	371
src	18:14	129.250.2.184	609		flt	18:13	129.250.2.184	188
src	18:19	129.250.2.184	667		flt	18:18	129.250.2.184	365
src	18:24	129.250.2.184	848		flt	18:24	129.250.2.184	156
src	18:29	129.250.2.184	865		flt	18:29	129.250.2.184	158
src	18:34	129.250.2.184	933		flt	18:34	129.250.2.184	172
src	18:40	129.250.2.184	983		flt	18:39	129.250.2.184	168
src	18:45	129.250.2.184	420		flt	18:44	129.250.2.184	396
src	18:50	129.250.2.184	356		flt	18:49	129.250.2.184	188
src	18:55	129.250.2.184	581		flt	18:54	129.250.2.184	229
src	19:00	129.250.2.184	746		flt	19:00	129.250.2.184	166
src	19:05	129.250.2.184	962		flt	19:05	129.250.2.184	232
src	19:11	129.250.2.184	543		flt	19:10	129.250.2.184	203
src	19:16	129.250.2.184	1186		flt	19:15	129.250.2.184	184
src	19:21	129.250.2.184	763		flt	19:20	129.250.2.184	191
src	19:26	129.250.2.184	974		flt	19:25	129.250.2.184	247
src	19:31	129.250.2.184	711		flt	19:31	129.250.2.184	146
src	19:36	129.250.2.184	1047		flt	19:36	129.250.2.184	150

src	19:42	129.250.2.184	808		flt	19:41	129.250.2.184	183
src	19:47	129.250.2.184	975		flt	19:46	129.250.2.184	204
					flt	19:51	129.250.2.184	140