Effect of Network OFF Times on Web Browsing QoE

Arunkumar Rajasekaran and Velanginichakravarthy Cherry

School of Computing
Blekinge Institute of Technology
37179 Karlskrona
Sweden
This thesis is submitted to the School of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering. The thesis is equivalent to 20 weeks of full time studies.

This Master Thesis is typeset using \LaTeX

Contact Information

Author 1: 
Arunkumar Rajasekaran 
M.Sc. Electrical Engineering (Telecommunication Systems) 
E-mail: arunece3@gmail.com 

Author 2: 
Velanginichakravarthy Cherry 
M.Sc. Electrical Engineering (Telecommunication Systems) 
E-mail: cherrychakri@gmail.com

University advisor: 
Junaid Shaikh 
Section/Unit: School of Computing, BTH 
Address: SE-317 79 Karlskrona 
Blekinge Institute of Technology 
Email: junaid.junaid@bth.se

School of Computing 
Blekinge Institute of Technology 
371 79 KARLSKRONA, SWEDEN

Internet: www.bth.se/com 
Phone: +46 455 385000 
SWEDEN
Abstract

The web user usually expects a better Quality of Service (QoS) from the Internet Service Provider (ISP) for the best Quality of Experience (QoE). User satisfaction and feedback is one of the most important factors for the service providers to determine their QoS and improve the network performance. Service providers are more interested in QoE to provide a better service to their users to maintain their customers in the competitive market. Since there is no much study work conducted in the QoE on web browsing, only a few studies are available for getting user feedbacks. So the ISP is facing a difficulty in the assessment of the user experience in the real-time network. Network level performance can be measured by the ISP for QoS and user feedback can be measured for QoE. There is no study available on relating both the QoS and QoE. Relating the network level performance and the user perception is a difficult task for the service providers.

In this study we have correlated both the network level traffic performance and user experience. In our experiment the user QoE is tested by applying various off times applied to some specific packets. Our main aim is to evaluate the network level performance and correlate it with the user feedback. Later, on focusing the network level performance network traffic is analyzed for different sessions with off times applied in DNS response, Base file response and Object response. We have discussed in the results by correlating the different sessions of off times that we applied and user feedback MOS. We have also discussed the relation of the network off time in the network with the number of requests sent from client to server and the number of flag bits like SYN & ACK, FIN & ACK and RST flags between the client and server. In this study we also discussed about the user feedback and how the user suffers on varying long response time. Finally, we conclude from our results about the major factor that affects the user feedback and the user interest in using the service again.
Keywords: Network Performance, Quality of Experience, Mean Opinion Score, Subjective and Objective measurements.
Acknowledgments

We are heartily thankful to the Almighty for giving us the strength and knowledge which bring us this far and working on this thesis with blessings.

Our special thanks to our beloved parents & family for their limitless love, support and encouragement for all these years.

We are grateful to our Supervisor Mr. Junaid Shaikh, for his continuous support, patience and guidance throughout our thesis work to finish successfully. He used to teach us a lot about the research work and guide us to work in the right way by correcting our mistakes to improve the quality of our thesis. He gave us his valuable feedback at each and every moment which helped us to gear up our work more.

We would express our grateful thanks to Mr. Patrik Arlos of his support and providing us guidance throughout our experiment setup and measurements.

We are thankful to Ramachandran Chakravadhamula and Mr. Tahir Minhas Nawaz for helping us in our experiments from the beginning and providing us with relevant information on our experiments.

Last but not the least we have to thank our friends for being supportive and while we were working on our thesis.

Arunkumar Rajasekaran
Velaniginichakravarthy Cherry
## Contents

- **Abstract**  i
- **Acknowledgments**  iii
- **Contents**  iv
- **List of Figures**  vi
- **List of Tables**  vii
- **Acronyms**  viii

### Introduction

1 **Introduction**  2  
   1.1 Motivation  3  
   1.2 Aims and Objectives  4  
   1.3 Research Questions  4  
   1.4 Research Methodology  5  
   1.5 Structure of Thesis  5

### Background and Related Work

2 **Background and Related Work**  7  
   2.1 Quality of Experience  9  
   2.2 Mean Opinion Score  9  
   2.3 Kaunet Traffic Shaper  9  
   2.4 Related Works  11
List of Figures

3.1.1 Basic Experiment Setup . . . . . . . . . . . . . . . . . . . . . . 14
3.1.2 Experimental setup . . . . . . . . . . . . . . . . . . . . . . . . 16
3.1.3 Format of shaping rule . . . . . . . . . . . . . . . . . . . . . . 18
3.2.1 Flow Chart of user experiment . . . . . . . . . . . . . . . . . 20
3.2.2 Participant Ratio . . . . . . . . . . . . . . . . . . . . . . . . . 21
4.1.1 DNS load time and score for first page . . . . . . . . . . . . . 26
4.2.1 Base file load time and score for first page . . . . . . . . . . 29
4.3.1 Average Number of Requests (client to server) . . . . . . . . 31
4.3.2 Flag Bits . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
4.4.1 User willingness for the reuse of service . . . . . . . . . . . . 36
A.0.1Page1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
A.0.2Score . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
A.0.3Page2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
A.0.4Page3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47
B.0.1client configuration . . . . . . . . . . . . . . . . . . . . . . . . 49
## List of Tables

<table>
<thead>
<tr>
<th>Table Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.1 ITU-T Five-level scale for quality assessment</td>
<td>9</td>
</tr>
<tr>
<td>3.1.1 Operating system on each machine</td>
<td>16</td>
</tr>
<tr>
<td>4.0.1 Sessions Description</td>
<td>24</td>
</tr>
<tr>
<td>4.1.1 DNS load time and score for first page</td>
<td>26</td>
</tr>
<tr>
<td>4.2.1 Base files load time and score for first page</td>
<td>28</td>
</tr>
<tr>
<td>4.3.1 Average number of Requests (client to server)</td>
<td>31</td>
</tr>
<tr>
<td>4.3.2 SYN and ACK Flag bits</td>
<td>33</td>
</tr>
<tr>
<td>4.3.3 FIN and ACK Flag bits</td>
<td>33</td>
</tr>
<tr>
<td>4.3.4 RST Flag bits</td>
<td>34</td>
</tr>
</tbody>
</table>
Acronyms

**ACK**  Acknowledge bit

**CI**  Confidence Interval

**DAG**  Digital Acquisition and Generation

**DNS**  Domain Name Server

**DPMI**  Distribute Passive Measurement Infrastructure

**HTML**  Hypertext Markup Language

**FIN**  Finish Bit

**GPS**  Global Positioning System

**HTTP**  Hypertext Transfer Protocol

**IP**  Internet Protocol

**ISP**  Internet Service Provider

**ITU-R**  International Telecommunication Union, Radio Communication Sector

**ITU-T**  International Telecommunication Union, Telecommunication Standardization Sector

**MArC**  Measurement Area Controller

**MP**  Measurement Point

**MOS**  Mean Opinion Score

**NIC**  Network Interface Card

**NTP**  Network Time Protocol

**OS**  Operating System

**RAM**  Random Access Memory
**RST**  Reset bit

**SD**  Standard Deviation

**SYN**  Synchronous bit

**QoE**  Quality of Experiment

**QoS**  Quality of Service

**TCP**  Transport Control Protocol

**UDP**  User Datagram Protocol

**VoIP**  Voice over Internet Protocol

**WWW**  World Wide Web
Introduction
Chapter 1

Introduction

The internet is an interconnected computer network that connects to several computers for communication and sharing information. Today recent technologies are being developed at a rapid rate fuelled by the increasing usage of the internet. With the availability of high speed data service the internet usage is being increased day by day. While using the web when the user performs some action and needs to wait for a response. If the response arrives within a short period or seconds then the user feels the system is responding immediately. Users expect to have a high network performance and less response time on the internet. Sometimes the congestion appears and latency increases which result in high response time. This results in the users impatient and the user behaviour changes.

When the user is not satisfied with the network performance then the user intends to stop or reload the browsing session which will also affect the network performance by establishing a number of connections and packet loss. The requests are sent multiple times again until the server response. So the latency increases with congestions.

For network operators or ISP (internet service providers), identifying the performance problems on their network will be the main criteria. If there is any problem in their network, they need a tool to identify and solve their issues. These tools should be having control on both network performance QoS(Quality of Service) and user perception QoE(Quality of Experience). The performance may be affected due to network performance like throughput etc. and application performance like response time. Sometimes minimum throughput may result in longer response time that annoys the user. In this case there is an analysis need to be done to correlating QoS and QoE to find out the user satisfaction.
TCP is a connection oriented communication protocol that is used to start a connection between two systems. TCP is a reliable protocol since, the status information about the connections are maintained by the TCP [24]. Normally the connections are defined by the IP (Internet Protocol) addresses and port numbers used by the two systems. TCP uses the three way handshake to establish a connection. Even if the TCP connections are established at the same time on both sides this three way handshake will be completed only in one direction. When a user starts browsing there are several processes takes place in the session from DNS (Domain Name Server) resolve till the complete data download from the server.

The main three stages that we are considering with are DNS resolve, base file load time, embedded object load time. The DNS resolution will be the time from the request sent from the client till the last response packet received in the client which includes the DNS lookup time in the server. Base file load time is the request sent to the Base HTML (Hypertext Transfer protocol) file till the last response packet for the request from the client is received. Once the Base HTML file is received the webpage will be loaded immediately in the progressive rendering method. Embedded object load time is the time taken to load an embedded object from the time of the object is requested and the last packet of the object is received from the client. We will see one by one in the upcoming chapter and how they are related to our experiment and the QoE on web browsing.

From the experiments conducted in this research work we came to know the results if the same amount of Network off time are applied at each stage on a web browsing. And also, when and which stage affects the user most to by annoyed from the network service.

1.1 Motivation

When the user expectation rises then it is the Service provider's responsibility to provide an expected QoS (Quality of Services). This makes a great competition between various service providers to serve high QoS to the users. For providing such QoS the providers might have to know the User satisfaction on their network performance and feedback of the user QoE (Quality of Experience). Thus it needs a tool to analyze both active and passive while they are using the internet.

When the network performance decreases the page response time increase, this will reflect on the user patience level. Investigations are done to analyze user experience on network-level performance, but it is difficult to
analyze user QoE in real-time operational network. In the previous research user experience was analyzed to get the feedback on the basis of periodic polls received from the users and the feedbacks were not instantaneous. So the polling results may be varied based on time delay and memory effect of the user. Later statistical methods are used with measurable metrics like MOS (Mean Opinion Score) and Perceptual Evaluation of Speech Quality. Web Traffic characteristics for estimation of QoE is also passively analyzed. In both cases performance monitoring and user impatience with passive measurement are done by offline analysis. It will be more interesting if we are measuring both the network performance from network level at the same time receive the user QoE on the web and correlate them. This motivated us for doing this research thesis to analyze the effects of network off time on web browsing QoE.

1.2 Aims and Objectives

The aims and objective of our research work are as follows.

- To estimate the web QoE of the web browsing users by getting a subjective assessment of web services.
- To correlate web browsing user feedback assessment with the network performance.
- Objective assessment of the TCP interruptions to evaluate the web browsing.

1.3 Research Questions

We have the following research questions to be answered in our research work.

RQ1: How does delay on DNS response affects the DNS load time and the web browsing QoE for an e-commerce website?

RQ2: How does delay on packets carrying the base HTML response affects the base HTML load time and the web browsing QoE for an e-commerce website?

RQ3: How does the delay on the embedded object response affects the embedded object load time and the web browsing QoE for an e-commerce website?
1.4 Research Methodology

The Research methodology chosen in our thesis is a literature review and experimentation. Literature review consists of study of related works that are done in our research area and also we have learned how KauNet works and to find out the network parameters that affect the web browsing QoE of e-commerce website. RQ1, RQ2, RQ3 we have conducted a user experiment to obtain a user MOS score. The results which are obtained from the experiment are plotted in the form of graphs and observations were drawn from the data and graph. Finally conclusions are drawn from the analysis. Detailed explanation about the experimentation is given in below section 3.1.

1.5 Structure of Thesis

The flow of the thesis is organized as follows, the first chapter describes about the introduction, motivation and scope of the thesis. The second chapter describes the background and related work. The third chapter describes the design and implementation which includes aims and objectives, research questions, research methodology, experimental setup and implementation of the experiment. The fourth chapter describes about the results and discussion of the results. Finally the fifth chapter describes conclusion and future work of our thesis.
Background and Related Work
Chapter 2

Background and Related Work

In this chapter we discuss about the background and related works on the basic topic that we cover in our thesis. This chapter will explain the complete field background for those readers who are new to this field. We are discussing the basic internet usage in today's technology, WWW (World Wide Web), QoE, MOS and Traffic Shaper KauNet.

Today the internet plays an important role in communication media providing various services worldwide for billions of users in different sectors. Whereas web browsing is one of the most popular activities on the internet. Web browsing has an enormous growth by increased number of applications available in WWW. Experience from the user differs depending on the application and the network performance. Web users usually evaluate their experience with page response time. That is defined as the time difference between the times of the first packet of a request sent from the client and the last packet of the response to the request sent by the client [5]. Service Providers improve the availability of high speed network access to the user for improving the QoE with recent technologies. But still congestion occurs due to TCP congestion control technique and increase the page response time. This disappoints the user experience and the service provider receives a bad feedback from the user.

Users are not interested in waiting for some application response time unnecessarily. Some research works are investigated in the early days on QoE on waiting time in web [28]. When the user requests for some web page and the response for the request is delayed then the user perceived quality is affected by the waiting times.
Thus the waiting time of a web page affects the user trust in the system and the service providers [28]. It is difficult to evaluate the user feedback as user expectation may differ. Users may wait for some reason but few users are not satisfied if they see response time is longer in web pages. For this reason the user perception is correlated with the network QoS, but still it is expensive and requires a lot of time [8].

The waiting time may increase due to various reasons as we have discussed, due to congestion control it may arise, due to connection establishment, due to DNS (Domain Name Server) lookup or some objects are delayed during the response from the server [29]. The user may wait for the contents to appear in the web page by taking some time, at that time if the user sees a half loaded page and some contents are not appearing then it will disappoint the user of the service.

As there are some factors in web browser loading a web page with different rendering technique. Some browser loads the page once they receive all the packets and some will not wait for receiving all the packets they load the page whenever they receive the packets immediately at once. When the rendering takes place the packets are appearing progressively before they are completely downloaded [28]. As a result the user perception of waiting time and latencies is blurred with the rendering process. This will also affect the user satisfaction as they expect a complete response from the service providers.

The only way to assess user satisfaction is to get feedback from the user on their QoE. Measuring the QoE is not an easy task as it depends on various factors. When a user gives feedback on a network performance it depends on the user background, knowledge, usage of internet, user mood and finally experience on the web. There are various methods of assessing user feedback one commonly used by the service providers is periodic polls from the user that depends on the user mood and as it is not instantaneous it may change from time to time. Another usual method is feedback from the user which is not effective as when the user is not satisfied then user changes the service [8]. There is a need of getting user feedback instantly when the user experience the service.

QoE is one of the most important factor, from the user to the service providers for giving feedback on the service when they introduce some new technologies. It is hard and expensive for the service providers to satisfy the user by providing a good QoS. The most difficult part of a provider is to access the user satisfaction. They should provide a better QoS for the users to hold their customers for continuing their service.
CHAPTER 2. BACKGROUND AND RELATED WORK

2.1 Quality of Experience

Most of the user expects a very good QoE when they use internet technologies. Normally any user will care about the experience of the service and not about the functionality. There are so many factors that influence the user experience such as personal experience and their background. Sometimes it depends on age and profession. Few users may wait for the loading of a web page but the regular internet user may not wait even for few second delays that may disappoint the user as they are used for immediate response from the web [28]. QoE may vary from different user to user. Getting user QoE is more expensive as it is reliable. There are some factors that may affect QoE such as environment, cost, credible, privacy and security.

2.2 Mean Opinion Score

QoE can be evaluated by various options such as collecting feedbacks, periodic polls and getting a mean opinion score. MOS is one of the most used options to assess the user experience. A user can give score for the QoE on the particular service once they use the service and experience it [25]. Mean Opinion Scores are commonly used from 1 to 5 that is from bad to Excellent. Quantitative results are calculated from the average of the subjective test user scores. The user test is done through the webpage and the scores are taken from the webpage itself directly from the user and stored in the database. The below table 2.2.1 shows how the user will score in the experimentation.

<table>
<thead>
<tr>
<th>Rating</th>
<th>MOS</th>
<th>Impairment</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Excellent</td>
<td>Imperceptible</td>
</tr>
<tr>
<td>4</td>
<td>Good</td>
<td>Perceptible, but annoying</td>
</tr>
<tr>
<td>3</td>
<td>Fair</td>
<td>Slightly annoying</td>
</tr>
<tr>
<td>2</td>
<td>poor</td>
<td>Annoying</td>
</tr>
<tr>
<td>1</td>
<td>Bad</td>
<td>Very annoying</td>
</tr>
</tbody>
</table>

Table 2.2.1: ITU-T Five-level scale for quality assessment

2.3 Kaunet Traffic Shaper:

A traffic shaper is an emulator that manages the traffic of a network between a server and client. To create a real time network environment for the user test, the network need to be managed by applying some delay on some packets to fulfil the user's expectation. KauNet is one of the most commonly used
network emulator. There are various emulators available such as NIST Net, NetEm, winPLEM etc. [21]. KauNet has larger control and repeatability for best performance compared to others.

KauNet has some special feature of reproducing the network behavior as many times as it is needed [19]. KauNet is an extension of DummyNet emulator with pattern handling for pattern creation and management. It is also a combination of IPFW program that is used for emulation setup and management [20]. KauNet has the additional abilities of bit-errors, reorder packets and use triggers to send information from the emulator to outside observers with the DummyNet default options of dropping of packets, bandwidth restrictions and delay to the packets. KauNet also allows the bit-errors, packet losses, bandwidth, packet reordering, triggers and delay changes controlled on time driven or data driven mode. Time driven mode is nothing but use of pattern based on per-millisecond and Data driven mode is the use of pattern based on per-packet [21].

The IPFW firewall configuration program is used by KauNet for emulation setup and management as IPFW is a firewall software application that can be installed in Linux machine with KauNet setup. Firewall configuration has some set of rules sorted from 1 to 65535 [20]. Packets are forwarded to firewall coming from the protocol. IPFW decides whether to allow the packets or deny the packet by comparing with the rules.

Pipes are one of the basic entities of an emulation system used by KauNet to regulate the traffic for applying Bandwidth, packet loss and propagation delay. For limiting the flow of traffic pipes are created. A queue is used for bandwidth sharing within the traffic flow. Multiple pipes can be created in parallel to have series traffic flows.

Delay in KauNet is calculated when the datagram completes its scheduled waiting time transmission delay and it is kept in the delay line queue. Here the bandwidth restriction is applied, but not the propagation delay. This will be done by dummynet_task() function. Delay changes are the fundamental data values representing new delay value. The possible range of delay is between 0 and 65525 milliseconds [20]. The change is applied after the specific number of packets has been transferred in data-driven mode.

To apply the restrictions of bandwidth, delay change, packet loss, packet reordering and bit errors on the network behavior the KauNet makes use of pattern to control the network traffic by per-packet or per-milliseconds.
Patterns are created for traces for network emulation from simulation experiments or identical pattern of real time network. A pattern will be generated to apply restrictions and later the generated pattern is to be inserted into the kernel for applying the network emulation. Let us see an example for the delay pattern in data driven mode. Let us see an example for the delay pattern in data driven mode.

```
root@shaper: ./patt_gen -del -pos test.dcp data 20 1, 15,10,5
root@shaper: ipfw -f flush
root@shaper: ipfw -f pipe flush
root@shaper: ipfw add allow all from any to any
root@shaper: ipfw add 1 pipe 100 ip from Server.IP to Client.IP in
root@shaper: ipfw pipe 100 config delay 10 ms bw 1Mbit/s pattern test.dcp
```

The pattern generated by delay property and specified for the position based data driven mode. The “test.dcp” is the pattern file created for inserting in the kernel for applying network emulation. Here 20 specifies the number of data the pattern should be applied and the 1 and 10 is the range of data that the pattern should be applied for first cycle with a 15 millisecond delay and from 10 to 20 data 5 milliseconds delay.

IPFW flush is to flush if there are any previously created pipe or pattern rules applied. Ipfw add 1 pipe will create a pipe with pipe id 100 from the source to destination IP. At the end the pattern gen file is loaded with the file name and specifying the pipe id for applying the network emulation with a delay of default 10 ms delay with a bandwidth 1Mbit/s. This applies a delay change in the specified packets.

### 2.4 Related Works

In this section we are going to discuss about the relevant research works done in the field of user experience in web. These relevant works in the particular area of user experience in web is not completely investigated. It is a wide range of area that can be investigated on the QoE of the web. Some of the research includes passive measurement of network traffic and user behavior observations.

In a paper [6] authors present passive analysis of web traffic characteristics. They have explained the effect of packet loss on web traffic and correlated with the user behavior. Experimented with the effect of packet loss on the duration and size of TCP connections and concluded that the duration and size of TCP connection are not practical parameter for detect-
ing user behavior. Inter-arrival time of connections is a practical parameter for detecting user behavior.

In [5] authors done a research work on user patience and web, they have presented a study on web user behavior on decrease in network performance results in increasing of page transfer time. Measurements on real time traffic to analyze the interference on network impacts the user impatience. User perceived throughput affects the interruption probability and they derived a traffic model that interrupts connections.

In [15] the author presented research on subjective responses from various types of web users with respect to response time of web browsing. MOS on response time which shows the impact on user background and expectations. In work [16] is a study on the web response time and losses in the network. The author is discussing the various effects of losses on the response time which is caused by the size of the transfers. In another paper [17] how the losses affect the QoE of the MOS is analyzed. In [18] the author presents the correlation on subjective and objective analysis of user experience on the web. In this paper author used the packet loss and throughput to compare the user experience and the Network performance.

In some of the above works the QoE is evaluated on objective measurement when the user is active on the network or subjective feedback from the user. But no study is presented in the correlation on the subjective and objective feedback of user experience based on the packet delay on a specific object. In this paper we are presenting the correlation on the traffic analysis in a controlled environment with real time scenario of a service provider network and user feedback. Also we analyze the traffic over network level to identify the delay applied to the specific packet and the effects of those delays on the user experiences.
Design and Implementation
Chapter 3

Design and Implementation

3.1 Basic Experimental setup

In this section the experimental setup and its design are discussed. In our experimental setup there will be three computers which fulfil the basic experimental setup, they are as follows.

Server for storing web pages and DNS installed in the server itself for domain name resolution. A traffic shaper is used to shape the traffic from server to client direction [9]. Users access the web pages from the server using client machine. The below figure3.1.1 illustrates the basic design.

![Figure 3.1.1: Basic Experiment Setup](image-url)
3.1.1 Experimental setup configuration

The experimental setup consists of five computers, including the basic experimental setup. They are server machine, traffic shaper machine, client machine, measurement point and consumer.

Server is installed with Ubuntu 10.10 version. The server provides the web services. The server is installed with Apache web server [13] for hosting the e-commerce website. According to Netcraft survey [14] in November 2012 apache captures the majority of market share with 57.23%, hence apache is chosen. The server is also installed with bind9 [31] DNS. As the e-commerce website has 12 domain names for 12 sessions. For more explanation on sessions refer table 4.0.1. The bind9 DNS resolve the domain name for respective phase.

The measurement point is connected in between from server to traffic shaper and from traffic shaper to the client. So now all the traffic gets captured at the measurement point in both directions, both on client side and server side. The MP is installed with two (Digital acquisition and Generation) DAG3.5E cards [10, 22] to capture the packets in both directions. The Endace DAG 3.5E cards have the ability of capturing the packets with timestamp accuracy of 60ns [22]. The MP is installed with DPMI (Distributed Passive Measurement Infrastructure) [11]. MP had a Crux 2.6 OS (Operating system) installed on it.

The Client machine is installed with the windows 7 OS and 17inch display screen. Client machine uses the google chrome web browser to send the request to the server. We had chosen chrome because it has progressive rendering feature in windows platform, as a progressive rendering is very important in our experiment.

The traffic shaper is installed with Kaunet traffic shaper. Since Kaunet has ability to shape the traffic in two different modes known as data driven mode and time driven mode, and we choose to do our experiment in data driven mode. MP and MArC (Measurement area controller) is properly clock synchronized with a GPS-NTP server.
### Table 3.1.1: Operating system on each machine

<table>
<thead>
<tr>
<th>Machine</th>
<th>Operating System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Ubuntu 10.10, kernel 2.6.32, 32 bit</td>
</tr>
<tr>
<td>Network Emulator</td>
<td>Ubuntu 10.04, kernel 2.6.32, 32 bit</td>
</tr>
<tr>
<td>Measurement Point</td>
<td>Crux 2.6</td>
</tr>
<tr>
<td>Client</td>
<td>Windows 7 Professional, 32 bit</td>
</tr>
<tr>
<td>Consumer</td>
<td>Ubuntu 10.10, Kernel 2.6.35, 32 bit</td>
</tr>
</tbody>
</table>

### 3.1.2 Phases and Parameters chosen to experiment

From the literature review we have found that the delay is the main factor that affects the user's score in web services so we have decided to apply a delay of 2, 4 and 8 sec delay [23, 39] on certain packets from server to client direction such as DNS response packets, packets carrying base and embedded objects of a web page.
We have chosen DNS, packets carrying base and embedded objects because in paper “Toward task-dependent evaluation of Web-QoE” [23] authors have applied delay to the beginning of the page, middle of the page and end of the page, all this work was done by applying delays in the browser itself. So we have decided to apply delay on the packets and so we choose DNS, packets carrying base and embedded objects.

The web pages chosen for the experiment is a shopping website that is similar to the real world shopping web sites like www.amazon.com. We have chosen shopping website as we would like to know the how the delay affects the user when he/she pay with a credit card and wait for the confirmation response message from the server. The other reason is that ITU-T (International Telecommunication Union, Telecommunication Standardization Sector) suggests to choose a search site or a shopping web site or a NEWS web site that is completely updated. As NEWS should be updated daily hence it is neglected. Whereas a search website has less objects and also search is restricted to some particular string as we have to work in local machines.

3.1.3 Shaping Rules applied in the Experiment

In the experiment the shopping site has three pages, the first page figure A.0.1 will contain images of the laptop and links, the second page figure A.0.3 will contain the product details and payment of the selected product. Finally in third page figure A.0.4 there will be “thank you message” for purchasing the product. In our case we have applied network off time only on the first page. The off time of 2, 4 and 8 seconds is applied in different phases of the web page. There are three phases in our experiment they are DNS, base and embedded objects. In the experiment there will be 12 sessions. In each session the user has to browse the three page website. A detailed explanation about the sessions is given in table 4.0.1. The below shaping rules were applied in the experiment.

Ref_1:0
DNS_2:1,2000,2,1 udp
Base_2:4,2000,5,1 tcp
Obj_2:63,2000,64,1 tcp
Ref_2:0
DNS_4:1,4000,2,1 udp
Base_4:4,4000,5,1 tcp
Obj_4:63,4000,64,1 tcp
Ref_3:0
DNS_8:1,8000,2,1 udp
In the above shaping rules, “Ref” represents a reference session where there is no off time applied. “DNS” represents off time on DNS phase. “Base” represents off time on packets carrying base information phase. “Obj” represents off time on embedded object phase.

We have done several test runs to find the packet number in case of DNS, base and object. In the test runs we have captured the response packets from server using Wireshark. In case of DNS, the first UDP response packet is delayed. In case of the Base, the fourth TCP packet from server is delayed for base HTML response. In case of Embedded object we have four important images, that a user has to select during the experiment process, the packet number of the images is 63<sup>rd</sup> packet.

### 3.2 Implementation

We have created a lab environment for the experiment. The experiment was conducted in a lab with a 17 inch flat screen with a resolution of 1280x1024 pixels; the user will be given instructions initially in the experimental procedure. Before starting the experiment, we have to make sure that there is no shaping in the traffic shaper and there is no capture of packets in the consumer. The first step of the experiment is to fill in a form for identification purpose and then user continues the experiment. The form contains fields such as
CHAPTER 3. DESIGN AND IMPLEMENTATION

- First Name:
- Last Name:
- Email:
- Age:
- Sex:
- What is your expertise level in web browsing?
- How often do you do web browsing?

After starting the experiment, in each session the user has to give the score for the first page, then continue to the second and third page. The user cannot go to the second page until the score is submitted for the first page. If the user forgot to submit the score and tries to go to the second page then a pop-up window appears for submitting the score. The score given by the user is stored in the local database of the client machine for analysis purpose. After completing all the 12 sessions, a page with “Sign out” button and “Thank you” message will appear. The user has to continue the experiment until the sign out button and thank you message appears on the screen. Each user will have 12 sessions, each session will have three pages. The user gives a score only for the first page. During the experiment, the user should never use stop, reload, back/forward or close the browser. So whenever a user starts a new session then the client triggers the traffic shaper and consumer.

The client triggers the traffic shaper to delete the old shaping rules and apply new shaping rules according to the session and triggers the consumer to stop the previous capture and start new capture.
3.2.1 Data Collection

In our experiment we have taken around 46 users MOS, from that 46 users, we have filtered the outliers from the total users. Outliers were defined based on the user feedback and score figure A.0.2. If a user satisfies any one of the below conditions then the user is called as outlier.

- When the user is submitting the score, the user was asked “Which page is this?” and the option “Product selection” has to be selected. If the user selects other options, then the user is called as outlier.

- If the score by a user is constant throughout all the sessions, then the user is called as outlier.
We have a total of 40 users out of that 34 users are male and 6 users are female. All 40 users use the internet every day. There were about 16 users, who are expert and the rest 24 are medium in using the internet. We have collected users with an age group range of 20-30 years. Users were from different nation and region.

![Participant Ratio](image)

Figure 3.2.2: Participant Ratio
Results and Discussions
Chapter 4

Results and Discussions

In this chapter we are discussing about our results analyzed from the data collected from the measurement point. This chapter deals with the network level analysis for different sessions of the user experiment by applying network off time for 2, 4 and 8 seconds. Users who did not participate in the experiment with interest are filtered out based on their scores and feedback. In our results we have calculated statistical parameters like Mean, SD (Standard Deviation) and CI (Confidence Interval) of the off time applied in DNS, packets carrying base HTML file and embedded object data using PERL. PERL is used because of its ease of use, speed of calculation and efficient memory usage [30]. The data collected from the measurement point is in binary format with file extension .cap. The .cap files are stored in consumer machine. Consumer is installed with capshow sofware that uses Libcap library utilities. Capshow sofware is a part of DPMI tool which is developed by Dr. Patrik Arlos [11]. We are converting the .cap file into a human readable text file format for analysis, using capshow command. The example of capshow command is given below.

```bash
#Capshow -x -H file1.cap > out.txt
```

In our user experiment there are 3 sets, each set with 4 sessions. Out of the 4 sessions, in one session there is no network off time applied and is named as “Ref” session which is used as a baseline for comparing with other sessions of respective set. The three sets can be classified as follows DNS, packets carrying base file and Embedded object respectively. In total there are 12 sessions, each session has network off time of ’t seconds’; where ’t’ can 2, 4 and 8 seconds as shown in the table 4.0.1.
CHAPTER 4. RESULTS AND DISCUSSIONS

24

Table 4.0.1: Sessions Description

<table>
<thead>
<tr>
<th>Session</th>
<th>Session Name</th>
<th>Session Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ref</td>
<td>No delay.</td>
</tr>
<tr>
<td>2</td>
<td>DNS_2</td>
<td>2 seconds delay on DNS response packet from server to client.</td>
</tr>
<tr>
<td>3</td>
<td>Base_2</td>
<td>2 seconds delay on Base HTML file response packet from server to client.</td>
</tr>
<tr>
<td>4</td>
<td>Obj_2</td>
<td>2 seconds delay on embedded object response packet from server to client.</td>
</tr>
<tr>
<td>5</td>
<td>Ref</td>
<td>No delay.</td>
</tr>
<tr>
<td>6</td>
<td>DNS_4</td>
<td>4 seconds delay on DNS response packet from server to client.</td>
</tr>
<tr>
<td>7</td>
<td>Base_4</td>
<td>4 seconds delay on Base HTML file response packet from server to client.</td>
</tr>
<tr>
<td>8</td>
<td>Obj_4</td>
<td>4 seconds delay on embedded object response packet from server to client.</td>
</tr>
<tr>
<td>9</td>
<td>Ref</td>
<td>No delay.</td>
</tr>
<tr>
<td>10</td>
<td>DNS_8</td>
<td>8 seconds delay on DNS response packet from server to client.</td>
</tr>
<tr>
<td>11</td>
<td>Base_8</td>
<td>8 seconds delay on Base HTML file response packet from server to client.</td>
</tr>
<tr>
<td>12</td>
<td>Obj_8</td>
<td>8 seconds delay on embedded object response packet from server to client.</td>
</tr>
</tbody>
</table>

First, there are 3 reference sessions 1, 5 and 9 without any off time in the network which is mentioned as Ref. Second, there are 3 DNS sessions 2, 6 and 10 by applying various off times of 2, 4 and 8 second in the DNS response packet which is mentioned as DNS_2, DNS_4 and DNS_8. Third, there are 3 Base sessions 3, 7 and 11 by applying various off times of 2, 4 and 8 second in the base file response packet which is mentioned as Base_2, Base_4 and Base_8. Finally, there are 3 object sessions 4, 8 and 12 by applying various off times of 2, 4 and 8 second in a particular embedded object response packet from the server which is mentioned as Obj_2, Obj_4 and Obj_8. In this chapter we will discuss about each set of sessions with their
corresponding user feedback score.

4.1 DNS Load Time

DNS load time is the total time taken by the client to send DNS query request and receive the DNS response from the DNS server. In the experiment every user goes through the DNS off time for 2, 4 and 8 second delay. The below Figure 4.1.1 shows the DNS load time (DNS_Load time) varying for different off times. In the graph we can see DNS delay is applied for session DNS_2, DNS_4 and DNS_8 with a reference sessions Ref. Here the reference session is not having any off time which loads the web page at a normal download time.

The DNS load time is calculated by finding the time of the first DNS request packet from the client to the server and the time is noted down as time \( t_{\text{start}} \), the time of the DNS response packet from the server to the client after DNS lookup is resolved is identified in the client and the time is noted down as time \( t_{\text{end}} \). With the time difference between \( t_{\text{end}} \) and \( t_{\text{start}} \) forms the DNS load time. Most of the time the DNS response will be carried out as a single packet from server to client. If there are any multiple numbers of DNS response packets from the server then the time of the last packet of the response is considered as \( t_{\text{end}} \). This difference time makes the DNS load time as requested from client to server and responded from server to client.

In table 4.1.1 shows the average off time applied for each session for all the users and the average score given by the users. Also it has the 95% of confidence interval for both the DNS load time (CI_DNS Load time) and User scores (CI_Score). The reference session is having the normal DNS load time approximately 0.0004 seconds that is the time taken from the client to send DNS query request to the server and DNS server responds to the client without any off time in the network. DNS load time with no off time has an average score of around 4.7 which shows the user perception is not affected. The other sessions are DNS_2, DNS_4 and DNS_8 in which the DNS off times is applied with 2, 4 and 8 seconds respectively. When 2 seconds off time is applied to DNS response we have an average of around 2.003 seconds DNS_Load time with a confidence interval of about 0.167 seconds. The user’s score for this response time has an impact with an average of 4.130 which is lesser than the reference session score. By applying long off time in DNS response the average DNS load time is increasing with the delay. As we can see the average DNS_Load time is 4.030 for DNS_4 and 8.004 for DNS_8 which received user score 2.7 and 2.0 respectively. When applying long off time in DNS the user’s score is highly affected as the user is not
satisfied with the initial response time from the server. The initial delay in
the DNS lookup makes the user suffer, the user feels that the network is not
responding which results in a bad impression on the network performance
or on the service. So the users gave a very low score for the DNS delay
compared to the base and object delay sessions.

<table>
<thead>
<tr>
<th></th>
<th>Ref</th>
<th>DNS_2</th>
<th>DNS_4</th>
<th>DNS_8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average DNS Load time [s]</td>
<td>0.004</td>
<td>2.003</td>
<td>4.030</td>
<td>8.004</td>
</tr>
<tr>
<td>Average score</td>
<td>4.7</td>
<td>4.1</td>
<td>2.7</td>
<td>2.0</td>
</tr>
<tr>
<td>95%CI_DNS Load time [s]</td>
<td>0.000</td>
<td>0.167</td>
<td>0.172</td>
<td>0.254</td>
</tr>
<tr>
<td>95%CI_Score</td>
<td>0.177</td>
<td>0.275</td>
<td>0.274</td>
<td>0.295</td>
</tr>
</tbody>
</table>

Table 4.1.1: DNS load time and score for first page

Figure 4.1.1: DNS load time and score for first page

As the DNS has to look up for the IP address for that particular web site.
So when the DNS response packet is delayed then there is no page displayed
until the DNS has been resolved.
CHAPTER 4. RESULTS AND DISCUSSIONS

DNS Query will be sent from the client to the server. DNS service uses UDP protocol. Once the DNS translates the domain name to the IP address, the server responds to the client and then the TCP connection requested from the client is sent to establish the connection [12]. When the DNS response is delayed the connection will not be established and the user doesn’t get any response in the browser. After the completion of ’t seconds’ off time in the DNS response, the TCP connection is established and the upcoming packets are loaded immediately. This initial off time in the DNS annoys the user even the page is loaded immediately after DNS lookup. Hence the MOS score is highly affected as the DNS load time is increased.

4.2 Base File Load Time

When a user starts a new session for browsing a web page the DNS is resolved and the three way handshake takes place. The three way handshake establish the TCP connection between the server and the client. Once the connection is established the Base file request is sent from the client to the server. The server responds to the request by sending the base file as packets to the client by sending followed by an acknowledgment. After the Base HTML file is received by the client, the client starts sending the GET request for the embedded objects one by one to the server. The objects are transferred from the client to the server once they receive the request one by one to display the web page content.

Here we apply off time in the base file response packet from the server that will also delay the consecutive packets as they are queued until the delay ’t seconds’ is completed. When the ’t seconds’ is completed then all the other packets that are in the queue will be received in the client without any more delay and the web page is loaded immediately. This will show the users that the web browser started downloading the web page by displaying the content one by one as the order of the packets received in the server after the delay. At first the title of the page will be displayed without any delay, then the page content will be downloaded and displayed one by one with an interval of time as the network off time is applied at the base HTML file. The users can see the page is loading slowly with an interval between the packets one by one. The user will not be able to see any objects in the page until the delay ’t seconds’ is completed. This kind of process makes the user think that the network performance is not good or up to the level expected. Here the users initially wait for the page to complete the download, but the waiting time annoys the user more and that affects the user score.
The Base load time is calculated by finding the time of the first Base HTML request packet from the client to the server and the time is noted down as time $t_{\text{bstart}}$, the time of the last Base HTML response packet from the server to the client as time $t_{\text{bend}}$. Once the connection is established after DNS lookup the client starts sending the GET requests. The first GET request will be the HTTP packet that is Base HTML packet. After HTML Base file is received the client sends the next GETs request one by one. Here the time of the HTML packet is identified with HTTP/1.1 200 OK (text/html) as the Base HTML end packet and the time is noted down as $t_{\text{bend}}$. The difference between the base start time and the end time will make the base load time on the client.

In this part of experiment the user goes through Base file off time of 2, 4 and 8 seconds. The table 4.2.1 shows the reference and the other three Base file delays with their respective scores. Reference session does not have any network off time as it loads in the normal network performance that has a base file load time of approximately 0.002 seconds. As expected the normal session score is approximately 4.7 as there is no off time in any packets that satisfies the user with less waiting time. The Base_2, Base_4 and Base_8 are base file load time when the network of time is applied to the base file response packets with a delay of 2, 4 and 8 seconds. Here when 2 seconds off is applied in the base response from the server we could see the base load time of about approximately 2.007 seconds as an average for all the users. This has an impact in the user feedback as the score of an average of approximately 3.8 is received. When the off time is increased from 2 to 4 and 8 seconds the base file load time is also increased to approximately 4.041 and approximately 8.004 seconds on average. This increase in base file load time and waiting time affects the user feedback MOS score as we get approximately 3.2 and approximately 2.8 of an average score of 4 and 8 seconds respectively.

<table>
<thead>
<tr>
<th></th>
<th>Ref</th>
<th>Base_2</th>
<th>Base_4</th>
<th>Base_8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Base file load time [s]</td>
<td>0.002</td>
<td>2.007</td>
<td>4.041</td>
<td>8.004</td>
</tr>
<tr>
<td>Average score</td>
<td>4.7</td>
<td>3.8</td>
<td>3.2</td>
<td>2.8</td>
</tr>
<tr>
<td>95% CI Base file [s]</td>
<td>0.0004</td>
<td>0.116</td>
<td>0.302</td>
<td>0.469</td>
</tr>
<tr>
<td>95% CI Score</td>
<td>0.177</td>
<td>0.170</td>
<td>0.213</td>
<td>0.231</td>
</tr>
</tbody>
</table>

Table 4.2.1: Base files load time and score for first page
CHAPTER 4. RESULTS AND DISCUSSIONS

Figure 4.2.1: Base file load time and score for first page

The figure 4.2.1 gives us the details about the base file load time. The figure 4.2.1 shows us how the base file load time is varied when we increase the delay in the base file datagram. Also the graph has the respective MOS score from the user for the base file delay which is affected due to the increase of delay in the base file.

4.3 Delay on Embedded Object

4.3.1 Requests from client

In our experiment, for each session it takes around a certain number of requests sent from the client to the server. In this section the number of requests from client to the server for all the sessions like reference session, without any off time in the network and other sessions with network off time applied to specific packets are calculated and analyzed. In the figure 4.3.1 we can see, when the network off time is increased the number packets transmitted from client to server is also increased.
In the normal TCP communication when the client sent a request to the server then the client will wait for the response from the server for some particular time. If the client is not receiving any response from the server until the requested time out then the client will retransmit the request again. This will increase the number of requests and the number of communication packets in between the client and the server. When there is some network off time in between the client and the server then the response packets were not received in the client at that time. So the client starts sending the retransmissions of request packets again. This increases the number of GET requests from the client to the server. Here we are counting the number of GET requests send from client to server for the whole session to calculate the average number of GET requests sent from client to server. The number of GET requested is counted from the beginning of the page download once the Base HTML is requested and till the last embedded object GET request is requested and downloaded. This will give us the number of GET request that are sent for retransmission.

From table 4.3.1 we could see the number of requests sent from the client to the server for all the sessions like DNS, Base HTML and Embedded object. The number of requests from client to server increase when the off time is increased, but this is not in the case DNS delay. When off time is applied in the DNS we do not see much difference in the number of requests. When off time is applied to a DNS response packet, it is not affecting the TCP packets which could be observed from our analysis. The number of requests for the reference session and DNS off time session is approximately same.

When some particular packet delays from the server in case of base and object session, we could see many numbers of TCP requests for the same file again and again sent from the client to the server. We could also observe many parallel TCP connections between server and client. So here the same file requests on a different connection. An increase in the number of requests and TCP connections will result in the retransmission. Retransmission overloads the network traffic. This affects the network traffic, due to the degradation in the network performance user faces a bad QoE. This could be the reason for the downfall of the MOS over web browsing user. In table 4.3.1 ARCS represent the “average number of requests from client to server”.
### CHAPTER 4. RESULTS AND DISCUSSIONS

<table>
<thead>
<tr>
<th></th>
<th>Ref</th>
<th>DNS_2</th>
<th>Base_2</th>
<th>Obj_2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ARCS</strong></td>
<td>22</td>
<td>22</td>
<td>23</td>
<td>27</td>
</tr>
<tr>
<td><strong>Score</strong></td>
<td>4.7</td>
<td>4.1</td>
<td>3.8</td>
<td>3.5</td>
</tr>
<tr>
<td><strong>95%CI_Score</strong></td>
<td>0.177</td>
<td>0.275</td>
<td>0.170</td>
<td>0.222</td>
</tr>
<tr>
<td><strong>95%CI_ARCS</strong></td>
<td>0.226</td>
<td>0.136</td>
<td>0.251</td>
<td>0.649</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Ref</th>
<th>DNS_4</th>
<th>Base_4</th>
<th>Obj_4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ARCS</strong></td>
<td>22</td>
<td>22</td>
<td>23</td>
<td>31</td>
</tr>
<tr>
<td><strong>Score</strong></td>
<td>4.5</td>
<td>2.7</td>
<td>3.2</td>
<td>2.9</td>
</tr>
<tr>
<td><strong>95%CI_Score</strong></td>
<td>0.171</td>
<td>0.274</td>
<td>0.213</td>
<td>0.213</td>
</tr>
<tr>
<td><strong>95%CI_ARCS</strong></td>
<td>0.117</td>
<td>0.153</td>
<td>0.266</td>
<td>1.154</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Ref</th>
<th>DNS_8</th>
<th>Base_8</th>
<th>Obj_8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ARCS</strong></td>
<td>22</td>
<td>22</td>
<td>25</td>
<td>32</td>
</tr>
<tr>
<td><strong>Score</strong></td>
<td>4.4</td>
<td>2.0</td>
<td>2.8</td>
<td>2.6</td>
</tr>
<tr>
<td><strong>95%CI_Score</strong></td>
<td>0.196</td>
<td>0.293</td>
<td>0.231</td>
<td>0.219</td>
</tr>
<tr>
<td><strong>95%CI_ARCS</strong></td>
<td>0.138</td>
<td>0.131</td>
<td>1.249</td>
<td>1.845</td>
</tr>
</tbody>
</table>

Table 4.3.1: Average number of Requests (client to server)

![Figure 4.3.1: Average Number of Requests vs Score](image)

Figure 4.3.1: Average Number of Requests (client to server)
4.3.2 TCP Flag Bits

From the experimental results, we observed some different behavior in the flag bits particularly in SYN & ACK, FIN & ACK and RST flags for different sessions of delay we applied. TCP packets contain flag bits in their header to influence the flow of data across the TCP connection. Flag bits are identified from the header of the each packet and differentiated their path from client to server and server to client. Here we have discussed about the number of flag bits. Flag bits are identified from the header of the TCP packet and differentiated by their path from client to server and server and client. When network off time is applied on the objects the number of flag bits varies from the normal sessions. These SYN & ACK, FIN & ACK and RST flags belong to the connection establishment and connection reset process. With these flag bits we can see the changes in the behavior of network performance when there is a network off time on the web browsing, which will result in the user satisfaction.

SYN & ACK: As we could see in the figure 4.3.2 the average number of SYN & ACK is increasing in all the sessions compared to reference session when we applied off time. As we know SYN & ACK represents the three-way handshake of a TCP connection. When the delay is applied to the packet which server responds to the request from the client, the response packet is not received by the client. As the server is not receiving the ACK from the client for the datagram sent to the request the server waits for the ACK segment from the client. If the client is running out of waiting time for the response, then the client initiates to close the connection. Again the request is sent from the client to the server for the same data in a new connection. So the new connections are created every time when we apply more delay. We could see that in the figure 4.3.2 the number of requests is increasing with the delay t seconds is increased when compared to the reference session. But still the server keeps on responding to the previous request. This makes the data traffic busy and flow down the network performance.
CHAPTER 4. RESULTS AND DISCUSSIONS

Table 4.3.2: SYN and ACK Flag bits

<table>
<thead>
<tr>
<th></th>
<th>Ref</th>
<th>DNS_2</th>
<th>Base_2</th>
<th>Obj_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYN &amp; ACK</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>95% CI</td>
<td>0.226</td>
<td>0.582</td>
<td>0.552</td>
<td>0.411</td>
</tr>
<tr>
<td></td>
<td>Ref</td>
<td>DNS_4</td>
<td>Base_4</td>
<td>Obj_4</td>
</tr>
<tr>
<td>SYN &amp; ACK</td>
<td>3</td>
<td>6</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>95% CI</td>
<td>0.365</td>
<td>0.480</td>
<td>0.601</td>
<td>0.656</td>
</tr>
<tr>
<td></td>
<td>Ref</td>
<td>DNS_8</td>
<td>Base_8</td>
<td>Obj_8</td>
</tr>
<tr>
<td>SYN &amp; ACK</td>
<td>3</td>
<td>6</td>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td>95% CI</td>
<td>0.314</td>
<td>0.433</td>
<td>0.689</td>
<td>0.830</td>
</tr>
</tbody>
</table>

Table 4.3.3: FIN and ACK Flag bits

<table>
<thead>
<tr>
<th></th>
<th>Ref</th>
<th>DNS_2</th>
<th>Base_2</th>
<th>Obj_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIN &amp; ACK</td>
<td>7</td>
<td>13</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>95% CI</td>
<td>0.724</td>
<td>0.945</td>
<td>0.854</td>
<td>0.985</td>
</tr>
<tr>
<td></td>
<td>Ref</td>
<td>DNS_4</td>
<td>Base_4</td>
<td>Obj_4</td>
</tr>
<tr>
<td>FIN &amp; ACK</td>
<td>7</td>
<td>12</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>95% CI</td>
<td>0.697</td>
<td>1.229</td>
<td>1.151</td>
<td>1.094</td>
</tr>
<tr>
<td></td>
<td>Ref</td>
<td>DNS_8</td>
<td>Base_8</td>
<td>Obj_8</td>
</tr>
<tr>
<td>FIN &amp; ACK</td>
<td>6</td>
<td>14</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>95% CI</td>
<td>0.719</td>
<td>1.241</td>
<td>1.093</td>
<td>1.213</td>
</tr>
</tbody>
</table>

FIN & ACK: With the SYN & ACK bits which we discussed above the server also sends more number of FIN & ACK bits to the client when we apply the delay in the base file. The graph shows the number of FIN & ACK bits increased in base file delay session compared to the other reference, DNS delay and Object delay sessions. This could be one of the reasons that bring down the network traffic due to the retransmission of a delayed datagram and number of TCP connections closed between the server and the client as the client creates new TCP connections when the server does not respond to the request. There is a steady increase in number of FIN & ACK bits when the delay t seconds are increased. So that show when the delay 't seconds' increased the number of TCP connection increases and the connection is not closed properly till the end of the session. This increases the network traffic and affects the user by delay in the load time. So the MOS has the impact of this network performance.

RST: Reset RST flag is another one of the reasons that is shown in the figure 4.3.2 which shows that applying delay 't seconds' to any object in the webpage the response will not be received in the client until 't seconds' is completed. The client keeps on retransmitting the request with new
TCP connections which is an abnormal behavior. So the server aborts the connection or denies the request that is repeated by sending RST flag bits. This increase in RST flag bits is also causing the network traffic to increase and reduces the network performance. When the network performance is reduced the user MOS score is also affected respectively.

These flag bits affect the network performance much in the sessions when delay ‘t seconds’ are applied on the object when compared to the delay applied on the DNS or Base file. This could be due to the retransmission of the same object request and response as when the delay is applied in one object the consecutive objects will be kept in the queue and those datagram are also delayed. So we are getting increased flag bits as seen in the figure 4.3.2 below. The figure 4.3.2 shows how the flag bits vary according to the different types of delay that we applied.

<table>
<thead>
<tr>
<th></th>
<th>Ref</th>
<th>DNS_2</th>
<th>Base_2</th>
<th>Obj_2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RST</strong></td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>95% CI</td>
<td>0.677</td>
<td>1.221</td>
<td>1.115</td>
<td>1.187</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Ref</th>
<th>DNS_4</th>
<th>Base_4</th>
<th>Obj_4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RST</strong></td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>95% CI</td>
<td>0.878</td>
<td>1.038</td>
<td>1.379</td>
<td>1.897</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Ref</th>
<th>DNS_8</th>
<th>Base_8</th>
<th>Obj_8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RST</strong></td>
<td>5</td>
<td>10</td>
<td>12</td>
<td>17</td>
</tr>
<tr>
<td>95% CI</td>
<td>0.840</td>
<td>1.655</td>
<td>1.538</td>
<td>2.162</td>
</tr>
</tbody>
</table>

Table 4.3.4: RST Flag bits
CHAPTER 4. RESULTS AND DISCUSSIONS

4.4 Willingness

The user gives the feedback for different type of off time applied. The user is asked whether he/she would like to use this service again. For those feedbacks the figure 4.4.1 is plotted with DNS, Base and Object off times. In the figure 4.4.1 “YES” represents the users who would like to use this internet service again and “NO” represents the user who is not interested in using this kind of internet service again. The figure 4.4.1 shows, as the off time in DNS and base file is increased the number of users who would like to use the service again is being decreased and those who are not willing to use the service again are increasing. The same scenario can be seen in case Base and Object delay. Comparing DNS_4, DNS_8 and Base_4, Base_8 delay, we can observe more number of users who polled “NO” is greater for DNS. Here in the Object delay we can see more users who polled “NO” than DNS and Base, but from the user MOS figure 4.3.1 we observed that DNS delay has more impact.
Figure 4.4.1: User willingness for the reuse of service
Conclusions and Future Work
Chapter 5
Conclusions and Future Work

5.1 Conclusions

RQ1. How does delay on DNS response affects the DNS load time and the web browsing QoE for an e-commerce website?
From the above results we conclude that the user has suffered a lot when we have a greater or equal to 4 seconds delay in DNS. For lesser delays like 2 seconds, the user is not much affected but still had some impact. Even the DNS delay is having less number of requests when compared to the Base and Object sessions the user gave lower scores for the DNS delay as the DNS off time affects the initial page load. In DNS delay user will not see any transfer of data until the name server has resolved the domain name and reply the response to the corresponding client. From the user perspective this initial delay makes the user feel that the network is not responding or the network performance is bad so that the page is not initially loaded.

RQ2. How does delay on packets carrying the base HTML response affects the base HTML load time and the web browsing QoE for an e-commerce website?
We conclude that the user is affected when the applied delay is greater or equal to 4 seconds on the base file. Here in base file delay as the DNS has replied with a response and also the connection is established for data transfer the page is not completely downloaded. The base file is delayed that will stop the users possibly viewing the web page till the delay time is completed. Also repeated request from the client is sent as the client is not receiving any response from the server during the off time, this will increase the network traffic which results in reduced network performance. From the user perspective the user feels better when compared to the DNS delay as there is some progress in the web page download, but the scores are better when compared with the scores given for DNS delay.
RQ3. How does the delay on the embedded object response affects the embedded object load time and the web browsing QoE for an e-commerce website?

If the object delay is more than or equal to 2 seconds then we observed that some of the user is not interested in the using this network again. This is because the object delay is introduced during the transfers of the data from the server. As the objects are delayed, the client creates new TCP connections as it is not getting the response from the server. These off times during the transfer annoy the users. If off time is increased there will be an increase in the number of packets during the transfer implies network congestion. We conclude that from the user perspective the object delay is having an impact for even 2 seconds of delay in the transfer, but the user’s score is more than the score for DNS delay. This shows that the user feel better when they are able to see some content is downloaded and some object is not loaded.

5.2 Future Work

This work can be extended for mobile web browsing, wireless network and video streaming. We could also extend the work in time driven mode, pattern based shaping by imitating the pattern from real time network traffic. It can also be extended in by applying the equal amount of off time both request and response packets.
Bibliography


[38] ———, “Network monitoring is people: Understanding end-user perception of network problems,” 2010.

Appendix A

Some screen shots of the web pages used in the experiment.

Figure A.0.1: Page1
Figure A.0.2: Score

Figure A.0.3: Page 2
Your Payment was Successful!
You have successfully purchased the product.
Thank you for your purchase.

Figure A.0.4: Page 3
Appendix B

Configurations of Experimental setup, modify the file "`/etc/network/interfaces`" and restart the networking services for permanent configurations.

**Server Machine:**
```
auto eth0
iface eth0 inet static
  address 10.0.1.1
  netmask 255.255.255.0
  broadcast 10.0.1.255
  gateway 10.0.1.2
  mtu 1500
```

**Traffic Shaper:**
```
auto eth2
iface eth2 inet static
  address 10.0.1.2
  netmask 255.255.255.0
  broadcast 10.0.1.255
  gateway 10.0.1.0
  mtu 1500
```

```
auto eth1
iface eth1 inet static
  address 192.168.0.100
  netmask 255.255.255.0
  broadcast 192.168.0.255
  gateway 192.168.0.1
  mtu 1500
```
**Router:**
Configure the Router with the network 192.168.0.0/24 and assign router with IP 192.168.0.1/24 and disable the DHCP server in router.

**Client Machine:**
As client machine is installed with windows7 OS, make sure to set the preferred DNS server to be the actual server.

---

![Client Configuration](image)

Figure B.0.1: client configuration