

**A**

**Project Report On**

**PACKET CAPTURE BASED POST MORTEM OF  
REAL TIME MULTIMEDIA APPLICATIONS**

**Submitted by**

<b>Kunal Parakh</b>	<b>B-80058586</b>
<b>Sakshi Marda</b>	<b>B-80058571</b>
<b>Sheeban Shaikh</b>	<b>B-80058611</b>

Under the guidance of

**Prof. S.S. Pande**

*In partial fulfilment of*  
**Bachelor of Engineering**  
[B. E. Information Technology]

*at*



**Department of Information Technology**  
**Pune Institute of Computer Technology**  
**Dhankawadi, Pune – 411043**

**Affiliated to**

**University of Pune**

**[2012 – 2013]**

# **Pune Institute of Computer Technology**

**Department of Information Technology**

**Dhankawadi, Pune – 411043**



## ***CERTIFICATE***

This is to certify that the Dissertation titled  
**“PACKET CAPTURE BASED POST MORTEM OF REAL TIME MULTIMEDIA  
APPLICATIONS”**

Submitted by

<b>Kunal Parakh</b>	<b>B-80058586</b>
<b>Sakshi Marda</b>	<b>B-80058571</b>
<b>Sheeban Shaikh</b>	<b>B-80058611</b>

is a record of bonafide work carried out by him/her, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Information Technology) at Pune Institute of Computer Technology, Pune under the University of Pune. This work is done during year 2012-2013, under our guidance.

---

**Prof.S.S. Pande**  
Project Guide

---

**Dr. Emmanuel M.**  
HOD, IT Department

---

**Dr. P. T. Kulkarni**  
Principal PICT

External Examiner : \_\_\_\_\_

**Date:**

## **Acknowledgements**

We are profoundly grateful to **Prof. S.S. Pande** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

We would like to express deepest appreciation towards **Dr. P. T. Kulkarni**, Principal PICT, Pune, **Dr. Emmanuel M.** HOD Information Technology Department and **Prof. Manish R. Khodaskar** (Project Coordinator) whose invaluable guidance supported us in completing this project.

We are particularly grateful to **Mr. Manish Sapariya** (Kpoint Technologies Pvt. Ltd.) who allows us to work in the company.

At last we must express our sincere heartfelt gratitude to all the staff members of Information Technology Department who helped us directly or indirectly during this course of work.

**Kunal Parakh**  
**Sakshi Marda**  
**Sheeban Shaikh**

# Abstract

The project aims at doing post-mortem analysis at packet-level. It involves detailed study of real time multimedia applications like virtual classrooms majorly consisting of protocols like HTTP, TCP, Jabber/XML and RTMP using packet capture. The basic purpose is to gather relevant information related to packets like average unacknowledged data from client/server over a period of time, average rate of dropped/out of sync/retransmitted/re-acknowledged packets, find out the jitter in network, etc. for a given connection. From this knowledge, it is possible to find the signs of abnormal behavior in the network which includes jitter, defective packets(out of order, retransmitted, lost), throughput and make sense out of all the data gathered to pin point the particular packet originating from the particular source. Another measure is to do statistical analysis on all the data that has been gathered and present it in the form of charts for better understanding. These statistics direct towards identifying the usage of networks by each customer and the cost borne by the server to serve such clients and to dig-into the minute details found appropriate from business perspective. From these statistics obtained, inference is drawn about the network conditions and issues raised by the users. Finally a web based tool is designed to identify user issues and understand user experiences so as to either fix the infrastructure or code.

Keywords - Protocols, Statistical Analysis, Issues

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Introduction . . . . .	2
1.2	Motivation . . . . .	2
1.3	Background . . . . .	2
1.4	Need . . . . .	4
<b>2</b>	<b>Literature Survey</b>	<b>5</b>
2.1	Survey . . . . .	6
2.2	Comparative Study (Wireshark vs. Xtractr) . . . . .	6
2.2.1	Wireshark . . . . .	6
2.2.2	Xtractr . . . . .	7
<b>3</b>	<b>Proposed Work</b>	<b>8</b>
3.1	Problem Definition . . . . .	9
3.2	Features . . . . .	9
3.3	Scope . . . . .	9
3.4	Goals and Objectives . . . . .	10
3.5	Constraints . . . . .	10
<b>4</b>	<b>Research Methodology</b>	<b>11</b>
4.1	Steps to acquire and process . . . . .	12
4.2	Interpret inputs for projects . . . . .	13
4.3	Steps to carry out project work . . . . .	14
<b>5</b>	<b>Project Design</b>	<b>17</b>
5.1	Software Requirement Specifications . . . . .	18
5.1.1	Software Engineering . . . . .	18
5.1.2	Development Model . . . . .	19
5.1.3	Procedural Model . . . . .	20
5.1.4	Architectural Model . . . . .	20
5.2	System Overview . . . . .	21

5.2.1	Capture . . . . .	21
5.2.2	Indexing . . . . .	21
5.2.3	Process Pcaps . . . . .	22
5.2.4	Dashboard . . . . .	22
5.3	UML Diagrams . . . . .	23
5.3.1	Activity Diagram . . . . .	23
5.3.2	Data Flow Diagram (Level 0) . . . . .	24
5.3.3	Use Case Diagram . . . . .	25
5.4	Hardware and Software Requirements . . . . .	26
5.4.1	Software Requirements . . . . .	26
5.4.2	Hardware Requirements . . . . .	26
5.4.3	Technologies Used . . . . .	26
<b>6</b>	<b>Implementation</b>	<b>27</b>
6.1	Workflow . . . . .	28
6.1.1	Wireshark . . . . .	28
6.1.2	Xtractr . . . . .	28
6.1.3	Ruby and Xtractr Query Language . . . . .	29
6.1.4	D3.js . . . . .	29
6.1.5	Sinatra . . . . .	29
6.1.6	HAML . . . . .	30
6.2	Results and Snapshots . . . . .	31
6.2.1	Source Wise Bytes . . . . .	32
6.2.2	TCP Details . . . . .	33
6.2.3	Source Wise Duplicate Bytes . . . . .	34
6.2.4	Duplicate Packets According to the Time Frame . . . . .	35
6.2.5	Jitter Wise All Packets . . . . .	36
6.2.6	Flow-Wise Analysis . . . . .	37
6.3	Testing . . . . .	38
6.3.1	Manual Testing . . . . .	38
<b>7</b>	<b>Scheduling</b>	<b>39</b>
<b>8</b>	<b>Conclusion and Future Scope</b>	<b>41</b>
8.1	Conclusion . . . . .	42
8.2	Future Scope . . . . .	42
	<b>References</b>	<b>43</b>

# List of Figures

1.1	Setup of a Virtual Classroom . . . . .	3
4.1	Graph for jitter analysis . . . . .	12
4.2	Source wise duplicate bytes . . . . .	13
4.3	Stepwise procedure . . . . .	14
4.4	sample packet capture using Wireshark . . . . .	14
4.5	Xtractr user interface . . . . .	15
4.6	Sample ruby snippet . . . . .	15
4.7	Sample graphs and charts . . . . .	16
5.1	Incremental Model . . . . .	19
5.2	System Overview . . . . .	21
5.3	Activity Diagram . . . . .	23
5.4	Data Flow Diagram . . . . .	24
5.5	Use Case Diagram . . . . .	25
6.1	Workflow of Packet Capture Based Post Mortem Analysis of Realtime Multimedia Applications . . . . .	28
6.2	Protocol Distribution Pie . . . . .	31
6.3	Source-wise Byte Distribution . . . . .	32
6.4	TCP Details . . . . .	33
6.5	Source-wise Duplicate Packets . . . . .	34
6.6	Arrival of Duplicate Packets w.r.t Time Frame . . . . .	35
6.7	Jitter Analysis . . . . .	36
6.8	Flow-wise Throughput . . . . .	37
7.1	Schedule: Gantt Chart . . . . .	40

# List of Tables

6.1	Test Cases . . . . .	38
-----	----------------------	----



# Chapter 1

## Introduction

## 1.1 Introduction

The basic objective of the project is to do post-mortem analysis at packet-level. It involves detailed study of real-time multimedia applications. The main purpose is to gather relevant information related to packets like average unacknowledged data from client/server over a period of time, average rate of dropped/out of sync/retransmitted/re-acknowledged packets, find out the jitter in network, etc. for a given connection. Another measure is to do statistical analysis on all the data that has been gathered and present it in the form of charts for better understanding. These statistics direct towards identifying the usage of networks by each customer and the cost borne by the server to serve such clients and to dig-into the minute details found appropriate from business perspective. Finally a web based tool is designed to identify user issues and understand user experiences so as to either fix the infrastructure or code.

## 1.2 Motivation

With the emergence of virtualization, real-time multimedia applications are gaining more importance. More and more number of people is interested in facilities like desktop-sharing, live meeting, viewing pre-recorded presentations, etc. While dealing with real-time multimedia applications, the user experience becomes the major concern. The user should have a good experience while taking advantage of these amenities. Despite the availability of different tools for analyzing network traffic, there remains plenty of work to be done to retrieve the relevant information (statistics) efficiently. A clear stumbling block till date has been the absence of efficient tool suitable to identify the server-side issues faced during real-time multimedia applications and the solutions for them. Building such tools remains a high-risk enterprise when the exact requirements are unclear and the patterns unrecognized.

## 1.3 Background

In recent years, the demand for virtualization has increased largely. So the concept of virtual classrooms is gaining more and more popularity. Consider a scenario wherein a virtual class is being setup. A virtual class has many slaves connected to a master and the interaction can take place between the two in an on-line manner. The purpose of virtual classrooms

is to promote e-learning virtually with only the professor presenting and students getting hang of the things on-line just by interacting through the session. As shown in (Fig. 1.1), Amazon EC2 is the cloud that will provide the media boxes (M1, M2, M3) to setup parallel connections with one or more servers. In this case, a connection is considered to be established between M2 and the host server on which a particular session is held. Three nodes (slaves) are connected to host server out of which A and B belong to the same network.

All the three nodes are communicating with the host server through a dedicated link. Now, a small analysis is carried out on the session where the patterns followed by all the three slaves while communicating are noted and compared. Suppose suddenly B goes down while communicating and reestablishes connection with the server after an interval. Now the pattern followed by B in the communication would be somewhat different from the patterns obtained from A and C which had a normal communication. Similarly, if A and B are following an abnormal pattern compared to pattern followed by C then it can be said that there exists some problem in the network of A and B. So, the aim is to identify different patterns by considering different test cases and generate a report out of it which will help in determining the causes of bad experience and corresponding solutions to make it better.

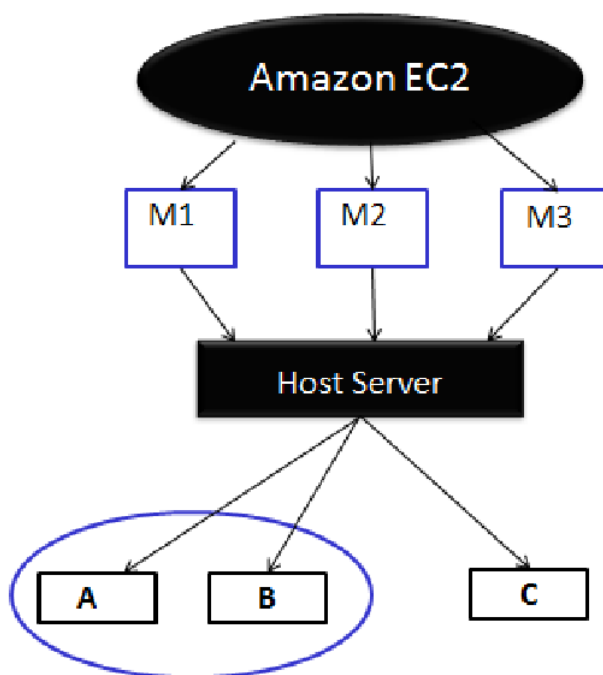


Figure 1.1: Setup of a Virtual Classroom

## 1.4 Need

During real-time multimedia, there can be lots of issues that a client faces like jitter, out-of-order packets, bandwidth-issues, RTT, etc. So, there is a possibility of bad user-experience. Since these issues are of concern for the network administrator, he should have knowledge about them. He should be able to identify these issues and give suggestions/ solutions for same. He should also be able to find average unacknowledged data from client as well as server over a period of time and have an idea about network conditions during a session. The proposed system focuses on all these issues and provides statistical analysis. These analytics could go into very high level or minute details as found appropriate from business perspective. Currently whatever tools are available for network forensics/ analysis deal with intrusion. There is no tool to find out the errors in network so as to enhance the user experience. This project is an attempt to build such a tool.

## Chapter 2

# Literature Survey

## 2.1 Survey

Network forensic tools are used to analyze the network and find out the anomalies. But, presently all the tools available deal with the Intrusion Detection Systems which are designed to identify the system activities for malicious activities or policy violations and produces reports to a management station. So, as of now, there is no such tool available which will provide statistics about the abnormal behavior in the network. The focus of this project is on statistical analysis because numbers dont make as much sense as the charts or graphs do.

So, the tool designed is not meant for detecting the intrusions in the system but to help the system to enhance the user experience by improving the network performance with the help of analyzing the patterns that are being followed between the communicating parties and the abnormal patterns that are causing the problems. So the main interest is in the abnormalities and the defective or damaged packets in the communication. The tool does the job of studying these erroneous packets to make inference and draw conclusions.

## 2.2 Comparative Study (Wireshark vs. Xtractr)

### 2.2.1 Wireshark

Wireshark is a powerful and irreplaceable tool used as a packet sniffer. The tool will capture the packets according to the arrival time of each packet. It is as good as a container where one after another all packets are dumped and then finding out a particular packet of interest becomes quite tedious. If real time multimedia applications are considered, then time frame becomes the most important factor but Wireshark cant filter the packets based on the time frame. It has very basic filters. So, if access to a particular bit of a particular packet arriving in a particular time frame is required, it becomes almost an impossible job if Wireshark is used.

Advantages:-

- Expert Protocol Analyst
- 850+ supported protocols

Disadvantages:-

- Poor at providing visual statistics

- No support for flow-wise data
- Information is available but cant be extracted and used
- Limited filters

### 2.2.2 Xtractr

Xtractr is an open source tool that eases the job of Wireshark and provides some additional functionalities. It sorts the packets according to the flows which are the links between two unique communicating parties. It has an enriched set of queries which can be used to reach to any field of the packet. Xtractr also has an online version called pcapr-local which supports parallelism. Many packets can be simultaneously dumped and the operations can be carried out on each one of them independently. Xtractr sorts according to the timestamps which certainly helps in finding out only the defective time frames[3,5]. Advantages:-

- Sorting of packets based on flows and timestamps
- Rich query set
- User interface for higher interactivity
- Bit level manipulation possible

Disadvantages:-

- Pattern and statistical analysis are difficult
- Parallelism not supported by Xtractr. Pcapr-local is an alternative

# Chapter 3

## Proposed Work



### 3.1 Problem Definition

To do post mortem analysis at packet level for kpoint in the scenarios such as live meeting or kapsule viewing that is watching pre-recorded videos. So the basic aim is to find the signs of abnormal behavior in the network which includes jitter, defective packets(out of order, retransmitted, lost),throughput and make sense out of all the data gathered to pin point the particular packet originating from the particular source. So, here packets are being manipulated up to bit level and different patterns that the packets follow in the conversation are studied. Finally based on the results, statistical analysis is carried out and the results are represented in terms of graphs/charts on a dashboard which is monitored by the network administrator to track the activities taken place in a day's time.

### 3.2 Features

- Efficient tool for resolving issues in the network.
- Capable of manipulating the packet at bit level.
- Statistical Analysis for better understanding.
- Historical Analysis to do comparative study and find out the persistent errors over a period of time.
- Interactive Dashboard- All information one place.

### 3.3 Scope

- Virtualization is evolving day by day. As a result, the number of users using virtual classrooms is increasing and so is the number of issues. So, a tool is very much necessary which helps in identifying the issues and resolving them with the help of a network administrator.
- The project will help in enhancing the user experience by resolving his issues and suggesting the remedies for the same. It is being planned to make the tool online and more interactive in future.
- In future, optimization of the packet capture activity can be done to provide least overhead which will involve kernel profiling and impact

of high-speed data capture on virtualized environment like Amazon EC2 as well as physical servers.

- In future, optimization of the packet capture activity and the storage of the captures can be done by means of an unstructured database.
- Least overhead which will involve kernel profiling and impact of high-speed data capture on virtualized environment like Amazon EC2 as well as physical servers.

### **3.4 Goals and Objectives**

I. Detailed study of Protocols(TCP,HTTP,XMPP,RTMP)

II. Enhance User Experience to Network Flow

III. Identify Signs of slow/fast server and client response

IV. Jitter Analysis and quantify the same.

V. Determine the usage of Network Resources and the cost associated.

VI. All Information One Place

### **3.5 Constraints**

- The administrator has to be a technical person who is very well aware of the network flow and basics of statistical studies.
- The tool works offline on the captured data.
- Deriving conclusions from the statistical analysis requires detailed knowledge about the network flow and patterns.

# Chapter 4

## Research Methodology

## 4.1 Steps to acquire and process

The basic purpose of the project is to gather relevant information related to packets and find following kind of information:

- Average unacknowledged data size from client (and server) over a period of time (Sign of client (and server) not responding fast enough) for a given connection.
- Average rate of dropped or out of sync packets (Gives idea about network condition at the time of class) for a given connection
- Jitter in the network.
- Identifying the usage of network resources by each customer and cost kPoint is bearing to serve such clients.

In order to obtain these analytics, different pcaps have been captured during live-meeting/ capsule viewing and through queries extracted the relevant information. With the help of this information, different graphs and charts have been plotted a comparative study is done.

Some of the graphs obtained during the procedure are shown below:

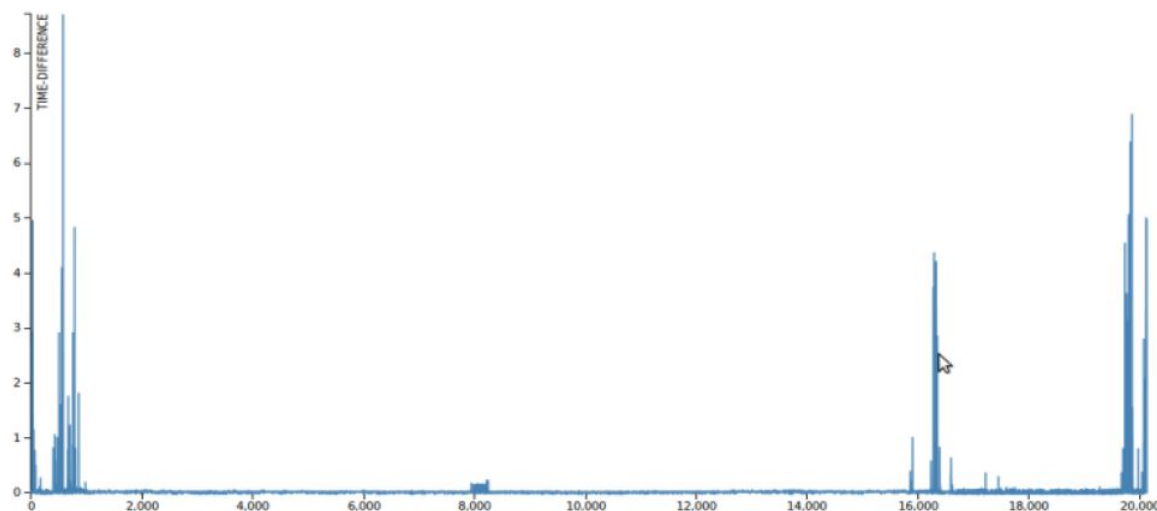


Figure 4.1: Graph for jitter analysis

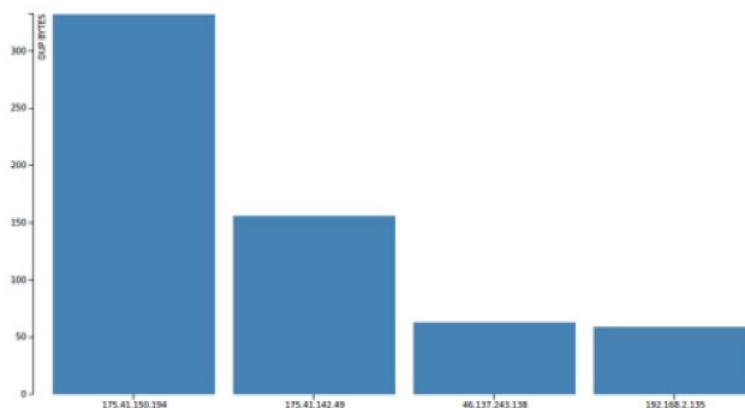


Figure 4.2: Source wise duplicate bytes

## 4.2 Interpret inputs for projects

Statistics-deriving part is research kind of activity. During the process of deriving the statistics, sample pcaps were captured in different networks like wireless network, dongle-connection, etc and compared the patterns obtained. Also, the same capture was viewed over and over under different conditions and compared the statistics thus derived. These statistics give the conclusion about the network condition during a particular time.

During the entire process, very less of programming was involved. However it involved a very close study of TCP/IP/HTTP/RTMP/XMPP protocols to get insights about how real time protocol works and co-relate the user experience to the network traffic flows.

### 4.3 Steps to carry out project work

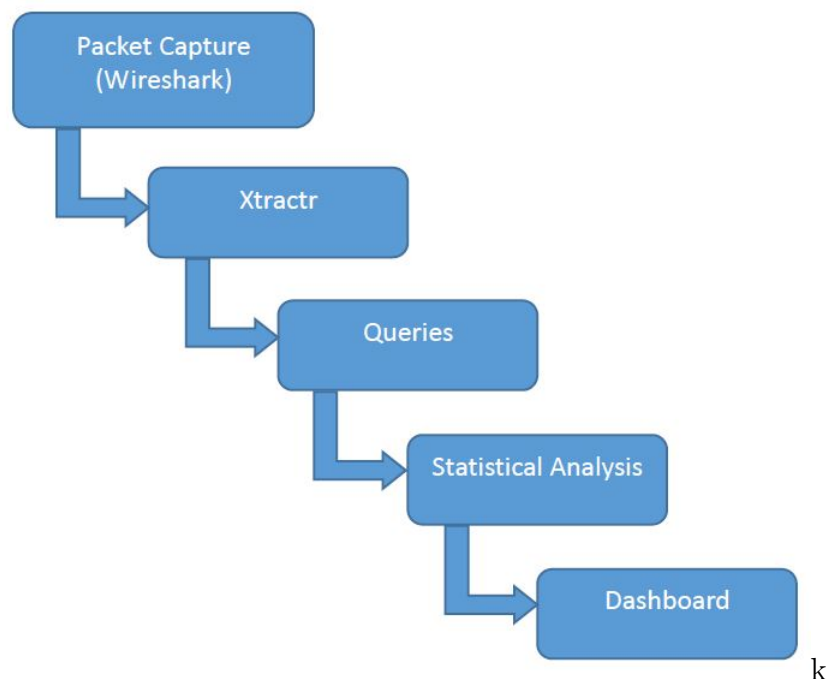


Figure 4.3: Stepwise procedure

Figure 4.3 shows the basic steps carried out to design our dashboard. These steps are explained below:

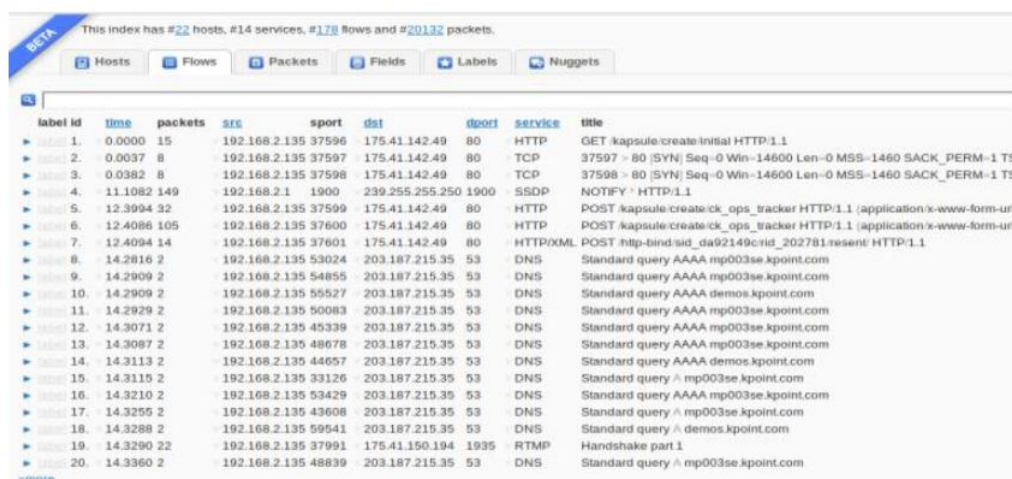
1. Different sample pcaps were captured with the help of Wireshark. These pcaps were of different scenarios like live-meeting, capsule-viewing, desktop sharing, etc. A sample pcap is as shown below:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.16.171	192.168.16.253	HTTP	55	Continuation or non-HTTP traffic
2	0.003955	192.168.16.253	192.168.16.171	TCP	66	rdl-aas > 50003 [ACK] Seq=1 Ack=2 Win=1100 Len=0 SLE=1 SRE=2
3	0.080982	192.168.16.171	192.168.16.253	HTTP	55	Continuation or non-HTTP traffic
4	0.084016	192.168.16.253	192.168.16.171	TCP	66	rdl-aas > 50428 [ACK] Seq=1 Ack=2 Win=593 Len=0 SLE=1 SRE=2
5	2.416486	192.168.16.171	192.168.16.253	HTTP	970	GET http://www.google.co.in/s?hl=en&sugexp=les%3B&q&s_nf=1&cp=1
6	2.419144	192.168.16.253	192.168.16.171	TCP	60	rdl-aas > 50428 [ACK] Seq=1 Ack=918 Win=730 Len=0
7	2.597929	192.168.16.253	192.168.16.171	TCP	933	[TCP segment of a reassembled PDU]

Figure 4.4: sample packet capture using Wireshark

2. The next step was to extract the relevant information out of these data available. For this, an open source tool called Xtractr has been used. Xtractr has an interface as shown below.[3]:

page



The screenshot shows the Xtractr user interface with a table of network traffic data. The table has columns for label id, time, packets, src, sport, dst, dport, service, and title. The data includes various protocols like HTTP, TCP, DNS, and RTMP.

label id	time	packets	src	sport	dst	dport	service	title
1	0.0000	15	192.168.2.135	37596	175.41.142.49	80	HTTP	GET /kapsule/create-initial HTTP/1.1
2	0.0037	8	192.168.2.135	37597	175.41.142.49	80	TCP	37597 → 80 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TS
3	0.0382	8	192.168.2.135	37598	175.41.142.49	80	TCP	37598 → 80 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TS
4	11.1082	149	192.168.2.1	1900	239.255.255.250	1900	SSDP	NOTIFY * HTTP/1.1
5	12.3994	32	192.168.2.135	37599	175.41.142.49	80	HTTP	POST /kapsule/create/ck_ops_tracker HTTP/1.1 (application/x-www-form-ur
6	12.4086	105	192.168.2.135	37600	175.41.142.49	80	HTTP	POST /kapsule/create/ck_ops_tracker HTTP/1.1 (application/x-www-form-ur
7	12.4094	14	192.168.2.135	37601	175.41.142.49	80	HTTP/XML	POST http-bind/sid_da92149c/rid_202781.resent HTTP/1.1
8	14.2816	2	192.168.2.135	53024	203.187.215.35	53	DNS	Standard query AAAA mp003se.kpoint.com
9	14.2909	2	192.168.2.135	54855	203.187.215.35	53	DNS	Standard query AAAA mp003se.kpoint.com
10	14.2909	2	192.168.2.135	55527	203.187.215.35	53	DNS	Standard query AAAA demos.kpoint.com
11	14.2929	2	192.168.2.135	50083	203.187.215.35	53	DNS	Standard query AAAA mp003se.kpoint.com
12	14.3071	2	192.168.2.135	45339	203.187.215.35	53	DNS	Standard query AAAA mp003se.kpoint.com
13	14.3087	2	192.168.2.135	48678	203.187.215.35	53	DNS	Standard query AAAA mp003se.kpoint.com
14	14.3113	2	192.168.2.135	44657	203.187.215.35	53	DNS	Standard query AAAA demos.kpoint.com
15	14.3115	2	192.168.2.135	33126	203.187.215.35	53	DNS	Standard query A mp003se.kpoint.com
16	14.3210	2	192.168.2.135	53429	203.187.215.35	53	DNS	Standard query AAAA mp003se.kpoint.com
17	14.3255	2	192.168.2.135	43608	203.187.215.35	53	DNS	Standard query A mp003se.kpoint.com
18	14.3288	2	192.168.2.135	59541	203.187.215.35	53	DNS	Standard query A demos.kpoint.com
19	14.3290	22	192.168.2.135	37991	175.41.150.194	1935	RTMP	Handshake part 1
20	14.3360	2	192.168.2.135	48839	203.187.215.35	53	DNS	Standard query A mp003se.kpoint.com

Figure 4.5: Xtractr user interface

3. The exact information was retrieved using the Xtractr query language[4]. This query language has been embedded in a ruby code. Sample code snippet is as follows:

```
#!/usr/bin/env ruby
Require '/home/bepro/xtractr/mu/xtractr'
xtractr=Mu::Xtractr. New
print "TIME\tID\n"
```

Figure 4.6: Sample ruby snippet

4.Finally, all the information has been embedded with the help of Sinatra web framework and presented in form of graphs and charts on a dashboard.[6]

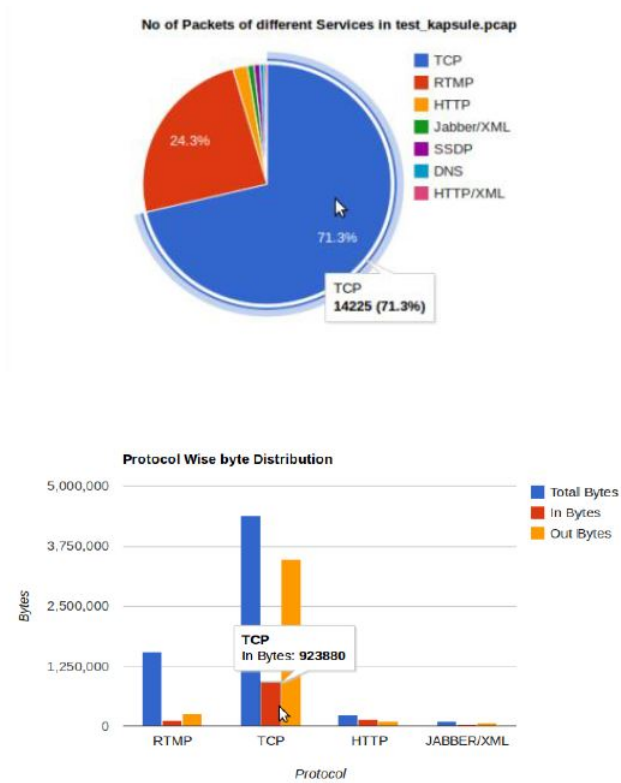


Figure 4.7: Sample graphs and charts



# Chapter 5

## Project Design

## 5.1 Software Requirement Specifications

This Software Requirements Specification provides a complete description of all the functions and specifications of the Real time trainer system. The purpose of this document is to explain the purpose and features of the Real time trainer system, the interfaces of the system, what the system will do the constraints under which it must operate and how the system will react to external stimuli.

### 5.1.1 Software Engineering

The purpose of System Design is to create a technical solution that satisfies the functional requirements for the system. At this point in the project lifecycle there should be a Functional Specification, written primarily in business terminology, containing a complete description of the operational needs of the various organizational entities that will use the new system. The challenge is to translate all of this information into Technical Specifications that accurately describe the design of the system, and that can be used as input to System Construction.

The Functional Specification produced during System Requirements Analysis is transformed into a physical architecture. System components are distributed across the physical architecture, usable interfaces are designed and prototyped, and Technical Specifications are created for the Application Developers, enabling them to build and test the system. Many organizations look at System Design primarily as the preparation of the system component specifications; however, constructing the various system components is only one of a set of major steps in successfully building a system. The preparation of the environment needed to build the system, the testing of the system, and the migration and preparation of the data that will ultimately be used by the system are equally important. In addition to designing the technical solution, System Design is the time to initiate focused planning efforts for both the testing and data preparation activities.

### 5.1.2 Development Model

Incremental model: Incremental model has been used to develop the project. The incremental model is as follows: The incremental model delivers a series of releases called increments that provide progressively more functionality for the customer as each increment is delivered. When an incremental model is used, the first increment is often a core product that is, basic requirements are addressed, but many supplementary features remain undelivered. The core product is used by the customer.

As a result of use and/or evaluation a plan is developed for the next increment. The plan addresses the modification of the core product to better meet the needs of the customer and the delivery of additional features and functionality. This process is repeated following the delivery of each increment, until the complete product is produced.

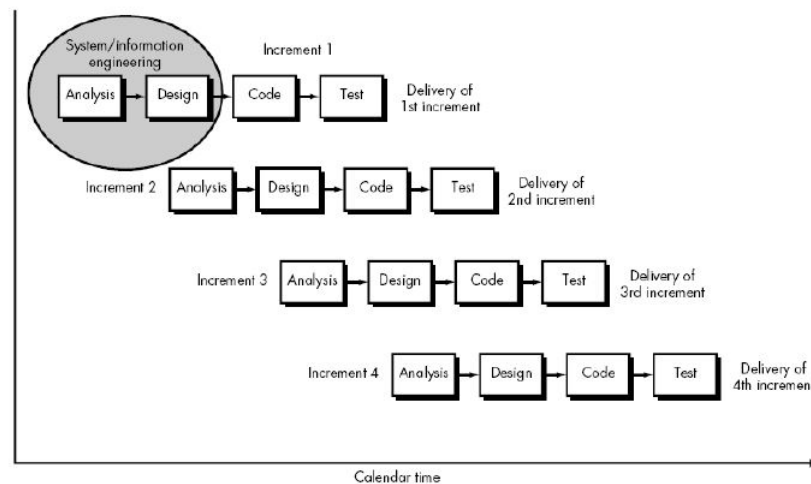


Figure 5.1: Incremental Model

Figure 5.1 basically correlates to Incremental model has been used to develop the dashboard of project. Each graph of the project has been plotted step-wise.

Increment 1 contains a basic pie-chart was plotted that gives the distribution of protocols. As more and more requirements became known, corresponding graphs and charts evolved.

The errors faced by the server were brought to the notice and graphs showing these errors were developed. Thus, with the help of customer requirements and efficient analysis, various statistics were produced and different functionalities addressed.

Increment 2 contains graphs that include jitter-analysis, TCP errors (dropped/out-of-sync/retransmitted), throughput and many more.

Increment 3 basically contains the actual cause of the problem. It includes source and destination IP addresses of the flawed packets.

### 5.1.3 Procedural Model

The procedural design describes structured programming concepts using graphical, tabular and textual notations. These design media enable the designer to represent procedural detail that facilitates translation to code. This blueprint for implementation forms the basis for all subsequent software engineering work.

### 5.1.4 Architectural Model

It is used in establishing the overall structure of a software system. The design process for identifying the sub-systems making up a system and the framework for sub-system control and communication is architectural design. The output of this design process is a description of the software architecture. An early stage of the system design process. Represents the link between specification and design processes. Often carried out in parallel with some specification activities. It involves identifying major system components and their communications.

- System Structuring - The system is decomposed into several principal subsystems and communications between these sub-systems are identified.
- Control Modelling - A model of the control relationships between the different parts of the system is established.
- Modular Decomposition The identified sub-systems are decomposed into modules.

## 5.2 System Overview

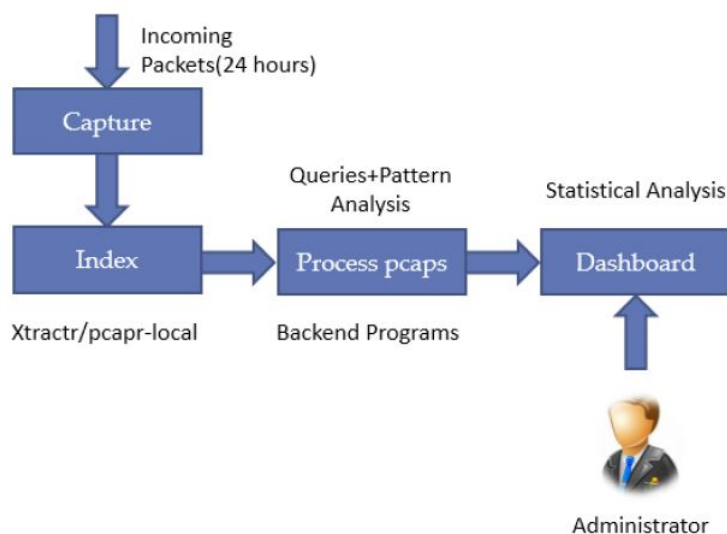


Figure 5.2: System Overview

### 5.2.1 Capture

Wireshark is basically a free and open-source packet analyzer tool used for live capturing of data from the network. Wireshark is very similar to tcpdump. It allows the user to put network interface controllers that support promiscuous mode into that mode, in order to see all traffic visible on that interface, not just traffic addressed to one of the interface's configured addresses and broadcast/multicast traffic. Wireshark captures data from the live wire and separate them into packets. The output of the Wireshark is a \*.pcap file.

### 5.2.2 Indexing

Indexing is done after capturing data from live wire with the help of Xtractr. Xtractr app that does three-way reconciliation of your pcaps, the Xtractr indexes and CouchDB similar to what Picasa does to images or iTunes does to your music collection. All the pcaps can be simply dumped into the configured pcaps directory and pcapr.local will index these things in the background and stick all the meta-data in CouchDB so it can be organized in the collection of pcaps and collaborated internally. Its also fully RESTful so various intra-pcap analysis can be run and get whatever is need. It also maintain pcaps according to the flows.

### 5.2.3 Process Pcaps

Processing pcaps basically includes indexing and extracting information from Xtractr. It uses Xtractr query language for fetching the required information. Xtractr uses Ferret, in combination with SQLite to store all the data extracted from the pcaps. In some cases Ferret query strings are mapped to SQL statements internally so that the externally visible language is always the powerful query language. So the basis for querying is fields and terms. In addition to all the tshark fields, Xtractr adds its own[3].

### 5.2.4 Dashboard

Dashboard is basically designed in Sinatra (web framework). The output of Xtractr query language consists of data. The data is mostly in the form Comma Separated Values (CSV) or Tabbed Separated values (TSV). This is given as an input to D3.JS for further processing. It uses HAML for accessing the web pages and displaying the graphs and charts.

## 5.3 UML Diagrams

### 5.3.1 Activity Diagram

This represents the workflow of the system; if a particular step executes, where does the system go next and in cases where conditional statements exist, the different flows are generated. It also describes flows independent of each other which can run in parallel.

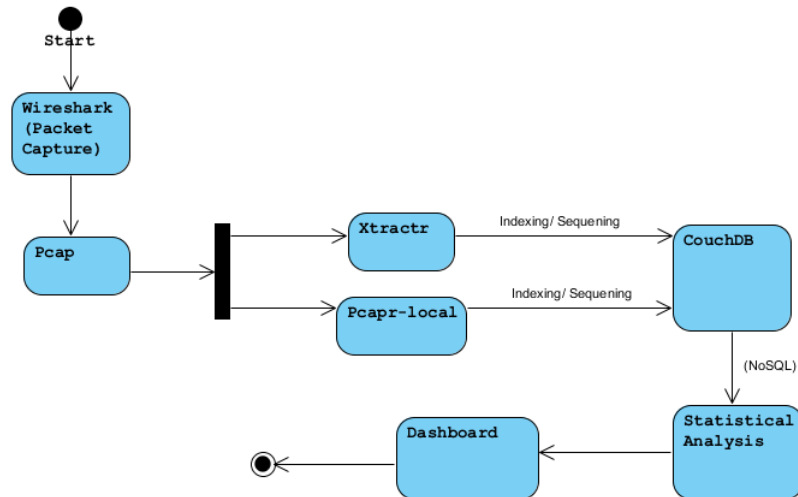


Figure 5.3: Activity Diagram

### 5.3.2 Data Flow Diagram (Level 0)

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

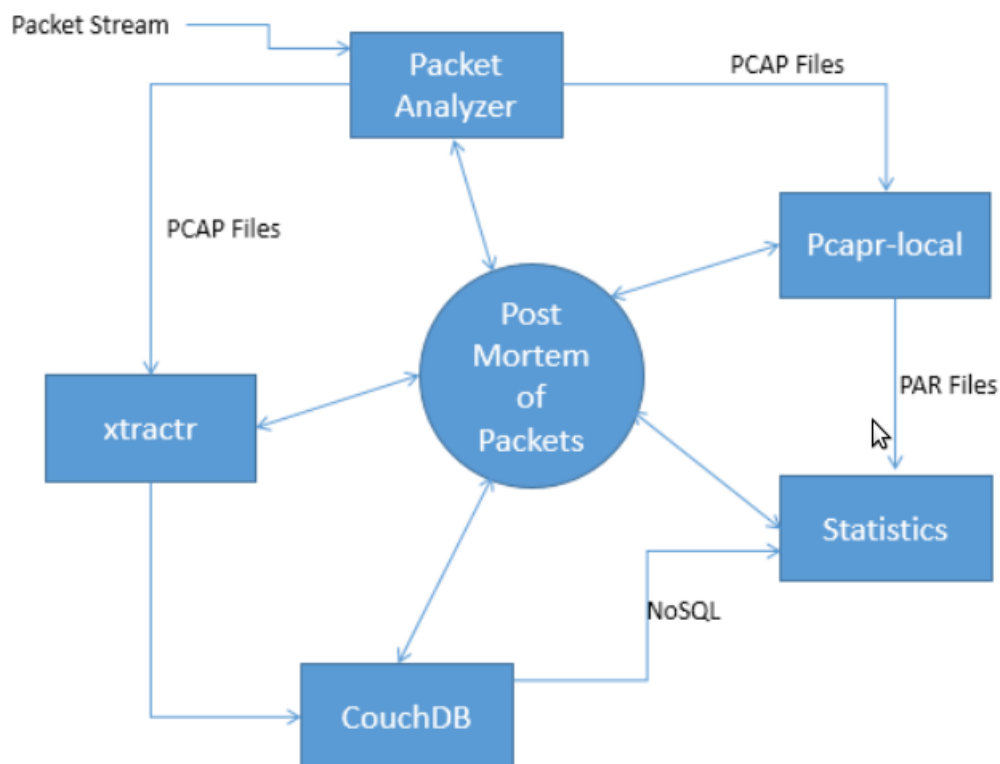


Figure 5.4: Data Flow Diagram



### 5.3.3 Use Case Diagram

Defines the actors which interact with the system, it also contains the various operations that can be performed by the system.

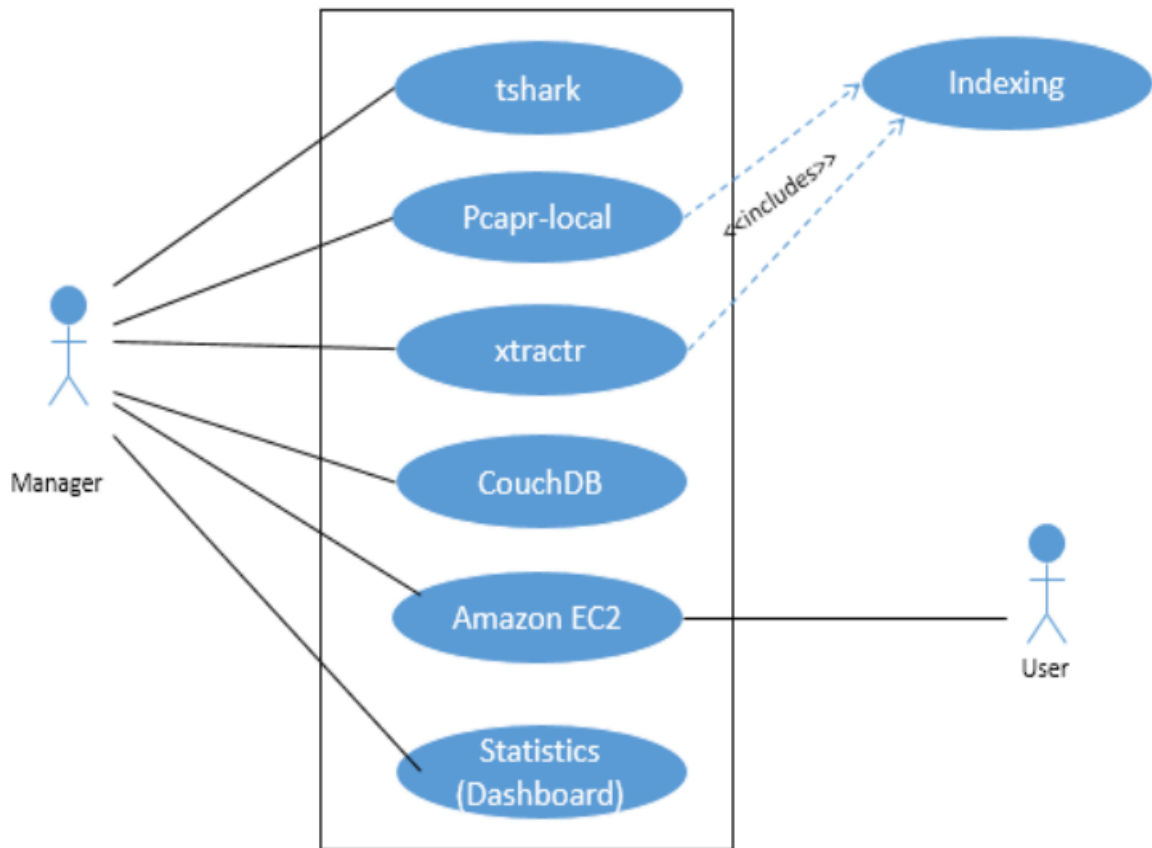


Figure 5.5: Use Case Diagram

## **5.4 Hardware and Software Requirements**

### **5.4.1 Software Requirements**

- Wireshark (Ver.1.2.1)
- Xtractr (Ver.2.0)
- Linux OS (Ubuntu 12.04)
- Ruby Sinatra Framework (Ver.1.4.2)

### **5.4.2 Hardware Requirements**

- Network Connection

### **5.4.3 Technologies Used**

- Ruby
- Javascript
- HAML

# Chapter 6

## Implementation

## 6.1 Workflow

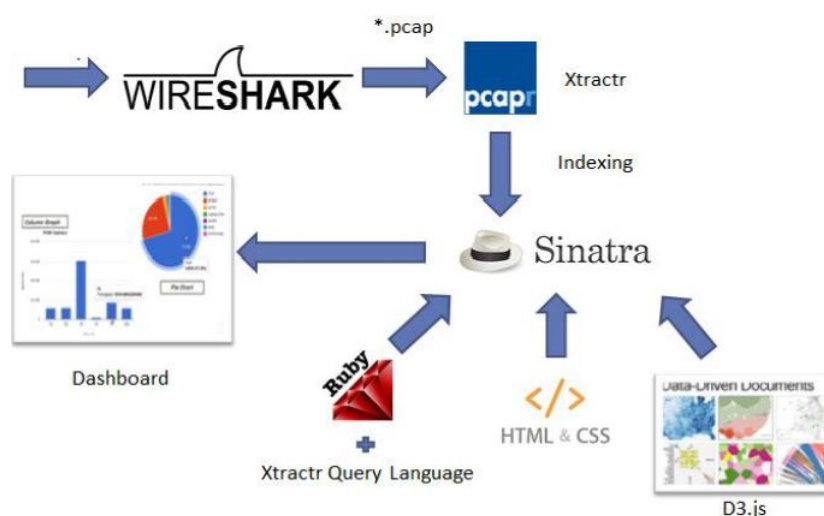


Figure 6.1: Workflow of Packet Capture Based Post Mortem Analysis of Realtime Multimedia Applications

### 6.1.1 Wireshark

The system starts with packet capture using network analyzers like Wireshark. The captured packets are stored in a .pcap file. The project considers the capture of traffic flowing through the network in a days time.

### 6.1.2 Xtractr

Xtractr is an online version of pcapr.local. It is a hybrid cloud application for indexing, searching, reporting, extracting and collaborating on pcaps. Its indexing property enables to rapidly identify field issues and perform network forensics and troubleshooting with just a few clicks. The lite version of Xtractr can index up to 10 million packets or 1 Gbyte of pcaps. The crux of this project is Xtractr. It is used to index the pcaps so as to determine where in network the problem persists and also to pinpoint a particular piece of information. This information helps in identifying the patterns and aids the server to find out the issue in the network.

The three common use-cases for Xtractr include:

1. Network Troubleshooting - Xtractr enables engineers to troubleshoot network issues by classifying massive pcaps, and extracting key flows from that data to create a concise report of the underlying issue.
2. Network Problem Isolation and Field Resolution - Xtractr enables engineers and testers to quickly troubleshoot and isolate field issues.

3. Network Forensics - Xtractr simplifies the network forensics process by pulling relevant data packets from massive packet captures, enabling engineers to create reports quickly and easily.

### 6.1.3 Ruby and Xtractr Query Language

The tool Xtractr comes with its query language known as Xtractr Query Language with the help of which the packets can be accessed up to the bit level and only the precise information of interest can be extracted. Xtractr uses Ferret, in combination with SQLite to store all the data extracted from the pcaps.

These queries are embedded in a ruby program. The reason for using Ruby as a language is, it provides a built in library pcap with the help of which the various fields of the packets can be accessed programmatically and the Sinatra web framework also deals with ruby syntax for its functioning.

### 6.1.4 D3.js

D3.js is a JavaScript library for manipulating documents based on data. D3 helps to bring data to life using HTML, SVG and CSS. This library is used for creating different graphs and charts. These graphs, made with the help of data acquired from the ruby programs, give the statistical details of the network such as in-bytes, out-bytes, throughput, etc. With the help of these statistics, it is possible to find out the jitter in the network and other related issues. These graphs are the main contents of the final portal which helps to perform the network forensics at a glance.

### 6.1.5 Sinatra

Sinatra is a web framework that allows to provide a web interface to our project. Using it, the final portal has been designed which shows all the necessary details through pie-charts, line-charts, bar-graphs, etc. Going through these charts, the different parameters of network like jitter can be identified. The ruby codes, java script files and style sheets have been embedded into the Sinatra framework and the corresponding procedures have been invoked.

### 6.1.6 HAML

Sinatra uses HAMLs instead of HTMLs for its rendering. HAML (HTML Abstraction Markup Language) is a lightweight markup language that is used to describe the XHTML of any web document without the use of traditional inline coding. Its designed to address many of the flaws in traditional templating engines, as well as making markup as elegant as it can be. However, Haml avoids the need for explicitly coding XHTML into the template, because it is itself a description of the XHTML, with some code to generate dynamic content.

## 6.2 Results and Snapshots

### Kpoint Statistics

#### Protocol Distribution

- Packet wise Details
- Flow wise Details
- Byte wise Details
- TCP Details

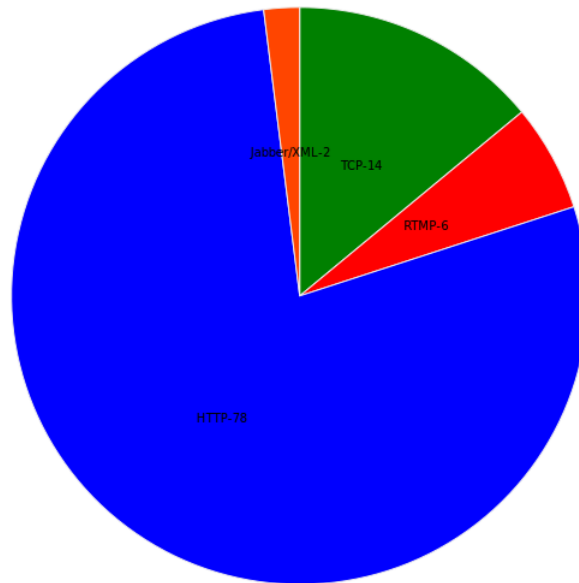


Figure 6.2: Protocol Distribution Pie

Fig.6.2 shows the protocol distribution and the packets of each protocol in the particular capture on which the statistics are designed. In the background, the ruby code for corresponding output and the java script for drawing the pie is executed. Sinatra framework serves as a localhost and is functional on port number 4567.

### 6.2.1 Source Wise Bytes

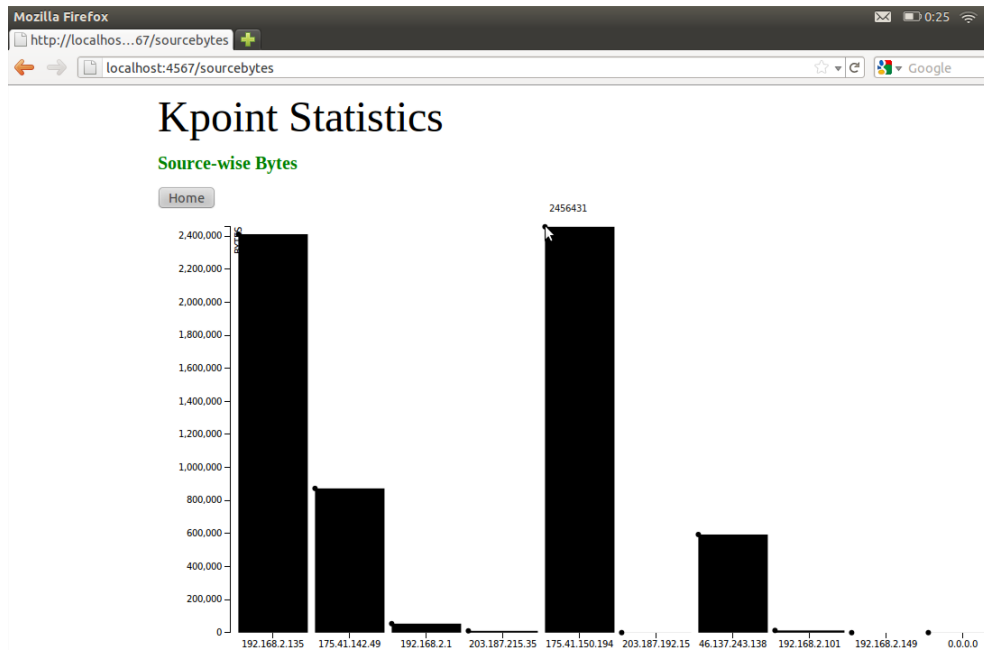


Figure 6.3: Source-wise Byte Distribution

Fig.6.3 shows the bytes transmitted by each source in a particular capture. The tooltip shows that from source IP 175.41.150.194, maximum numbers of bytes i.e., 2456431 bytes are transmitted, so the focus would be on that particular host and the details of that particular source can be analysed.



## 6.2.2 TCP Details

### Kpoint Statistics

#### TCP Details

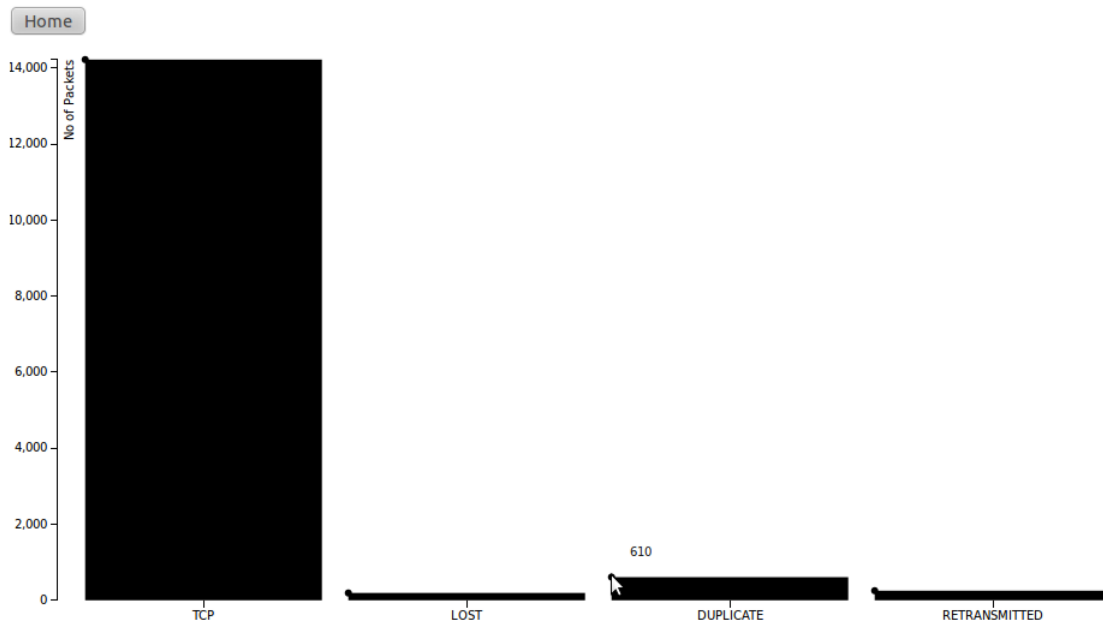


Figure 6.4: TCP Details

Fig.6.4 shows the TCP details of a capture which has the total number of TCP packets and Lost, Duplicate, Retransmitted packets as well. As only the erroneous packets are of interest, TCP defective packets are looked at. As shown in the figure, numbers of duplicate packets are maximum, so now the origin of these duplicate packets become a concern to the administrator.

### 6.2.3 Source Wise Duplicate Bytes

Source-wise Duplicate Packets

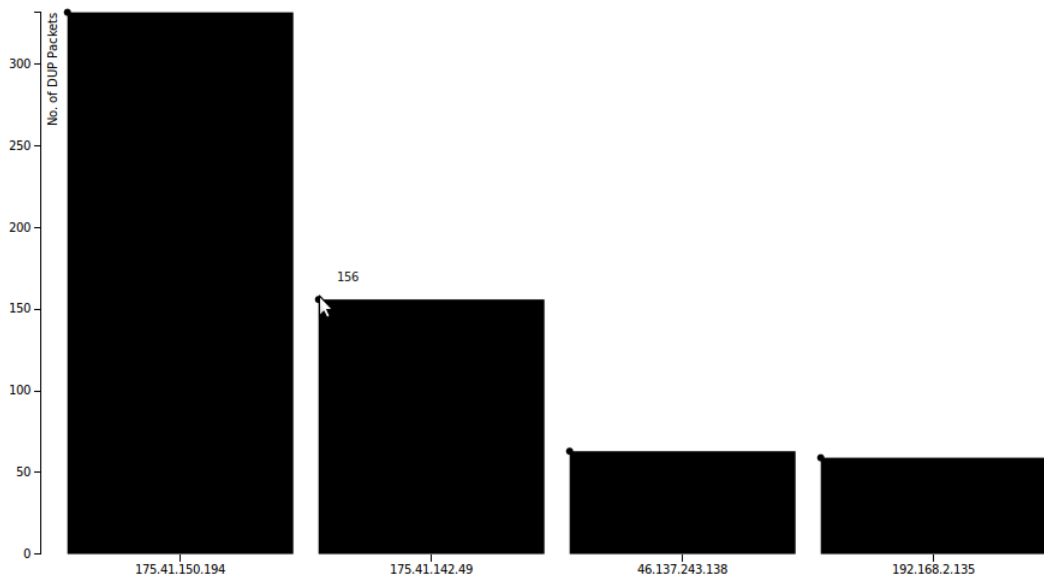


Figure 6.5: Source-wise Duplicate Packets

Fig.6.5 shows the duplicate packets sent from various sources. From the Fig.6.3, it has been found out that host 175.41.150.194 was the most active user as it had transmitted the maximum number of bytes. But, now from this figure, it can be inferred that maximum numbers of packets sent from that source are duplicate and that's where the focus shifts now as this is the source which might have caused problems in the network.

## 6.2.4 Duplicate Packets According to the Time Frame

### Duplicate Packets According to the Time Frame

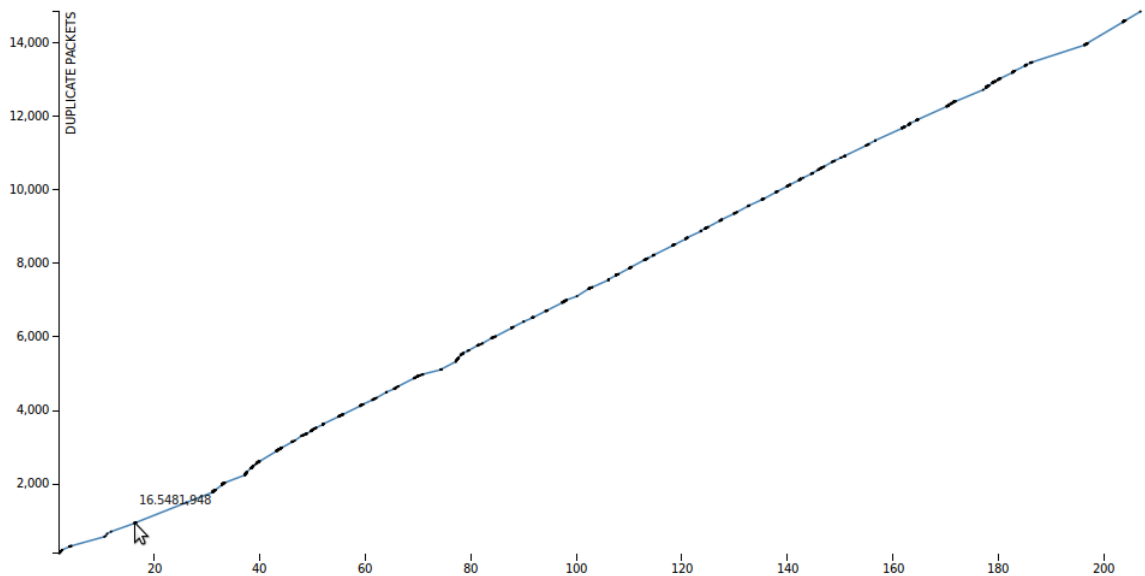


Figure 6.6: Arrival of Duplicate Packets w.r.t Time Frame

Fig.6.6 shows the arrival time frame of all the duplicate packets in a particular capture. This statistics will help to focus only on that time frame in which the most number of duplicate packets arrived. In the figure above, the congestion of black dots signifies that the frequency of occurrences of duplicate packets was very high and in that time frame, the network didnt work fine.

## 6.2.5 Jitter Wise All Packets

### Jitter Between All Packets

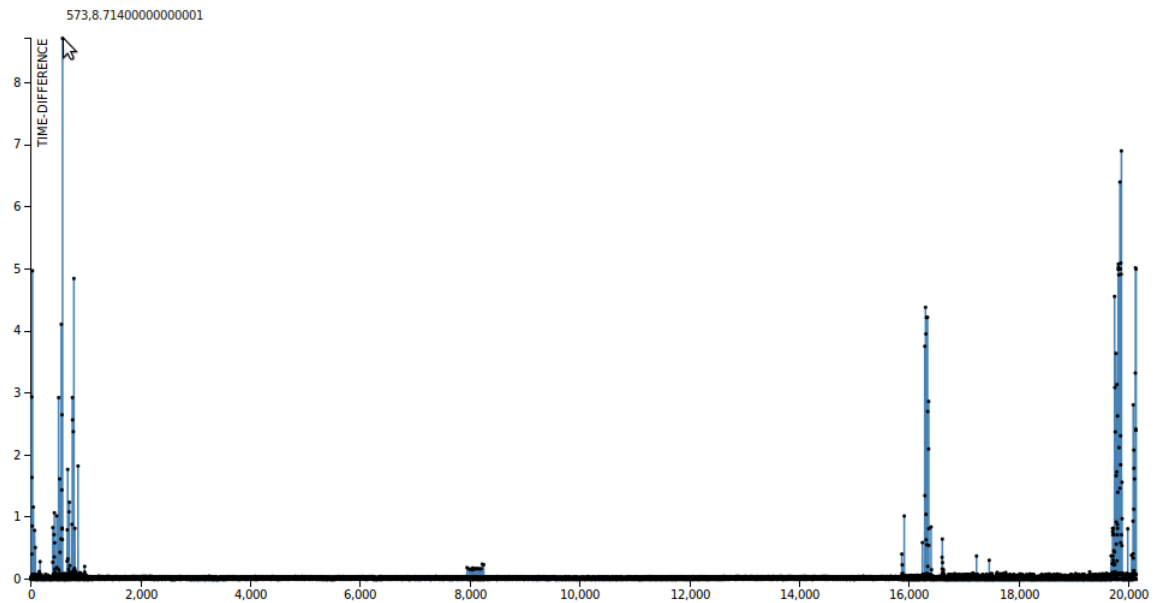


Figure 6.7: Jitter Analysis

In real time multimedia applications, inter packet delay i.e., jitter is the major concern and it has to be minimum for proper communication. Fig.6.7 shows the jitter rose during the communication. With the help of tooltip, pinpointing the particular packets between which the delay was maximum becomes possible and then those two packets can be analyzed from a stack of millions of packets.

## 6.2.6 Flow-Wise Analysis

### Flow-wise Analysis

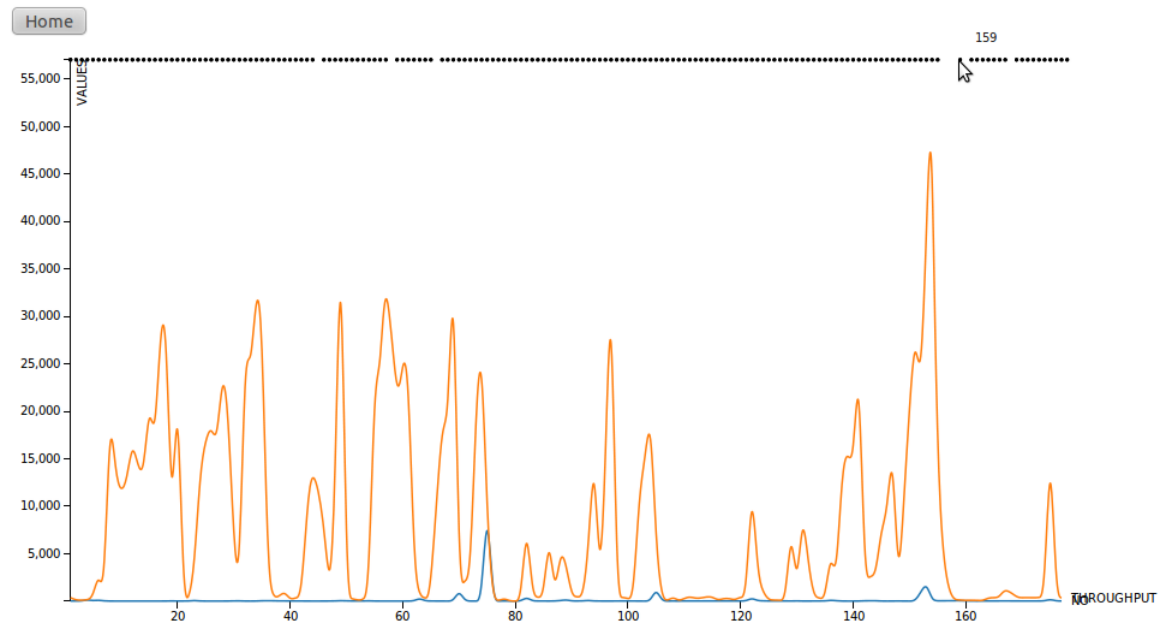


Figure 6.8: Flow-wise Throughput

Fig.6.8 shows the flow-wise throughput i.e., bytes/sec. This figure will be helpful in identifying the flows where there is less number of packets but more throughput and vice-versa. The flows which follow a normal pattern can be distinguished from those which show variations.

## 6.3 Testing

### 6.3.1 Manual Testing

Manual testing is the process of manually testing software for defects. The system has been tested manually for any error. The testing has been done at each step right from working of Xtractr to the final dashboard. To ensure completeness of testing a set of important test cases has been prepared. All these test cases are dynamic. These test cases have been shown below:

Table 6.1: Test Cases

Sr.No	Test Steps	Test Data	Expected Output	Actual Output	Status
1	Browse a pcap after indexing it	Sample pcap	Browsed Successfully	Browsed Successfully	PASS
2	Browse another pcap when one is already being browsed	2 pcaps	localhost already in use	localhost already in use	PASS
3	Make 5 jitter graphs on different captures of same capsule	5 pcaps	Graphs should be similar	Similar graphs observed	PASS
4	Open localhost without running sinatra	localhost port	Sinatra cannot find this	Sinatra cannot find this	PASS
5	Click on a button in GUI	Dashboard	The relevant graph should be displayed	The relevant graph is displayed	PASS
6	Input incorrect tsv to the html code	HTML file	Graph obtained is not as expected	Graph obtained is not as expected	PASS
7	Input correct tsv to the html code	HTML file	Graph obtained is as expected	Graph obtained is as expected	PASS
8	Showing tool-tip on graphs	Dashboard	The tool-tip should show the correct x-values and y-values	Correct x and y values	PASS

# Chapter 7

## Scheduling

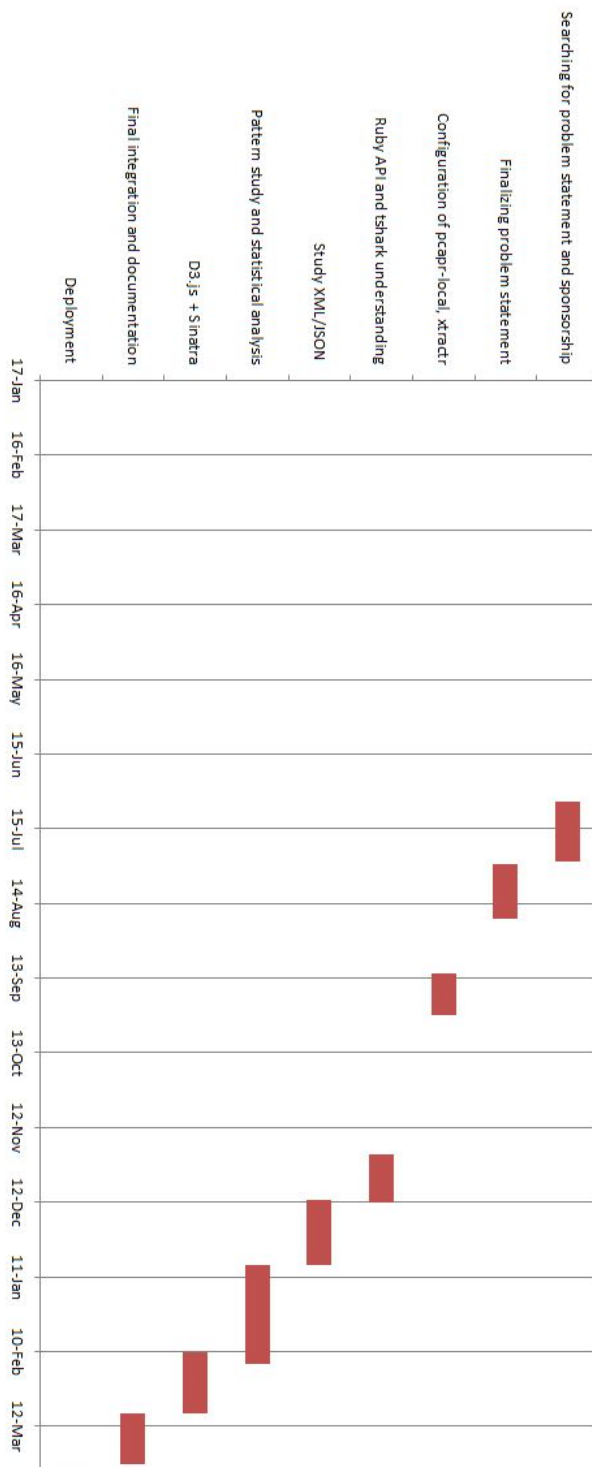


Figure 7.1: Schedule: Gantt Chart



# Chapter 8

## Conclusion and Future Scope

## 8.1 Conclusion

Despite the availability of different tools for analyzing network traffic, there remains plenty of work to be done to retrieve the relevant information (statistics) efficiently. A clear stumbling block till date has been the absence of efficient tool suitable to identify the server-side issues faced during real-time multimedia applications and the solutions for them. Building such tools remains a high-risk enterprise when the exact requirements are unclear and the patterns unrecognized. The work presented in this project is an account of an attempt to build this tool. The work on real-time multimedia applications has been described, discussing the experiences in analyzing the patterns and building out the related statistics, as well as some of the tools built and techniques developed over the years to build such tool more efficiently. Some of the initial experiences with the use of ruby queries, d3.js and Sinatra in finding out these statistics have also been described. Finally, some of the results of the experiments carried out during the entire statistical analysis process have been presented.

## 8.2 Future Scope

In the future, we plan to optimize the packet capture and its storage. Least overhead which will involve kernel profiling and impact of high-speed data capture on virtualized environment like Amazon EC2 as well as physical servers. Historical Analysis can also be added to the scope as it deals with identifying and comparing patterns over a period of time. We also plan to make the tool dynamic and more interactive as for now the tool is offline.

# Bibliography

- [1] Terry Nelms and Mustaque Ahamad, "Packet Scheduling for Deep Packet Inspection on Multi-Core Architectures", IEEE VOL. 40, NO. 2, APRIL 2011, Georgia Institute of Technology, USA.
- [2] <https://github.com/obfcurity/descartes>
- [3] <http://www.pcapr.net>
- [4] <http://blog.mudynamics>
- [5] <https://github.com/pcapr-local>
- [6] <http://net.tutsplus.com/tutorials/ruby/singing-with-sinatra>