Context Based Storage and Retrieval of Multimedia Contents

By

Erum Afzal
(2009-NUST-MS PhD-IT-32)

Supervisor

Dr. Sharifullah Khan

A thesis submitted in partial fulfillment of
The requirements for the degree of
Masters in Information Technology

In

School of Electrical Engineering and Computer Science

National University of Sciences and Technology (NUST)

H-12, Islamabad, Pakistan

(September 2013)

APPROVAL

It is certified that the contents and form of thesis entitled **"Context Based Storage and Retrieval of Multimedia Contents"** submitted by **Erum Afzal**, have been found satisfactory for the requirement of degree.

Advisor: _____

(Dr. Shrifullah Khan)

Committee Member: _____

(Dr. Muhammad Murtza Khan)

Committee Member: _____

(Mr. Maajid Maqbool)

Committee Member: _____

(Mr.Muhammad Bilal)

IN THE NAME OF ALMIGHTY ALLAH

THE MOST BENEFICENT AND THE MOST MERCIFUL

TO MY PARENTS

CERTIFICATE OF ORIGINALITY

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: Erum Afzal

Signature: _____

ACKNOWLEDGEMENTS

First of all I am extremely thankful to Almighty Allah for giving me courage and strength to complete this challenging task and to compete with international research community. I am also grateful to my family, especially my parents who have supported and encouraged me through their prayers that have always been with me.

I am highly thankful to Dr. Sharifullah Khan for his valuable suggestions and continuous guidance throughout my research work. His foresightedness and critical analysis of things taught me a lot about valuable research which will be more helpful to me in my practical life.

I am highly grateful to Dr. Muhammad Murtza Khan, Mr. Maajid Maqbool and Mr. Muhammad Bilal for their help and guidance throughout the research work. I am also thankful to all of my teachers who have been guiding me throughout my course work and have contributed to my knowledge. Their knowledge, guidance and training helped me a lot to carry out this research work.

Erum Afzal

TABLE OF CONTENTS

LIST OF FIGURES ..... **v**

LIST OF ABBREVIATIONS ..... **vii**

ABSTRACT ..... **viii**

INTRODUCTION **1**

1.1 Problem Background Introduction 1

1.1 DSPACE: Web Applications 2

1.2 Types of Hosting (DSPACE) 3

1.3 Characteristics of the Web Applications DSPACE 4

1.4 Problem Statement 6

1.2 Outline of Thesis 7

BACKGROUND **8**

2.1 Streaming 8

2.2 Types of Streaming 9

2.3 Features of Streaming 10

2.4 Difference between streaming and downloading 10

2.5 Requirements of the Streaming 11

2.6 Streaming media servers 11

2.7 Streaming Distribution Types 11

2.8 Web servers versus streaming servers 13

2.9 Internet protocols 14

RELATED WORK **19**

3.1 Fresh Analysis of Streaming Media Stored on the Web 19

3.2 Multimedia Streaming at YouTube 20

3.3 HTML5 Streaming on Demand 21

3.4 Apple Live Streaming 22

3.5 Microsoft Smooth Streaming 24

SYSTEM ARCHITECTURE AND DESIGN **27**

4.1 Plug-in based Streaming Implementation: 28

4.2 Implementation of Players Technology: 30

4.3 HTML5 Technology: 33

4.5 Implementation of HTML5 Technology: 34
4.6 Storing and Capturing Context 36
4.7 Implementation of indexing on visual Data Fields in DSPACE: 36
EXPERIMENTAL RESULTS AND EVALUATION 39
5.1 Motivation 39
5.2 Procedure 39
5.3 Frame Rate: 40
5.4 CPU Utilization:- 43
CONCLUSION AND FUTURE DIRECTIONS 49
6.1 Conclusion 49
6.2 Future Directions 50
References 51

LIST OF FIGURES

Figure 1: Video streaming environment 9
Figure 2: Downloading 9
Figure 3: Streaming 9
Figure 4: Uni Cast Streaming 12
Figure 5: Broadcast Streaming 12
Figure 6: Multicast Streaming 13
Figure 7:HTML5 Streaming on Demand 21
Figure 8: Apple's Live Streaming Environment 23
Figure 9: Disk Format in Smooth Streaming 25
Figure 10: File Format in Smooth Streaming 26
Figure 11: DSPACE Environment 27
Figure 12: Flash Player code example 29
Figure 13: Plugin based Environment in DSPACE 30
Figure 14: MP3 Audio Streaming Code for DSPACE 31
Figure 15: Quick Player DSPACE code 32
Figure 16: Flash Player DSPACE Code 33
Figure 17: HTML5 Video Tag 34
Figure 18: HTML5 Code Environment in DSPACE 34
Figure 19: Table tag iin XML code of DSPACE 35
Figure 20:HTML5 code in DSPACE 35
Figure 21: FRS at cable network connection 41
Figure 22: FRS in wireless connection 42
Figure 23: CPU Utilization in cable network 44
Figure 24 : CPU Utilization in wireless network 45
Figure 25: CPU Utilization without using Video Card Device 46
Figure 26: Jan Ozer's analysis at Streaming Learning Center 47

LIST OF TABLES

Table 1: Application layer and Transport Layer Protocol 14
Table 2:FRS at cable network connection 40
Table 3: FRS in wireless connection..... 41
Table 4: Comparison of FRS online-I..... 43
Table 5: Comparison of FRS online-II 43
Table 6: CPU Utilization in cable network..... 44
Table 7: CPU Utilization in wireless network 45
Table 8:CPU Utilization without using video card devices 46

LIST OF ABBREVIATIONS

IR	Institutional Repository
FRS	Frame Rate Per Second
HTML5	Hyper Text Mark Up Language version 5
HTTP	Hyper Text Transfer Protocol
GUI	Graphical User Inter Face
Web	World Wide Web
API	Application Programming Interface
IE	Internet Explorer
P2P	Peer to Peer
JSP	Java Server Pages
VOD	Video on Demand

ABSTRACT

Web access is growing rapidly around the globe which is leading to the demand of the web application development and delivery of multimedia contents on the web. Streaming multimedia contents to meet the requirement with the existing web applications over HTTP are proved to be very costly. Multimedia players over the web application have reduced the cost of streaming server and caching but they require coding and processing at the server end during the transmission of media files. When payload increases performance of the server may have to compromise. HTML5 video tag is browser based technology which has reduced even the cost of processing at the time of transmission. It allows the browser to communicate with multimedia contents directly. In our findings we have implemented multimedia players (Quick Time, Flash and MP4) and HTML5 API in DSPACE for the streaming of media file to compare both of the techniques and find out which of the technology is optimum. DSPACE is IR a web based application where scholarly contents are collected, managed and archived. We have found both technologies are not best solution for dynamic growth of the streaming of the media file but each of them has its own priority over each other in respect to payload, size, quality, interactivity and hardware dependency.

CHAPTER 1

INTRODUCTION

This chapter includes introduction of the research work that has been undertaken in this thesis. It includes the motivation for research, problem background and problem definition of the research work.

1.1 Problem Background Introduction

Due to advent and quick accessibility of the latest technology multimedia contents are getting more popularity on the Web. People capture each event through digital cameras in the form of digital photos and videos and share it on the Web. Multimedia contents generated by users are being shared in the social networks like Face book, Twitter and YouTube etc.

Web Applications are most widely used as information storage hub on the Web; examples are Institutional Repositories (IRs) like DSPACE and Fedora. In these IRs data is stored in all formats including presentations, PDF files, MS Excel sheets, MS Word documents and audio video lectures. To bring up different types of data into single integrated platform has been a challenging task ever because each data type has different parameters and requirements in term of storage and retrieval. At an abstraction level, data is viewed as two forms visual and textual forms.

In DSPACE almost all format of data is stored and also accessed from different resources. In DSPACE textual data is accessed through simple HTTP over Tomcat web application server. Whereas for multimedia content there is need to add multimedia server which could stream multimedia contents online. If streaming server is implemented in DSPACE there would be different over heads to be managed at server one of them is data format. As multimedia data is generated at single point, it can be accessed or store from different sources. If we install an Adobe Flash streaming server, then all of data must be stored and accessed in acceptable format defined in the Adobe's Server. At the server end, it is possible to handle the problems by using conversion programs for storage and retrieval whereas at a user end it is difficult to convert the data in an appropriate format before storing and retrieving. There are a large number of Web users who store and retrieve data on IRs but they don't have even basic concept of multimedia format and conversions. It is the main reason that no such streaming technology has been implemented in DSPACE. A user can upload multimedia contents and download contents on the system to view.

1.1 DSPACE: Web Applications

Web applications are applications which deliver web contents to clients through a server. A web application is type of an internet application which includes web mail, digital repositories, web logs, discussion forums, wikis, and online retail shops. One reason of the popularity of the web applications is the ubiquity of the net browser. Browser acts as a client and mostly a thin client. It has ability to maintain and update web contents without disturbing or installing application software over millions of the clients; accessing the contents which is also a reason of its popularity among the clients. On internet all applications are not web applications as streaming server and gaming server.

During the last few years we are relaying on internet application in the affairs of business, education, entertainment etc. An institutional repository is used to manage huge volume of data and archive literary contents of an institute. As by its name, IR is uphold by an educational institute for the purpose of managing the contents necessary for an educational purposes. For example, a university may have IR, where all publications of students and faculty members, lectures, details of conferences and workshops are archived. Students and faculty members can access these contents through internet from any part of the world. Due to the popularity of the internet applications IRs are becoming more demanding applications. These applications are also referred as a hosting platform. It is necessary to host application on the server to make it available on the internet. For example, a website is hosted on the web server which is a more powerful computer that deals with hundred of requests simultaneously. It must be connected to the internet 24/7 so that it can be accessed at any time. The complexity and cost of maintaining the host platform, infrastructure is the key reason of the emergence of the web hosting by a third party. A web hosting provider sells space of their web servers. They provide 24 hours high speed reliable internet connection, so that web contents would be accessed easily. Yahoo's *Small Business Web hosting* service is one of the examples.

1.1.1 Complexity and cost in Web Application (DSPACE) Management:

There are following complexity and cost issues in managing an own web server.

1. Servers and software (Web server, email server, database server, streaming server, data protection software etc) for individual can be costly.
2. It requires 24 hours high bandwidth of the internet connection which can also be expensive.

3. Configuration of all servers and ensuring availability of the server can be complicated.
4. Maintenance of a web application requires 24 hours support, expertise and understanding of the system and working of the system.

Hosting platforms let a user focuses on the utilization of multimedia contents instead of storing and managing the contents through latest technology. These platforms are expected to give the guarantees of response time and throughput of the web contents on the web applications in return for revenue [15]. The services and guarantees are made upon the agreement of the service and its level.

1.2 Types of Hosting (DSPACE)

With the advent and demand of latest technology in computing and communication prices of hardware is falling constantly. This has made more attractive to build server clusters with commodity hardware for web application hosting platforms alternative to the conventional huge microprocessor servers. Depending on the requirement of the resources of the web application, performance and guaranteed services a platform may be developed in dedicated or sharing host fashion.

1.2.1 Dedicated Server hosting web Application

In a dedicated server hosting for web applications, a web site is running on the subset of the hosting server. It is used to run huge volume application (lager Clustered). The Server is allocated to the only one part of the application at most in specific period of time. In this fashion the server cannot be shared with others due to the demand of work load entailed on the single application or a part of the application. In such a situation an entire application is run on one cluster, for example, a web search engine Google etc or a single module in the cluster is dedicated to an application, for example, service provided by IR to students.

1.2.2 Shared Server Hosting Web Applications

In a shared server hosting of a web application, a large number of the web applications are running on the same cluster. The applications may include streaming server, web servers, email, servers, gaming servers and business web sites [17]. In some cases, a number of the web applications increase by the number of nodes in the cluster. Each application is run on the subset of the nodes and these subsets can overlap. In dedicated infrastructure one slot of application or whole application that affirm their extra charge, financial grounds of room, power, chill, and

charge make shared hosting fashion an attractive option for so many web application hosting environment.

1.3 Characteristics of the Web Applications DSPACE

There are following design requirements for the web application design, we have analyzed them with respect to DSPACE.

1 Current web applications are complex and are designed in a multi-tire distributed trend. A multi-tier infrastructure is a flexible way for such applications. Each tier consists upon an individual layer which provides particular functionality to its preceding (top) layer and uses the functionality of the successor (bottom) layer to accomplish the part of processing of an incoming request. DSPACE is one of the examples of multi-tier web application. All layers in a system continue processing on in-coming requests. As results produce by one layer is taken as an input by other layer, so there may be a need of replications and caching at some layers. These features of web applications make inferring requirements and provisioning capability of non-trivial jobs.

2 On internet, dynamic content delivery is getting more popularity as compared to the static data [17]. It requires more intensive resources in the creation of the dynamic contents as of the static contents. That is reported for the volume of the traffic on the internet for few years back. So while designing a web application, designers have to review the perspective of dynamic contents and their development behavior.

3 Diverse software components are used in the development of the web applications. For example DSPACE consists of three tiered architecture. A front end layer that is for user interfaces and work on HTTP processing, the central layer is the core of the application where all processing is taken place and backend or bottom layer is database layer. All layers are different in providing functionalities and processing. Each layer has its own requirement of software for development and processing which may be differing from each other.

4 Web applications are heterogeneous in nature. New hardware can be added and removed. It is required that replacement in hardware would not affect the structure and functionalities of the application.

5 Web application come across diverse dynamic variations load [11, 16], it consist upon long term fluctuation which remained for few hours or the whole day. On other hand short term variation remained for one transaction or few long transactions. It is very difficult to estimate the peak work load in an application.

1. Third Party Plug-in

HTML5 supports video streaming without using any third party plug-in. It takes advantage over other rich internet application, HTML5 code renders multimedia Applications "Out of Box" on any device [21] which means it does not require installing any proprietary player to encode the contents before playing into browser. While on other hand In Plug-in based Technology a player is embedded inside the web which can play only the type of the media format which is defined in the standard. There are different proprietary for different media players as most of the organization has developed their own format to stored media over the web. For Example Flash by Adobe, Silverlight by Microsoft and Quick Time by Apple have different proprietary format. In addition to this Gears by Google and Java FX by Oracle are web technologies for offline storage and client side based web application [22].

If client want to play flash SWF file he/she has to install flash player on the system which leads to series of limitations as player should be compatible with version of the player installed on the device and the version used for the development of the contents at the server end [20].

Steven J. Vaughan-Nichols has said that W3C's has answered the proprietary technologies through HTML5. The open standard to be adopted by two dominating products Google and Apple has made developers to develop rich graphics, typographic effects animated sequences and transitions without using any proprietary Plug-in like Flash, Quick Time and Sliverlight. In short it supports all functions that are expected by proprietary add ins.

Steve Jobs the CEO at Apple Inc. has termed HTML5 as "Opened" standard and believed that standard on the web should be open. There are many products developed in Flash but it is 100% proprietary. Only Adobe has authority for further "enhancement" in the products. According to Jobs, Adobe system is closed system as Flash products are openly available on the Web but these products are controlled by the Adobe. Apple has adopted open standards HTML5, CSS and JavaScript for iPhone Applications. According to Jobs, by using these open standards it is tried to ship all mobile devices with "High performance and low power implementations.

Ian Hickson[23] HTML5 editor said, one of the major goal of HTML5 is to make free from proprietary technologies such as Flash, Silverlight, and JavaFX etc. According to another director of w3C Tim Berners-Lee, HTML5 is big shift in web pages it is making web more and more client side applications platform.

Every organization implements most convenient and optimal method to access the contents. Text Based Web Applications do not provide streaming of audio video online. It is used in purely multimedia based applications as Youtube and Vismo both applications are solely multimedia applications. HTML5 provides mechanism to access Video data using HTTP method. As by Tim Berners Lee, "We've had the pieces for a while," and bringing them together in HTML 5 "multiplies the power of each one." There is no need of Proprietary software. We have to just add HTML5 video code into browser and multimedia contents can be accessed.

HTML5 is also preferable in DSPACE as it is not multimedia based application we may require techniques to manage and stream high quality multimedia contents. It has power full content management storage and retrieval methods on various data formats of texts than of streaming. It is great advantage if multimedia contents would be streamed without using any proprietary Tools.

1.4 Problem Statement

In DSPACE data is archived in textual and visual format. Development with respect to storage and retrieval of textual contents is being carried out rapidly. On other hand development in the storage and retrieval of visual contents is slow. Reason of slow development in visual contents managements is that visual contents are diverse in nature. Each visual object is different from other object in term of format, size, quality and bitrate. This difference requires different methods of storage and retrieval. So, it is very difficult to implement all methods in one repository as data is archived from different resources and also delivered to different destinations. There are several methods to manage visual contents on the web. YouTube is one of the most popular online repositories of multimedia contents management. It has implemented streaming server at server end and flash media player. At client side it is required to install an Adobe flash player to stream contents online. DSPACE is ruing at Tomcat (web server) and backend database server (PosgreSQL or Oracal), it has three layered architecture. Adding streaming serve would lead to add new layer in architecture and one more server for content management. Other than this it requires to add player as well to play contents at both client and

server end. All in-coming and out-going data would be required to convert in appropriate standard format defined by player. To change the parameters of player according to the requirement of repository we need to buy propriety of the player. Same player version is required to install at both ends. It is challenging task to implement streaming server efficiently at DSPACE. So, currently visual contents are just stored in DSPACE and are downloaded at client side to view contents.

In our thesis we have purposed two solutions regarding to the visual contents storage and retrieval. First, we have purposed storage and retrieval of visual contents over HTTP like textual contents are accessed. We have implemented HTML5 technology where there is no need of third party plug-in, special server and protocol/standard. We have compared the HTML5 technology with plug-in technology in terms of CPU utilization and frame rate. Both technologies have their own merits and demerits on each other. We have concluded that in the environment like DSPACE where diverse kind of data is archived, HTML5 is a better choice over plug-in based technology. Secondly we have purposed method to access context by adding textual description with visual object at the time of storage. We can add index and access the contents by using attached information.

1.2 Outline of Thesis

The rest of the thesis is organized as follows: Chapter 2 describes background study. Chapter 3 discusses literature review followed by summarization and limitations of the current research. Chapter 4 highlights the proposed methodology and system architecture and configuration of the HTML5 code and multimedia player on DSPACE. Chapter 5 shows the compression of the streaming versus media player. Chapter 6 shows the conclusion and future work.

CHAPTER 2

BACKGROUND

This chapter provides background knowledge that is helpful in understanding the context of this research. It includes basic definitions of important terms like streaming, Types and working etc.

2.1 Streaming

The idea of multimedia streaming is not as older but its demand is increasing [1, 18]. As streaming is not old term it has been introduced almost two decade back or less. The concept of the streaming is the result of ubiquity of the Web, rapid growth of social networking and advent of the latest digital devices. This change made multimedia developers user to look at the different modes to deliver multimedia contents on the Internet. Making research on the broadcast mode of television and radio these organizations were able to deliver live or on demand multimedia contents using the web. Live streaming referred to the programs that are performed at somewhere and user is able to access it from home through live streaming server where as on demand access means stored multimedia contents in any repository and you can access at any time from any part of the world through internet. People can watch live contents like sport games, music concerts, news shows and other entertainment, these contents are also available on demand through the Internet. The demand of multimedia streaming is also growing with rapid advent of the handheld devices like PDA, iPhones, iPod, iPad etc.

Streaming is only possible in real life environment means multimedia contents are only delivered when some particular event is taken place. Real time is also referred as contents delivery with minimum delay where time of transmitted and received contents is same. Streaming requires high data transfer rate and consumes additional bandwidth in accessing multimedia objects than conventional web contents. Multimedia traffic seems busty and delay sensitive that's why it required more bandwidth of the internet connection [5] .Multimedia contents require more storage space as they are huge in size it leads to the requirement of the more media server and caches for proxy. Multimedia contents take more time in downloading than of from convention web contents. Therefore as contrast with traditional web page designs it presents new challenges for web application developers. Figure1 shows the working environment of streaming

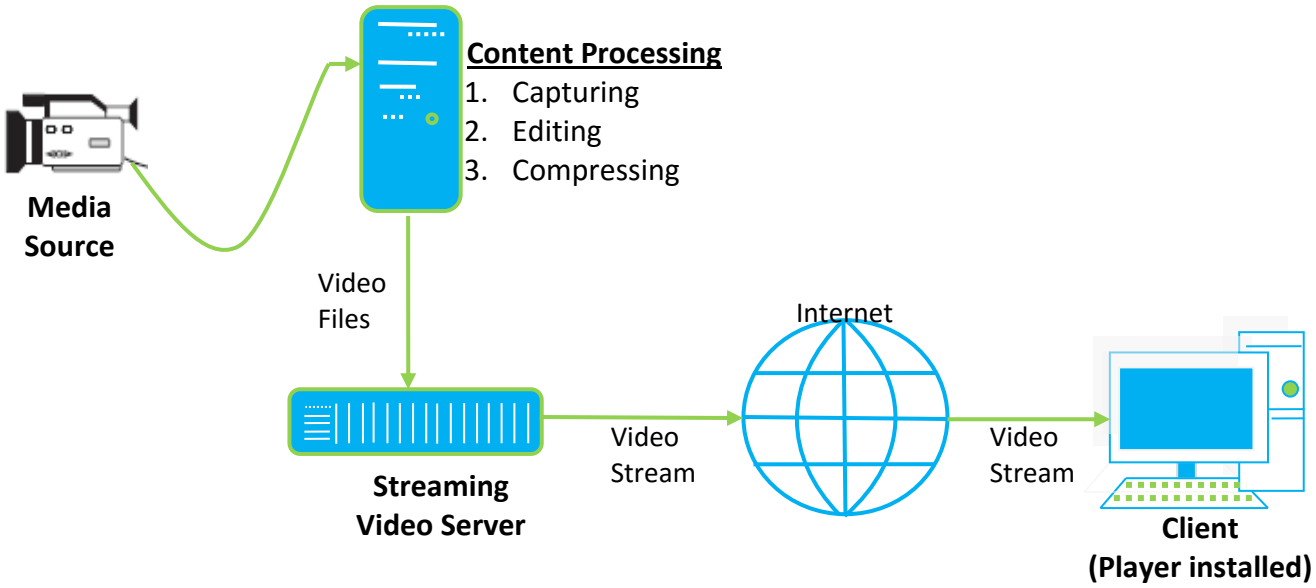

Figure 1: Video streaming environment

There are two [1] approaches of multimedia transmission through the Web. One is downloading approach in which user can watch/listen the contents after downloading whole file from the web to his/her device(PC, mobile phone). Downloading is effected by two factors size of file and bandwidth of the internet. For instance if we want to download an .mp3 (audio file of 4.8 MB size) with 28.8k Modem speed, in 40 minutes the whole file will be downloaded. Second approach is streaming in which there is no need to download whole file at client side. You can have access to particular contents after parts are streams and coded in cache or cookies of the web browser.

Figure 3: Streaming

Figure 2: Downloading

In both approaches are referred to the analogy of drinking water as shown in figure2 and figure3 respectively. Streaming approach is as you drink water directly from water bottle without pouring into glass, whereas downloading approach is of the analogy like first to pour water in glass then drinking it (download then play). There is an another approach of accessing multimedia content is Pseudo streaming which is also known as progressive streaming. In this

type of streaming contents are not downloaded or streamed before play. It plays contents in the player embedded in the web without interactivity of the user, which means user can only view contents which are streamed over time he/she cannot play any part of audio/video even from current playing file. If speed of the Internet is high user is high he/she can access file fast, without any delay. Media player is interleaved [1] with multimedia file so that the parts which are played would be in same sequence in actual file. This approach is just like television transmission mode, where contents are broadcasted, played and vanished. You cannot replay anything. In this approach important contents are recorded and can be played later. But for this you need to add up other tools with the Web.

2.3 Features of Streaming

There are following features of the streaming

The features of the streaming are delivering live multimedia contents for example any sport game, musical show or political speech from any part of the universe as it is being played or happening. It can also make us to have access to long videos which are stored in any repository ; the access to these videos is not like pseudo streaming we can play any part of the video from any location just like we play on media player. There is no need of extra hard disk space to download contents as media file is not downloaded at client side it is directly played from streaming server that is attached with digital repository. User can only store user setting of particular media file and web link of that file. Streaming servers uses the bandwidth required for streaming rate of bandwidth remained fix, in case media streamed content increases this rate contents can be broken as data packet lost due to exceed in the bandwidth rate. In network if one multimedia object is required to all clients, it can be multicast throughout the network without streaming same object at client side.

2.4 Difference between streaming and downloading

These features vary in different approaches of streaming. In progressive approach live streaming feature would not be useful, user interactivity is limited as he/she cannot play/pause the video contents and multicasting is not applicable, while in downloading approach even in low bandwidth lost data can be recovered, no special server application(software) is required to download data. Downloading and streaming approaches have advantages over each other and is used for different causes. Both technologies are complement for each other. If we want to pass

on live contents and posses high band rate we can use streaming technology otherwise at low broadband rate we can access stored contents through downloading techniques.

2.5 Requirements of the Streaming

The requirements of media streaming on the internet include bandwidth, delay and packet loss. The internet connection does not endow with any assurance that data packet containing media contents will not be lost on the way or reach at destination following the defined route. It is possible that data packets may not delivered in same order as these are sent. Taking this issue into consideration different protocols and application level mechanisms are implemented. Sometimes streaming media contents are compressed so that it would fulfill the requirement of the throughput of the internet at user end. There is difference [6] between actual throughput and the throughput provided by dial up or DSL ISPs (Internet Service Provider), Connection throughput stated by ISPs is higher than actual throughput as it is affected by other factors e.g. noise and propagation over connection line. In cable modem [1] available bandwidth is reduced as number users increase as one head provide connection to many end users. In contrast to cable modem ISDN and E-1 internet connection provides guaranteed bandwidth rate as dedicated network cable for the internet connection is used. It is rule that media is coded at lower rate bandwidth than actual bandwidth rate.

2.6 Streaming media servers

Streaming media contents are encoded in streaming files which are also called as movies in quick time. These files are coded into particular streaming format to access over streaming server. There is variety of streaming file formats such as .wma and .wmv is format of Microsoft streaming server, .ra and.rm is format of Real time streaming server, .flv is format of flash streaming server and .mov is format of Quick Time streaming server. Streaming server use timing control information to handle flow rate of the multimedia contents especially audio and video files, timing control information is defined at the time of file coding format.

2.7 Streaming Distribution Types

There are three types [14] of stream distribution over the network;

1. First one is uni-cast in which one streamed media file is sent to each user individually which is opened by clicking on RTSP URL of the media file as shown in figure 4. This

method does not use bandwidth efficiently but it lets user to use different functionality of RTSP (streaming protocol) like playing different parts of the file and playing different files at the same time.

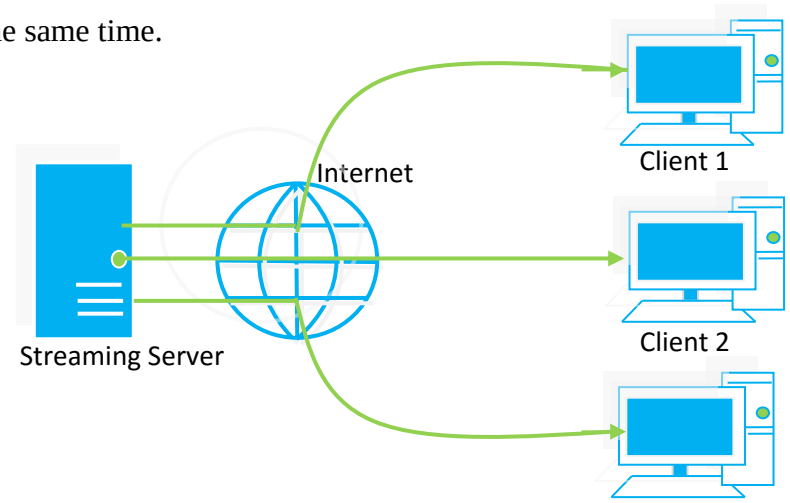

Figure 4: Uni Cast Streaming

2. Second type is broadcast streaming in which one streamed file is send throughout the network as shown in figure5. Internet does not support broadcast it can be used in LAN. In broadcasted media file interactivity support is limited as user cannot play any part of the video as well as replay, pause action on file.

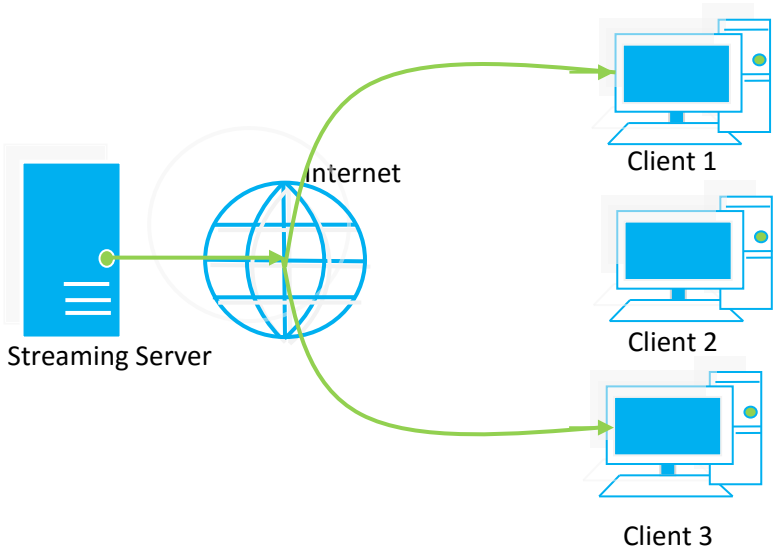

Figure 5: Broadcast Streaming

3. The third type is multicasting where bandwidth is utilized efficiently as shown in figure 6. In this type of stream distribution one copy of the streamed file is send to the down branch of the network there can be multi clients at one branch of the network. It required

router which route the streamed file as required by particular client. Clients device can direct be connected with nearest router to access streamed file instead of communicating with streaming server. There is no user interactivity with streamed file in multicasting casting approach as well which means user[1] can see part of his/her choice from streamed file.

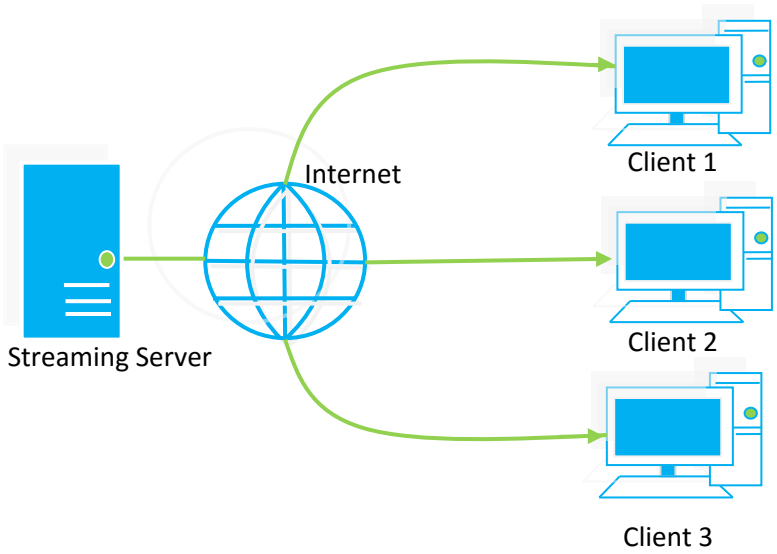

Figure 6: Multicast Streaming

2.8 Web servers versus streaming servers

The difference between web server and media streaming server is that web server can download media file from website it does not stream the contents. It does not provide user interactivity over streamed contents. It does not provide the guarantee of on time delivery but ensure the accurate delivery of the contents. To deliver Media files on html pages web servers uses HTTP Protocol. Streaming contents have different requirements. It requires real time delivery [1] with required QoS requirements where as at reasonable level transmission errors are acceptable. Streaming through web server is possible whereas there would be no control over the delivery speed of stream contents. Delivery of streaming content would slow if the congestion of network is high. Means when delivery speed of network would higher than the speed of media player of the packets may lose on the way. In such situation at receiving end there is required a large size cache to buffer the contents before these are played.

On other hand streaming server is type of the server which delivers the multimedia contents of the web, it streams live webcasted contents as well as recorded stored contents on the web. It

provides user interactivity towards the streaming files. There are following additional functions [1] of streaming server over ordinary HTTP web server. First it provides real time flow control of the multimedia contents; second it provides intelligent switching, third and foremost important function is user interactivity. It also has Skip Protection [1] function. On client side, it uses faster buffer ahead than real time streaming by using excess bandwidth. If data packets are lost due to network congestion or any other reason only lost packets are retransmitted instead of whole stream it reduces the network traffic.

The QuickTime streaming server can deliver up to 4,000 media streams contents by using single server. Any number of streaming servers can be added to meet increasing traffic requirement.

2.9 Internet protocols

To understand the working of internet is to understand the layers. Each layer performs a specific function. On predefined agreed interface each layer interacts with its below layer to receive flow of contents and to deliver processed contents interacts with its above layer. Internet comprises on many networks [1] and used in variety of applications. Following table shows different internet applications along with transmission protocol.

Table 1: Application layer and Transport Layer Protocol

Application	Application-layer protocol	Transport-layer protocol
E-mail	SMTP	TCP
Remote terminal access	Telnet	TCP
Web	HTTP	TCP
File transfer	FTP	TCP
Remote file server	NFS	UDP
Streaming media	RTSP or proprietary	UDP
Voice-over IP	Proprietary	UDP
Network management	SNMP	UDP
Routing protocol	RIP	UDP
Domain name translation	DNS	UDP

2.9.1 Internet Protocol (IP)

Internet protocol is communication protocol at 3rd layer of OSI model. It is considered as unreliable for three reasons

1. It suffers from variable latency of network
2. Data packets are received from different order as these are sent.

3. Data packets may be damaged or lost on the way.

These insufficiencies cannot be resolved by IP itself but can be fixed at upper level layers i.e. at Transport layer and Application layer. Transmission control Protocol also known as TCP is communication protocol at Transport Layer. It [1] is provides highly reliability in delivery of general purpose data but It does not support streaming. Size of IP data packet is 40 bytes. Its header has three parts; IP, UPD and RTP header with 20 bytes 8 bytes and 12 bytes respectively. It is not necessity in stream to have same IP Packet length. As packet format is shown in figure.

There is no restriction in embedding IP data packets it can be delivered in any physical network (cable or wireless), in two way communication network or broadcasting one way delivery method. Network Layer is considered as convergence point as all data packets are likely to move through this layer.

2.9.2 Transport Control Protocol (TCP) and User Datagram Protocol (UDP)

Transport Control Protocol (TCP) and User Datagram Protocol (UDP) are two protocols at same layer but they are used differently.

TCP creates sequence of bytes and sends signals at destination about the arrival of the packet if sent packet is not acknowledged at specified period of time it is retransmitted. This method can detect missing bytes and can easily be retransmitted. This retransmission can affect latency but it major issue in network transmission.

With video and audio multimedia contents viewer or listener requires continuous streams. Lost in packets and then retransmission of the lost bytes become the cause of delays in transmission. If data transmission errors are increased the stream receiving buffer of the stream media player is need to be unfilled which interrupt the sequence of the file. To avoid form situation retransmission is not done [1].UDP does this very efficiently. It does not retransmit lost packets. The loss of packets may cause the loss of few frames of file and destruction of the actual quality pixels but media players are designed in a way to cover up these insufficiencies.

2.9.3 HyperText Transfer Protocol (HTTP)

HTTP [14,9] is protocol at application layer. HTTP protocol carries HTML pages of web sites user can have access different web pages using hyperlinks in a websites. Data protection is ensured by using checksums technique. Client server has two way communications between them as client computer sends feedback so if any of the data packets is lost it can be figured out easily and retransmitted to client. If there are fewer concurrent clients (user) they can down load

multimedia files of small size using HTTP protocol. Multimedia [1] file can be downloaded even with slower internet speed than media rate but this file may not be played efficiently. Transfer time of particular file depends upon two factor speed of internet connection and size of the file to be downloaded. There are various versions of HTTP.

1. HTTP/0.9 is first version of HTTP. I was very simple protocol it can transmit just web contents of the web pages over the web.
2. HTTP/1.0 is improved version of HTTP/0.9 it is use to transmit web contents in the format of MIME. MIME contains the metadata of the contents. It is used for semantic request and response of the client (user).
3. HTTP/1.0 was not taken into consideration as it does not support proxies in hierarchy caching of the web addresses, the requirement of the persistent internet connections, and virtual servers. These required come up with HTTP/1.1 version of the protocol. It was establish to implement deficiencies of the earlier version effectively. It has improved the browsing option from simple retrieval of data to user's requirement searching capabilities with annotation and front end updates.

HTTP has various methods to implement for request purposes and most of the methods are open ended not restricted. This protocol is builds on the principle of reference of URI (Uniform Resource Identifier), location URL () and name URN () to represent the resource on which different technique are implemented. MIME [1] defines the formats used by internet messages so messages are passed in defined format. HTTP is considered as generic protocol to be used for communication between users and other gateways of the internet systems like SMTP, NNTP, and FTP protocols. In this way HTTP can have access of hypermedia available at other various applications.

2.9.4 Real-time Transport Protocol (RTP)

Real time Transport Protocol is used to stream multimedia contents over IP protocol. Multimedia [1,14] data for streaming is encapsulated in RTP Packets without taking format and contents of the file into consideration.RTP also include timestamps and sequence number data field in header. It is executed on UDP and functionalities used are multiplexing and checksum. RTP allows media server to server video stream at appropriate speed. At client end media player receives RTP packets and assemble them in an accurate order then. After sequencing packets streams are played at an appropriate speed. RTP transmission is real time transmission where lost

packets are not retransmitted. Error concealment and packet replications are the methods that are used to cope up with missing packets. If internet speed is lower than the data transfer rate of media file transmission breaks up and media player plays content poorly or does not play sometimes. On other hand if internet speed is faster the extra bandwidth does not affect the speed of the media contents being played on media player.

2.9.5 Real-Time Control Protocol (RTCP)

Real Time control Protocol is used with Real Time Transport Protocol. It is used for client server connection over TCP. Function of RTCP is to support in synchronization of different multimedia contents like audio and video contents in a file [14]. It is used to take feedback on the quality of network provided to each client in particular RTP session. It informs the server about quality of internet, statistics of the viewer and identity of the viewer. Message [1,14] in RTCP consists upon statistics of lost packets and jitter. The information includes in message can be used to improve transmission and control the session in high level applications for example to control congestion bit rate of the stream can be rectify using the information enclosed in RTCP.RTCP[14] message can also use to control video conferences which is interactivity among various clients with each other.

2.9.6 Real-Time Streaming Protocol (RTSP)

Real Time Streaming Protocol is used at Application layer. It is used to control real time streaming contents. For data delivery it uses RTP as underlying content transmission protocol. It provides user interactivity [1, 14] like Play, Stop, Pause, Fast Forward and Review. User can access any part of the stream content. It [14] does not show the delivery signals rather it allows user to control the signals. It act like dispatcher which does not transmit data packets but controls how and when packets are transmitted using RTP. It acts [14] like remote control for multimedia server. Using RTSP server can adjust the bandwidth according to the requirement. RTSP chooses optimum route to deliver contents to client. For example if UDP is not working in particular network due to security configuration multicast UDP or TCP can be used for transmission.

RTSP and HTTP/1.1 are more likely to be same in syntax and operations but differ in applications. In RTSP request can be made from any side client or server during interaction whereas in HTTP only client can make request.

2.9.7 RTMP

Real Time Messaging Protocol is defined by Adobe systems for streaming multimedia contents over the web. Content transmission is done over TCP. Mostly web server transmits multimedia contents using HTTP where client make a socket connection with adobe flash server using RTMP. This RTMP connection allows contents streaming between client and server [14, 7]. Client and server communicate with each other using RTMP messages, the data of message can also be multimedia data.

2.9.8 SMIL

The Synchronized Multimedia Integration Language (SMIL) is used [14] to combine different type of multimedia data like audio, video and images etc through the interface of website. It is embedded in web page to control action of user input on the web. It allows the publisher to capture timing and behavior of the object. It is similar to html and can be created in any text editor. Layout of the content is described in parameter of language e.g. layout window for text, image and video.

2.9.9 SDP

The Session Description Protocol is used to describe the multimedia session. It is mostly used in Intranet environment but can also be used in multimedia conferences of other network fashion. [14] At server ends SDP announces the session of conference using Session Announcement Protocol (SAP). It periodically multicast announcement packet to the ports and multicast addresses reside at server end. It transmits the stream information in particular multimedia session and allows the clients to know about the participants in particular session.

.

CHAPTER 3

RELATED WORK

This chapter reviews the related work of the research undertaken in this thesis.

3.1 Fresh Analysis of Streaming Media Stored on the Web

Multimedia contents are diverse in nature therefore, there [2] is very little literature about the streaming contents stored on the web in digital repositories. We have reviewed literature on the analysis of multimedia contents on the web. In [5, 7] study, multimedia contents were analyzed and measured at client side. In [4, 7] study was made on specific web applications like YouTube etc. But streaming on all the web applications do not behave like the streams on the YouTube. Streaming media is stored and retrieved on the requirement of the client on each web application. So there is need to find generic behavior of the multimedia contents on the web application. We have seen that in literature, only specific behavior of multimedia contents was analyzed. There is need to analyzed generic behavior of multimedia contents on the web so that people would able to achieve contents accordingly.

Properties of the Media File

There are no recent research studies on the characteristics of the multimedia files on the web applications. In [2, 3,8], authors had discussed the properties of multimedia file in 1998 and 2003 respectively. But this study is too old. The facts discussed are out dated and also changed with the advent of latest technology. In 1997 [2], author accessed the multimedia contents stored on the web using Alta Vista search engine and analyzed each file individually. It was detected that web engine and website did not support the streaming in the real time environment. Whereas in the study of 2008[8], author analyzed 29% multimedia storage was encoded on the bitrates of the modem. In 2003 video resolution was of the 320 X 240 pixel depth. Now resolution rate has double to it or more. The code dominating formats were Real Media (.rm), Window Media (.wm) and Quick time (.mov). Now Adobe Flash Files (.flv) have taken more popularity over the web being lighten in size. In 2011[2], author studied that flash files are replacing previous codec in both audio and video formats. It analyzed that bulk of the information on the internet is several thousand times higher than the previous studies. Author used Larbin tool, a crawler that was customized with the web to crawl through the web search engine. On each instance 12.5 URLs were crawled and other tools used to find type of content. Media file parameters which had been considered as potential in impacting the quality of streaming over the web were accessed using

tool. The parameters consist upon size of the file, format of coding, frame rate, resolution, and duration of the file and average of the bit rate of encoded file.

This study shows the fresh analysis of the parameters of the streaming media but there are few short fall in the work.

1. Bit rate of multimedia object has been taken from the header of the object stored in the web portal but most of the time bit rate in header is different from the bit rate of the file played at client side. The study also has analyzed the public data only.
2. Paid contents and data from social networks like Facebook and Twitter have not been analyzed. Behavior and information of the multimedia objects in paid (private) portal may be different from the publically available contents.
3. There is considerable information on peer to peer network like Torrent etc. An analysis on it is ignored as contents are downloaded before playing. But parameters of the multimedia objects at peer to peer network can be examined and used in other quality improvement metric.

3.2 Multimedia Streaming at YouTube

Almost each internet user has familiarity with the YouTube. It popularizes the user generated multimedia objects which can be accessed by any one on the internet. Any user can upload and view any type of multimedia contents over YouTube. It is becoming more popular website day by day. It is ranked as third popular site which is most viewed universally. According to the survey [3], author claimed that YouTube spends one million dollar per day, on the consumption of bandwidth. It is increasing on the daily bases due to the client/server architecture.

In [4, 5, 6, 7, 12, and 13], who studied the most commonly used multimedia object source YouTube. They focused on the properties of the stored multimedia objects. In [4], authors examined ratio of the distribution of the media object generated by the user. They also analyzed the contents which were accessed and uploaded repeatedly. In [13], authors studied caching at a user ends. A huge volume of the traffic on the YouTube was gathered to explore the cashing. Cashing is basically memory area in the web browser which contain particular amount of the stream and then play at the client side. In [7], author studied the traffic analysis on the YouTube. This analysis tracked the transactions of the YouTube inside the campus network. Focus of the study was to analyze video access from the network boundary

side. There is relationship between the geography of the network and properties of the multimedia contents on social media and YouTube [6]. The study of [5, 12] presented the workload management in YouTube. In it traffic of the YouTube is collected then it is compared with traditional web application workload. The analysis is based on the utilization of the network bandwidth, popularity of the service of the object and sharing pattern. It has been examined the workload of the streaming media in respect of the effectiveness of performance optimizations on YouTube.

In this section we have studied literature on YouTube, as YouTube is one of the largest multimedia web applications. The study is about properties of multimedia object, caching, demand of contents on geographical bases and workload management of huge volume of data traffic on YouTube. Purpose of this section is to know how multimedia contents are stored and what parameters are required to consider while creating a multimedia repository.

3.3 HTML5 Streaming on Demand

With the increasing demand of streaming every web application tries to store media file on its web portal. But to add and manage streaming server is a complex and cost effective task. A web based streaming model has solved the problem. Streaming plug-in are embedded inside the browser. There is no need of special streaming server for stream of the media file. Swarm plugin is one of the examples, used for the streaming through browser. It is available for “Internet Explorer” and “Fire Fox” browsers. This plug-in is based on the media player VLC and the Tribler software which is funded by European Union. In this project Swam plug-in was used to distribute contents over P2P environment. It reduced the cost of back end infrastructure by distributing contents directly instead of storing them first at streaming server and then distribution among the network. Figure 7 represent infrastructure of HTML5 streaming on demand.

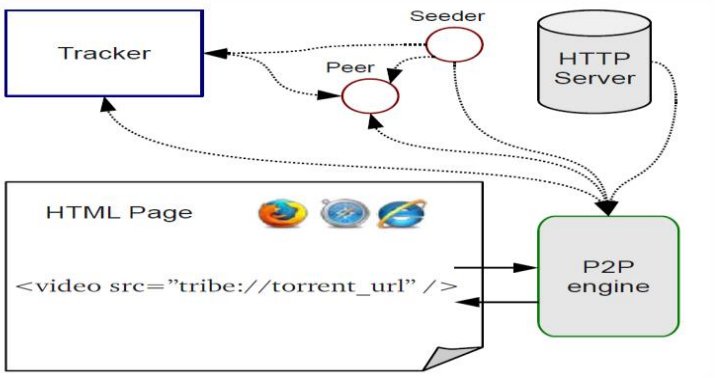

Figure 7:HTML5 Streaming on Demand

This system design was developed to overcome two shortcoming of the existing design of Bit Torrent (downloading software) P2P system with respect of VOD.

1. Implementation was placed inside the browser.
2. Overcoming low delays in accessing initial blocks of multimedia file.

For VOD streaming Browser was main target. Streaming plug-in was created to implement this technology which was taken right step. Initially this plug-in was developed on Mozilla Fire Fox browser on P2P-Next platform [11]. HTML5 tag named <video> was used to play videos. This tag was used to implement the properties of the player inside the browser.

Purpose to study this case is check how HTML5 technology can be implemented in real time streaming application. We have found that implementation of HTML5 is very simple as compare to install separate player at every user end. In HTML5, fallback of the video is streamed over HTTP instead of using standard streaming protocols. In P2P platform client have to install plug-in only when he/she participates to download or upload. The more desirable functionality would be that technology must be implemented inside the browser by default. We are also trying to implement HTML5 technology inside the browser.Along with HTML5, almost all browsers are trying to be more standardized, to more away from plug-in and towards developing web application as native applications. If technology offered through plug-in becomes part of the technology available in HTML5 browser based there would be no need of separate plug-in implementation.

3.4 Apple Live Streaming

Apples described the HTTP live streaming [11]. It uses HTTP streaming by taking advantage of MPEG-2. Apple has submitted their approach of live streaming using HTTP to the IETF. They use three components serve, distributor and client

1. The server manages the streamed media, encodes it digitally and encapsulates the result into the format that is delivered to the client. The server has small components named encoder and segmenter which are used to break a long stream file into a sequence of short video file.
2. The distributor consists upon the web services that accept different requests from clients and after analyzing a request, delivers a video file to the clients.

3. Client component is software at a user end. It initiates the request to access the required file. After the request is made, the required file is downloaded and segments are reassemble in order to play the actual file in continuous streams.

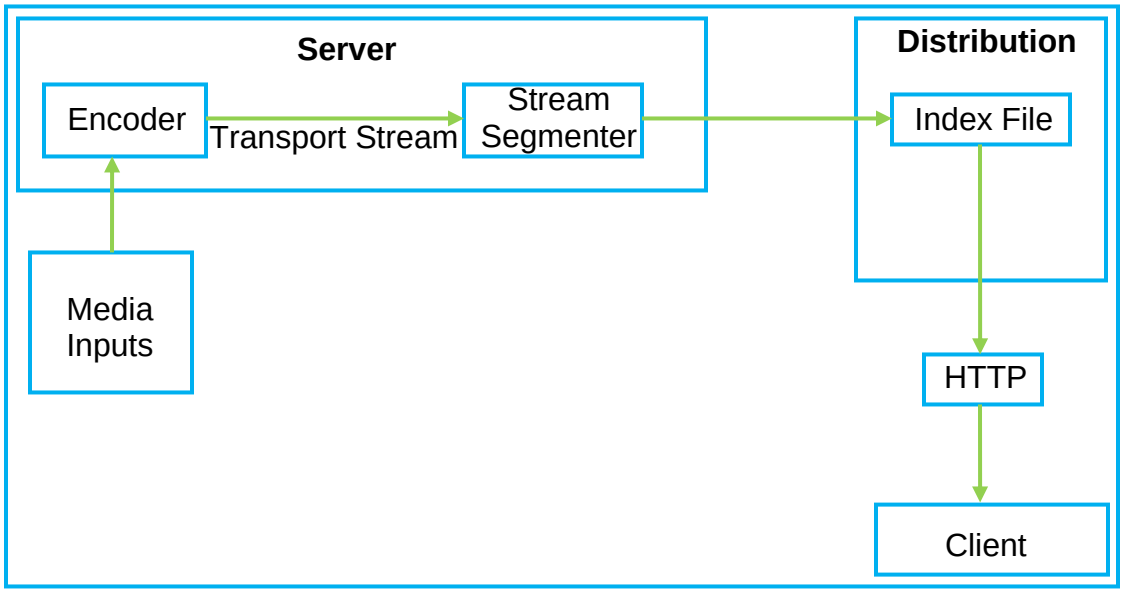

Figure 8: Apple's Live Streaming Environment

The above figure8 shows the configuration of Apple's HTTP streaming. The coder component takes media file in the form of audio or video, it encodes the file and then encapsulates the encoded file in MPEG_2 TS. After encapsulation encoder delivers encapsulated file to the segmenter. Segmenter takes the transported stream and divides it into small media file. It also creates an index of the media file. Index file contains the references of the each media file which is updated every time when a new file is segmented. At a client side, upon the request of the stream index file is fetched, then from index file URL of the desired media file is selected. After selecting a file it is reassembled to play it. The software offers three types of the configuration of encryption for data security.

1. In the first method, segmenter adds the URL and an encryption key in the index file. In this mode, one encryption key is used to encrypt all the files.
2. In second method, segmenter generates random keys, save them with file to the location. The reference of the location is added to the index of the file. Segmenter goes on through this cycle while making segment of the file.

3. In third method, key rotation concept used to generate a random key from n keys. In this method, a key is saved in a particular location and added as a reference in the index file. This cycle goes on to encrypt group of n files with different keys but same n keys would be used to encrypt next n files.

In this section we have studied working of Apple's smooth streaming method using MPG-2 technology. Purpose of this study was to see the working method of existing streaming technology. The details of video storage and retrieval had described above.

3.5 Microsoft Smooth Streaming

Microsoft has brought in smooth streaming based on HTTP as adaptive streaming extension. This extension was added as a web server feature at IIS7 (Internet Information Services). It provides streaming regardless the transmission bitrates of the media file. Streaming is made possible by detecting dynamically condition of the network connection. MP4 container was used to deliver media stream file. This type of container used disk format to store file on the web portal and wire format to transmit the media file across the network.

Every chunk of the video is known as MPEG-4 fragment. In contiguous MP4 file, chunks of the fragments of media file are stored. A full length file is stored on the disk in disk format and converted into chunks virtually upon the request of a client. The separate file is generated for each bitrates. Specifications and a set of complete protocols are available on the Web. Silverlight is online plug-in that can be added on Hotmail is one of the example of Microsoft smooth streaming. There are following assets of the Microsoft smooth streaming

1. MP4 in smooth streaming is different from traditional MP4 format. It uses two formats; .isma and .ismv. The extension .isma keeps the audio files, whereas .ismv keeps video and audio files both.
2. Server manifest file (.ism) informs about the relationship among the file on disk, available bitrates of the file and tracks of media file.
3. Client manifest file describes the available streams for a client. It also keeps the information of the CODECs used, bit rate encoded, resolution and other information of the media file.

Microsoft has migrated format from ASF (Advanced Systems Format) to MP4.Microsoft has anticipated numerous reasons for this migration from ASF to MP4 [10]. Following are reasons of format exchange

1. MP4 format bears less overhead as compare to .asf format.
2. MP4 supports H.264 codec standards which are high resolution standards and getting more popularity among media contents on the web.
3. In MP4 parsing is very simple and easy.
4. Mp4 Support fragmentation which is used to manage payload.

Disk File Format

For contiguous files on the disk Microsoft smooth streaming defines disk file format. The basic unit of the container is represented by a box. Metadata and data itself is stored in theses boxes as shown in figure 9. The disk file format container keeps movie metadata (.moov) which is file level metadata. The fragment part of the file keeps the payload. There can be a number of fragments of a file depending upon the size of the file. Fragment section consists upon two parts one is a fragment of movie with extension of .moof and other is a fragment of media file with extension of .mdat. The movie fragment keeps more precise metadata at the fragment level where as at media data section actual media is resided. Accurate and random search in the file is made at the movie fragment on the random access information (mfra).

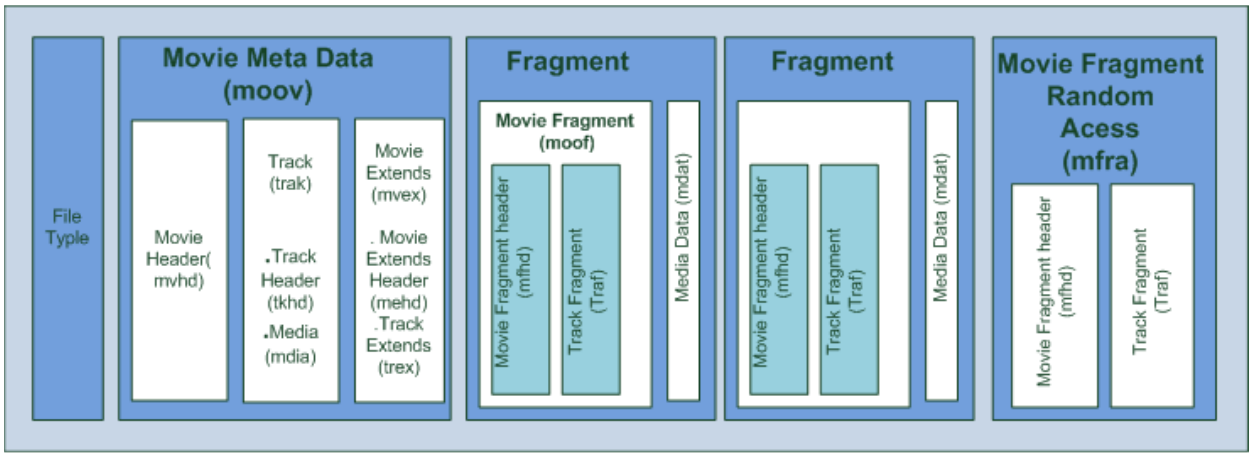

Figure 9: Disk Format in Smooth Streaming

File Format

Wire file format is a subset of the disk file format. When a user requests for a part of media file on the web, the server looks up the requested part from MP4 file and delivers the required fragment to the client [8]. Figure 10 is showing file format.

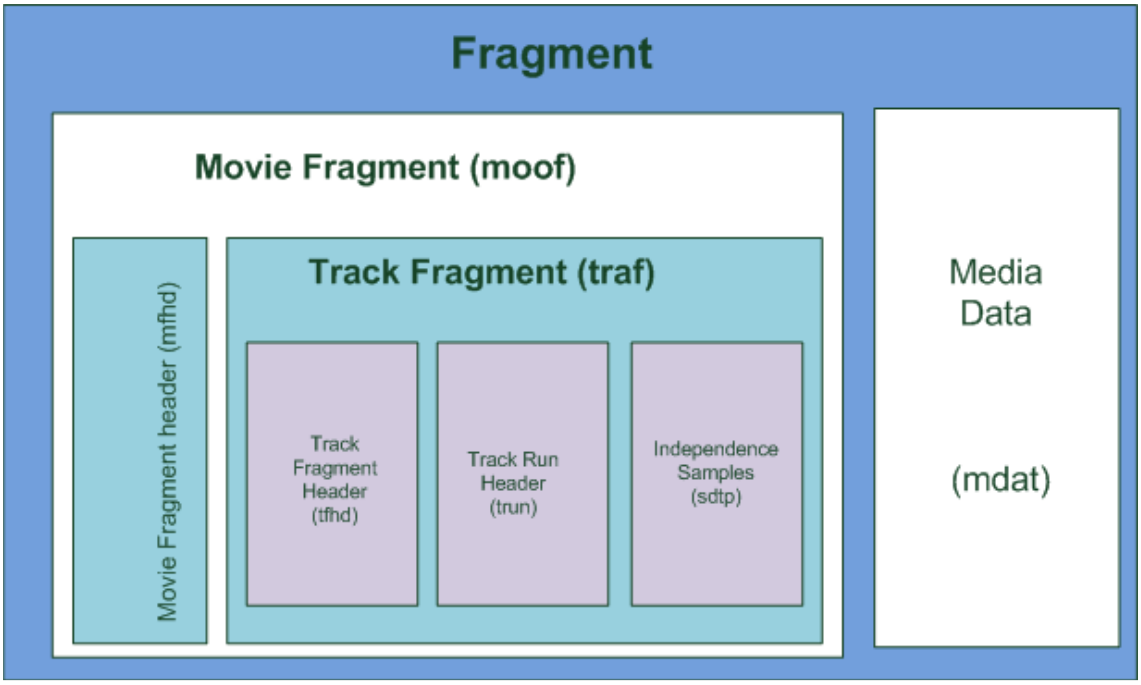

Figure 10: File Format in Smooth Streaming

Following is an example of payload when a player requests to play the stream of media file. A client requests the client manifest file through a server in smooth streaming. Basing on this information, a decoder is initiated by the client at the time of execution. At the same time a playout pipeline of the media is built for playback. On receiving client request, the server examines the related server manifest files and checks the request in MP4 files available on disk. It checks whether a request is for .isma (audio file) or .ismv (video). Then it (server) reads the track fragment index box to finds the actual requested fragment which can be format of movie or data (moof and mdat). After locating exact fragment, it is extracted from MP4 file of disk and delivered to a client. A fragment can be cached on cloud so that client would access same fragment if he/she requested again for it.

There is very small research on the streaming of multimedia contents. In our research we have studied three methods of streaming environments. We have studied working environment of live streaming method used by Apple and smooth streaming method by Microsoft. It has been noticed that both of the technologies are working under standard format defined by the vendor. Formats other than .mov in case of live streaming by Apple and .pm4. Purpose of this study is to check working of existing systems using third party plug-in technology. This was very helpful while embedding player inside our DSPACE.

CHAPTER 4

SYSTEM ARCHITECTURE AND DESIGN

In our study, we have studied two streaming technologies over DSPACE. There is no need of separate streaming server to store and process streams. One is HTML5 which is getting popularity by offering new features embedded inside the browser whereas other technology is player based technology where a media player is embedded inside the webpage to retrieve the stream of multimedia contents. We have DSPACE installed on the Tom cat web server. We have uploaded Quick Time Videos and HTML5 videos on the Portal. Both technologies have been compared in terms of third party plug-in CPU Utilization, and Frame Rate.

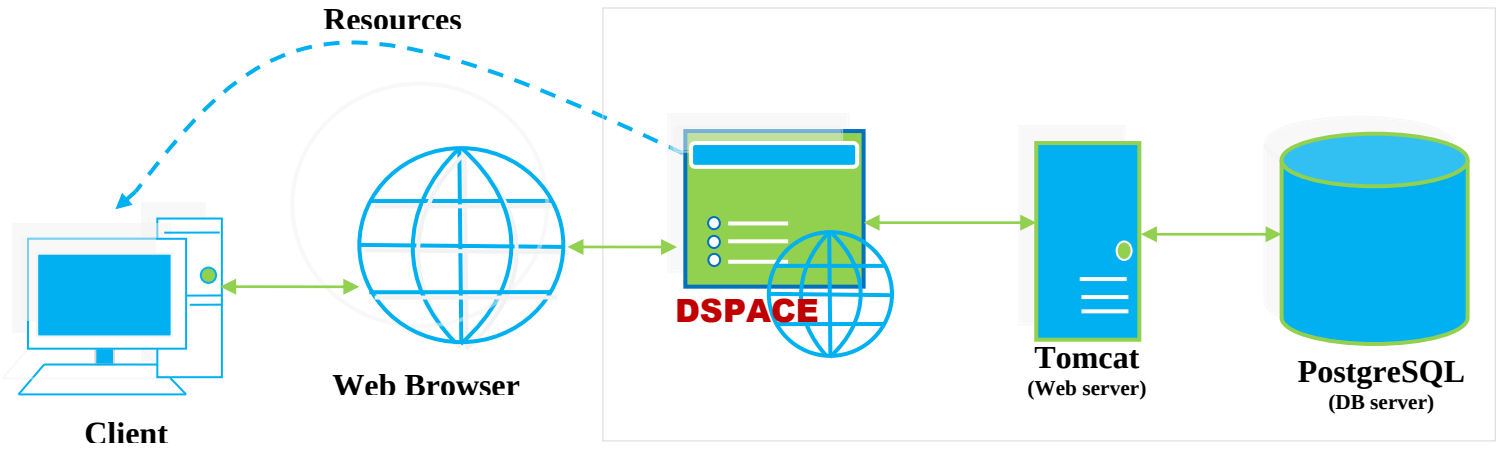

Figure 11: DSPACE Environment

Figure 11, shows the detailed architecture of DSPACE. It shows the working environment of DSPACE with integrated components. There are following components in DSPACE.

Client:

A client is an end user; he/she can be a student, faculty member or any person who want to access data from DSPACE. In IRs Users/Clients are managed through permission rights. User names and passwords are assigned to authorized users so that they can have access to the contents according to the authentication permission they have been granted. There are different categories of the users. These are following categories.

1. Those, who can only view the contents available in DSPACE. They cannot upload or download data.

2. Those, who have rights to upload and download contents from DSPACE but they cannot change the data management policy.
3. Those, who have administrative right and manage the whole system. They are authorized to make any change in system.

Web Browser:

A web browser is an application that is used to access data from DSPACE. There are different browsers which support different types of open data format. We can access the format without using third party plug-in or format compatibility software.

DSPACE Interface:

DSPACE Provides GUI (Graphic User Interface). A user of repository can upload, download and browse the contents using user friendly graphical interface. Authorized user can login to their user account and manage their contents accordingly.

Tomcat:

Tomcat is an open source web server, it is also known as to servlet container. It implements JSP and java servlet specifications. It provides HTTP web services for java code. Configuration and management of web a server can be done using XML. DSPACE uses Tomcat as a web application server. It provides configuration of JSP as well as XML interfaces so IRs can be configured and designed according to the requirement.

PostgreSQL:

PostgreSQL is an open source relational database. In DSPACE PostgreSQL and Oracle can be used as backend. We have used PostgreSQL in our repository as shown in figure 11.

4.1 Plug-in based Streaming Implementation:

In 2002, Macromedia introduced Sorenson Spark for Flash videos. In 2003, X-Flv format was introduced with very high codec of VP6. It had good compression quality as well. Google lunched multimedia based web application named YouTube in 2005. It implemented FLV format exclusively to manage multimedia contents. YouTube Provide simple interface to download and upload videos on the web, it is playing videos on the web flawlessly which is one of the reason that has increased the popularity of the flash player. With this popularity flash has become the de facto standard on the web [20].

But beside popularity on the YouTube, problems with managing online videos were not reduced. Placing flash videos on the web require a depth of knowledge of Adobe action script and proprietary tools to encode videos and other controls of the player. A player also requires some processing at the time of video storage and retrieval. Video is taken from input resources and then it is compared, compressed and edited according to the standards and formats defined in standard of the player.

Following figure 12[1] shows the complex code example of Flash player. This code is embedded in the web page to view content on the flash player. Parameters of video are defined in <param> tag. Following code will play all flash files in flash player.

Flash Video Code

```
<object id="UNIQUEID" height="480" width="320"
codebase=http://download.macromedia.com/ classid="clsid:d27cdb6e-ae6d-11cf-96b8-
444553540000" >
 <param value="../player/myVideoPlayer.swf" name="movie" />
<param value="true" name="allowFullScreen" />
 <param value="all" name="allowNetworking" />
<param value="always" name="allowScriptAccess" />
<param value="opaque" name="wmode" />
<param value="myVideoFile.flv" name="FlashVars" />
<embed height="520" width="528" src="../player/mds_player.swf" id="UNIQUEID"
wmode="opaque" allowscriptaccess="always" allownetworking="all" allowfullscreen="true"
swf="../player/myVideoPlayer.swf" flashvars="myVideoFile.flv
pluginspage=http://www.macromedia.com/go/getflashplayer type="application/x-shockwave-
flash" quality="high" />
</object>
```

Figure 12: Flash Player code example.

Other than this, there is compatibility issue of the software versions at both ends. If at client and server side, two different version of same player are installed, video would not be played. It is necessary to update version of player when higher version is installed at server or client ends.

1. Figure is taken from http://www.ibm.com/developerworks/library/wa-html5video/

We have embedded three players on the DSPACE; Flash Player, Quick Time Player and MP3 player. Following figure13 shows the environment of the player embedded in the DSPACE.

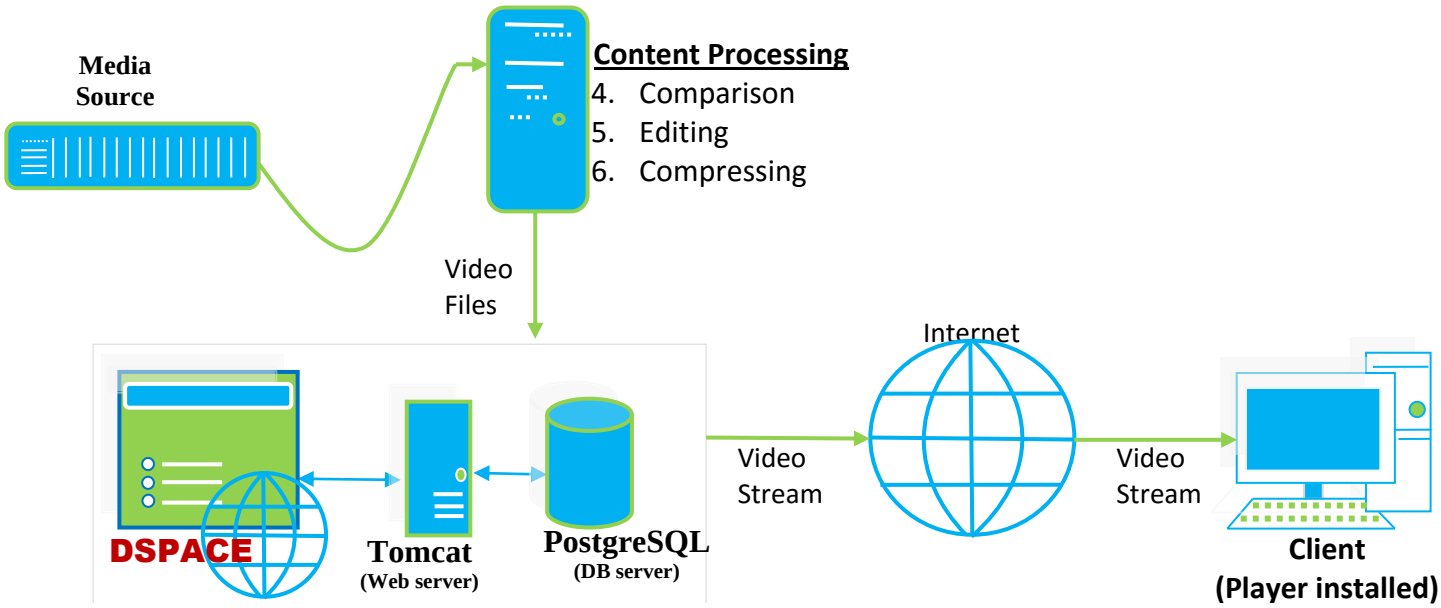

Figure 13: plug-in based Environment in DSPACE

In figure 13, a plug-in based environment has been shown in graphic form. Media source is file to store in DSPACE. Content Processor is controller embedded with player it takes files from input then compare with defined format, edit it (make small packets) and compress it for storage. After processing file is stored in DSPACE and accessed by the user using required web browser. It is necessary to install required player at user end to stream multimedia contents on the web.

4.2 Implementation of Players Technology:

In plug-in based environment there is need of player plug-in at a client end. Video can be added in one particular format. We have registered MIME type on the players like flash players with FLV format, quick media player with MOV and window media player for MP3 and MP4 formats. To add a player in DSPACE following steps are required.

Step1: First of all download the players from website

1. Flash Player: http://get.adobe.com/flashplayer/
2. Mp4 and Mp3 players: http://www.finalmediaplayer.com/

Step 2: Copy all the players in following directory of C:\

C:\DSPACE-3.0-src-release\DSPACE-XMLui\src\main\webapp\themes\Mirage\lib

Step3: open item-view.xsl file from following source

C:\DSPACE-3.0-src-release\DSPACE-

XMLui\src\main\webapp\themes\Mirage\lib\xsl\aspect\artifactbrowser

After these lines in file item view file

```
MP3. Audio Streaming:
<!--video Streaming Code -->
<!--MP3 Streaming code-->
<xsl:iftest="./mets:fileSec/mets:fileGrp[@USE='CONTENT']/mets:file[@MIMETYPE='audio/
mp3']">
  <hr/>
 <Left>
      <h2>Preview Content</h2>
      </left>
<xsl:foreachselect="./mets:fileSec/mets:fileGrp[@USE='CONTENT']/mets:file[@MIMETYPE=
'audio/mp3']">
  <div style="text-align:left">
 <object type="application/x-shockwave-
flash"data="/themes/Mirage/lib/player_mp3_maxi.swf"width="220" height="25">
<param name="movie"value="/themes/Mirage/lib/player_mp3_maxi.swf" />
<paramname="FlashVars"value=".mp3={./mets:FLocat[@LOCTYPE='URL']/@xlink:href}&
amp&amp;showvolume=1"/>
        </object>
      </div>
    </xsl:for-each>
   <hr/>
   <br />
   </xsl:if>
```

Figure 14: MP3 Audio Streaming Code for DSPACE

Figure 14 shows the code we have written in XML to embed mp3 player inside our repository. This code is written in XML to play sound file which are supported by mp3 player. We have defined our parameters in <param> tag. Parameters include size of file to display inside the browser, allow full screen option and volume control icon. The code will call player when MIME type of the file matches.

MP4. Video Streaming:

```
<xsl:iftest="./mets:fileSec/mets:fileGrp[@USE='CONTENT']/mets:file[@MIMETYPE='video/.mp4']">
 <hr/>
<center>
 <h1>Preview content</h1>
      </center>
                                                                                    <xsl:for-eachselect="./mets:fileSec/mets:fileGrp[@USE='CONTENT']/mets:file[@MIMETYPE='video/.mov']">
      <div style="text-align:center;">
 <object type="application/quicktime-movie"data="/themes/Mirage/lib/quicktime player"width="490" height="395">
      <param name="movie"value="/themes/Mirage/lib/quicktime player" />
      <param name="allowFullScreen" value="true"/>
 <param name="MovieVars" value="movie={mets:FLocat/@xlink:href}&amp;width=490&amp;height=395&amp;showstop=1&amp;showvolume=1&amp;showtime=1&amp;showfullscreen=1&>
 <embedsrc="/themes/Mirage/lib/movieplayer" PLUGINSPAGE="http://www.finalmediaplayer.com/" TYPE="application/quicktime-movie" WIDTH="480">
</object>
</div>
 </xsl:for-each>
   <hr/>
 <br />
 </xsl:if>
```

Figure 15: Quick Player DSPACE code

Figure 15 shows the code we have written in XML to embed quick time player inside our repository. This code is written in XML to play quick time video file. We have defined our parameters in <param> tag. Parameters include size of file to display inside the browser, allow full screen option and volume control icon. The code will call player when .mov MIME type of the file matches.

```
Flash Video Streaming:
<xsl:iftest="./mets:fileSec/mets:fileGrp[@USE='CONTENT']/mets:file[@MIMETYPE='video/flv']">
 <hr/>
<center>
 <h1>Video Preview</h1>
</center>
<xsl:foreachselect="./mets:fileSec/mets:fileGrp[@USE='CONTENT']/mets:file[@MIMETYPE='video/flv']">
 <div style="text-align:center;">
<object type="application/shockwave-
flash"data="/themes/Mirage/lib/flv.swf"width="475" height="380">
<param name="movie"value="/themes/Mirage/lib/flv.swf" />
<param name="allowFullScreen" value="true"/>
<paramname="FlashVars"value="flv={mets:FLocat/@xlink:href}&amp;width=475&amp;height=380&amp;showvolume=1&amp;showtime=1&amp;
showstop=1&amp;showfullscreen=1&>
<embedsrc="/themes/Mirage/lib/flv.swf"PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer" TYPE="application/shockwave-flash" WIDTH="485">
</object>
</div>
</xsl:for-each>
<hr/>
<br />
</xsl:if>
```

Figure 16: Flash Player DSPACE Code

Figure 16 shows the code we have written in XML to embed flash player inside our repository. This code is written in XML to play flash file inside the browser. We have defined our parameters in <param> tag. Parameters include size of file to display inside the browser, allow full screen option and volume control icon. The code will call player when MIME type of the file matches.

Step4: Rebuild DSPACE especially Maven and ANT Directories.

4.3 HTML5 Technology:

HTML5 is newer version of Hyper Text Markup Language. It has <video> tag which lets a web developer to play videos on the web without using any third party plug-in. <video> tag is embedded on the Web page.

Recently, there are many internet browsers which support HTML5. In upcoming years HTML5 will replace and reduce the application of a media player technology on the web [18]. Other than the support of the web browsers, HTML5 is providing new features of simple interface, offline storage, security services and many more. Figure 17[2] shows basic <video> tag that is used to play video inside the browser without embedding third party plug-in.

```
<video controls src="video.mp4"></video>
```

Figure 17: HTML5 Video Tag

4.5 Implementation of HTML5 Technology:

In HTML5 based environment there is no need of 3rd party Plug-in at client end. Code is embedded inside the web and video can be accessed through particular browser. Now google chrome supports all most all functions of HTML5. Figure18 shows the working environment of HTML5 technology in DSPACE. There is no need of third part plug-in. We need to embed XML in webpage at server side. To implement HTML5 code in DSPACE following steps are required.

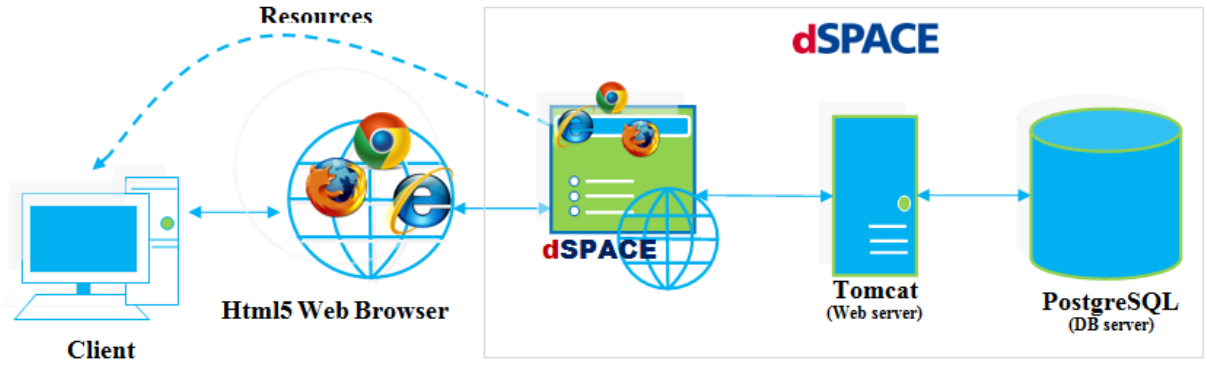

Figure 18: HTML5 Code Environment in DSPACE

Step1: Add the .mp4 to the metadata registry using the instructions found here —> Add a new format to the bitstream registry.

Step 2: To modify the header, locate the section of DIM-Handler.xsl that begins with following code.

[2] Figure is taken from http://www.ibm.com/developerworks/library/wa-html5video/

```
<xsl:template match="dim:dim" mode="itemSummaryView-DIM">
```

Step3: open item-view.xsl file from following source. Immediately following the line that reads <table> adds the following code: Figure 19 shows the line in item view.xsl file.

```
<table class="ds-includeSet-table">
```

Figure 19: Table tag in XML code of DSPACE

Step4: add the following code:

```
<xsl:choose>
    <xsl:when
test="ancestor::mets:METS/mets:fileSec/mets:fileGrp[@USE='CONTENT']/mets:file
[@MIMETYPE='video/mp4']">
            <tr class="ds-table-row odd">
                    <td><span class="bold"><i18n:text>xmlui.dri2xhtml.METS-
1.0.item-preview</i18n:text>:</span></td>
                    <td>
                            <xsl:attribute name="href"><xsl:value-of
select="@OBJID"/></xsl:attribute>
                            <video>
                                    <xsl:attribute name="src">
                                            <xsl:value-of
select="ancestor::mets:METS/mets:fileSec/mets:fileGrp[@USE='CONTENT']/mets:fi
le/mets:FLocat[@LOCTYPE='URL']/@xlink:href"/>
                                    </xsl:attribute>
                                    <xsl:attribute name="width">
                                            <xsl:value-of select="320" />
                                    </xsl:attribute>
                                    <xsl:attribute name="height">
                                            <xsl:value-of select="240" />
                                    </xsl:attribute>
                                    <xsl:attribute name="controls" />
                                    <xsl:attribute name="autobuffer" />
                            </video>
                    </td>
            </tr>
            </xsl:when>
            <xsl:otherwise>
            </xsl:otherwise>
        </xsl:choose>
```

Figure 20:HTML5 code in DSPACE

Figure 20 shows the code we have written in XML to use HTML5 technology in DSPACE. This code is written in XML to play video inside the browser without using any third party plug-in. We have defined required attributes in <video> tag and <attribute>tag as sub tag inside the video tag. Attributes include size of file to display inside the browser, allow control and auto buffer option. The code will be called when MIME type of the file matches.

Step5: Rebuild DSPACE
Step6: Add and Retrieve Video mp4 without using third party plug-in

4.6 Storing and Capturing Context

In textual based storage contents several access methods can be applied by using texts data whereas in visual contents there is a need of extra information to access visual data. In DSPACE, to capture context we have stored abstract of each element at the time of storage. On abstract we have created index discovery so that user would be able to retrieve contents using information stored in index. This discovery method is like Amazon use to match brand and price of a request by client. With visual contents, it would be very powerful method in browsing and retrieving of visual contents in DSPACE. Traditional image processing techniques and algorithms require complex programming and deeper knowledge of types and format of visual contents. Other than this, each type of content has a different processing requirement which leads to write separate algorithms for each type of contents. As in DSPACE data is ingested from different resources so it is hard to know exact requirements at design timing. So our purposed mechanism would cater a lot in the processing of integrated visual contents environment.

Implementation of indexing on visual Data Fields in DSPACE:

Following steps are required to implement indexing in DSPACE

Step1: To enable index open "XMLui. xconf" file

Step2: Remove comments from Discovery Aspect in the file

Step3: To update configuration open "DSPACE.cfg" file.

```
<xmlui>
    <aspects>
        <!--
            @deprecated: the Artifact Browser has been devided into
ViewArtifacts,
            BrowseArtifacts, SearchArtifacts
            <aspect name="Artifact Browser"
path="resource://aspects/ArtifactBrowser/" />
        -->
        <aspect name="Displaying Artifacts"
path="resource://aspects/ViewArtifacts/" />
        <aspect name="Browsing Artifacts"
path="resource://aspects/BrowseArtifacts/" />
        <!--<aspect name="Searching Artifacts"
path="resource://aspects/SearchArtifacts/" />-->
        <aspect name="Administration"
path="resource://aspects/Administrative/" />
        <aspect name="E-Person" path="resource://aspects/EPerson/" />
        <aspect name="Submission and Workflow"
path="resource://aspects/Submission/" />
    <aspect name="Statistics" path="resource://aspects/Statistics/" />

        <!--
         To enable Discovery, uncomment this Aspect that will enable it
            within your existing XMLUI
            Also make sure to comment the SearchArtifacts aspect
            as leaving it on together with discovery will cause UI
overlap issues-->
        <aspect name="Discovery" path="resource://aspects/Discovery/" />


        <!--
            This aspect tests the various possible DRI features,
            it helps a theme developer create themes
        -->
        <!-- <aspect name="XML Tests"
path="resource://aspects/XMLTest/"/> -->
    </aspects>
```

Figure 21: enabling Discovery option

Figure 21 shows the code to enable discovery option in “XMLui. xconf” file . Discovery options are disabled by default. To implement discovery it is necessary to uncomment already defined discovery aspect in DSPACE.

Step4: Add Abstract to discovery list.

```
# default synchronous dispatcher (same behavior as traditional DSpace)
event.dispatcher.default.class = org.dspace.event.BasicDispatcher
#event.dispatcher.default.consumers = versioning, search, browse,
abstract, harvester
event.dispatcher.default.consumers = versioning, search, browse,
discovery, abstract, harvester
```

Figure 22: Adding index field to discovery list

Figure 22 shows how to add abstract in discovery list. To enable discovery on any field it is required to add particular field in the discovery list of Event dispatcher and synchronous dispatcher as shown in figure 22 .

Step 5: Clearing submission count to zero so that indexing could be applied to all contents already stored in DSPACE.

```
#Put the recent submissions count to 0 so that discovery can use it's
recent submissions,
# not doing this when discovery is enabled will cause UI overlap issues
#How many recent submissions should be displayed at any one time
#recent.submissions.count = 5
recent.submissions.count = 0
```

Figure 23: Counter Clearance Code

Figure 23 shows the process of clearing counter to zero. Purpose of clearing counter is to implement index on all stored items. If counter is not set to zero then changes on index will be effective for new submission only.

Step 5: Setting Index port to 8080 in "discovery.cfg"

```
##### Search Indexing #####
solr.search.server = http://localhost/solr/search
```

Figure 24: Port setting Code

Figure 24 shows the code to set index port to 80. By default solar does not run at port 80.

Step 6: Update Index of DSPACE with command

```
[dspace]/bin/dspace update-discovery-index
```

Figure 25: Update Index Command

Figure 25 shows the method of updating index. Update-discovery-index would rebuild the index.

In this chapter we have implemented two technologies in DSPACE. We have modified existing view methods of multimedia contents. Two players have been added in viewSummery file of DSPACE. This file displays the content of the DSAPCE inside the browser. We have embedded player codes and HTML5 codes inside browser. We have changed simple HTML code of players used in web pages into XML. Other than embedding players we have added index on abstract field of file. Full text search is possible by using this code.

:

CHAPTER 5

EXPERIMENTAL RESULTS AND EVALUATION

In this Chapter we have defined benchmark of two streaming technologies and analyzed how HTML5 even being open standard suitable for DSPACE and other IRs.

5.1 Motivation

In analysis we have compared two streaming technologies with each other regardless streaming server. There are many similarities and differences between each other as well as merits and demerits. There are many aspects in which both are better to each other. We have analyzed few parameters according to the requirement of the IRs. We have set benchmarks on which evolution of the both of the technologies have been tested. We have compared our findings with already existing tests to get to know how particular technology works with other web applications as compared with DSPACE. There are following bench marks on which technologies were compared.

1. Frame Rate
2. CPU and Memory Utilization

5.2 Procedure

To test the benchmarks we have used the following procedure.

1. Step1: I have cleared the cache of my browsers, updated Graphic Drivers and closed all running processes on Browsers as well as system.
2. Step 2: Load the video in browser when the video is streamed. Then played any part of the video.
3. Step3: When record the CPU monitoring to capture CPU utilization graph and Fraps to capture the frame rates.

Following software were used to collect results for analysis

4. **FRAP:** It is benchmarking application that is used to capture the frame rate per second.
5. **CPU Mon:** It is also benchmarking application to define utilization of CPU in computer system and consumption of battery in hand held devices.

To test the system following Browsers were used

1. IE (Internet Explorer)
2. Google Chrome

Data Set:

We have stored video of 8MB in DSPACE with 800X 600 resolutions. Same video is stored in two formats; quick time format for quick time player and mp4 format for HTML5 player.

5.3 Frame Rate:

As video is comprises upon continuation of the frames. In animation and video, frame rate is taken in two contexts first one rendering of the frames(quick access of contents) and second one is quality metric (high quality output). Here in our analysis we are more concerned with accessing contents as compare to the quality of the contents. FRS defines the number of frames that can be rendered per second at client side. It is recommended that frame rate must be in between 25 to 3o fps basing upon the codec of PAL and NTSC. There are more chances of system crash when Fps rate increase and at low rate. Some software allows a player to use GPU(graphical processing unit) in rendering which increase the render rate and become cause of system crash.

Table 2:FRS at cable network connection

Browser	Frames Rate	
	Player	HTML5
Internet Explorer	26	24
Chrome	28	26
Fire Fox	27	25

Table 2 shows the data set of frame rate in cable network connection. On x-axis there are three players where as on y-axis frame rendering rate of each browser is shown. Frame rate of both technologies is compared. As we have tested data set in three browsers which are using two technologies. Figure 26 shows the graphical representation of the results. Blue bar shows render rate of player technology where as green bars are representing frame rate renders at HTML5.

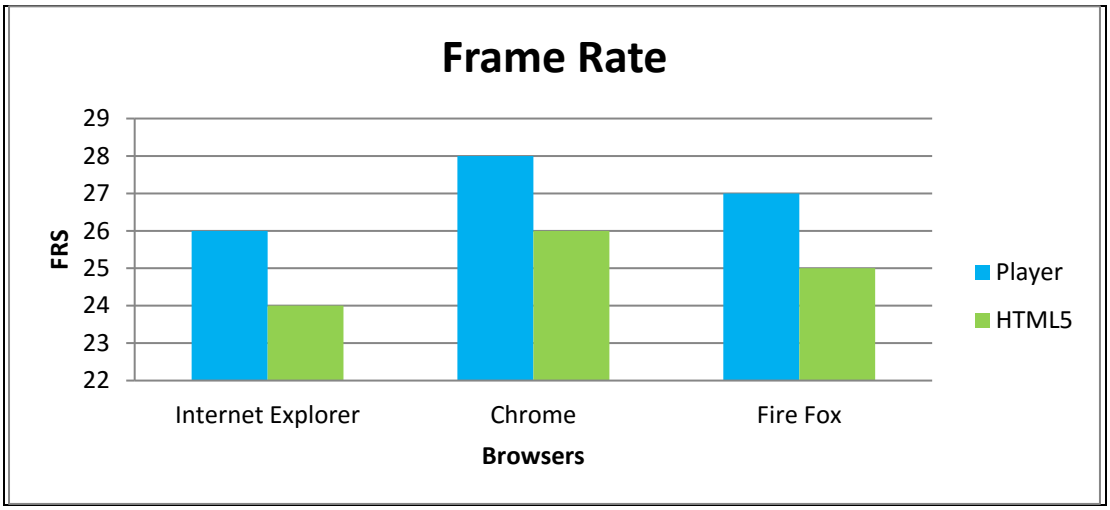

Figure 26: FRS at cable network connection

In cable network we have captured FPS

- In IE browser FPS is 26 where as in HTML5 rate is 24 which is slightly better than player technology.
- Where as in Chrome Browser FRS is 26 in HTML5 and 28 in player.
- In Firefox HTML5 is at 25 FRS and player is 27.

In all browsers HTML5 is better from player although there is slight difference.

We have test frame rates of both Technologies in three different Browsers as shown in figure 26. In our analysis although there is bit difference between FRS of both of the technologies but this slight difference counter a lot when size of video is increased. There is almost same difference with respect to the browsers. In Internet Explorer HTML5 have 24 FPS with the comparison of 26 Frames on the Player where as in Chrome and Fire Fox FRS is 26 and 25 respectively with the comparison of 28 and 27.

Table 3: FRS in wireless connection

Browser	**Frames**	
	Player	HTML5
Internet Explorer	25	19
Chrome	31	24
Fire Fox	31	35

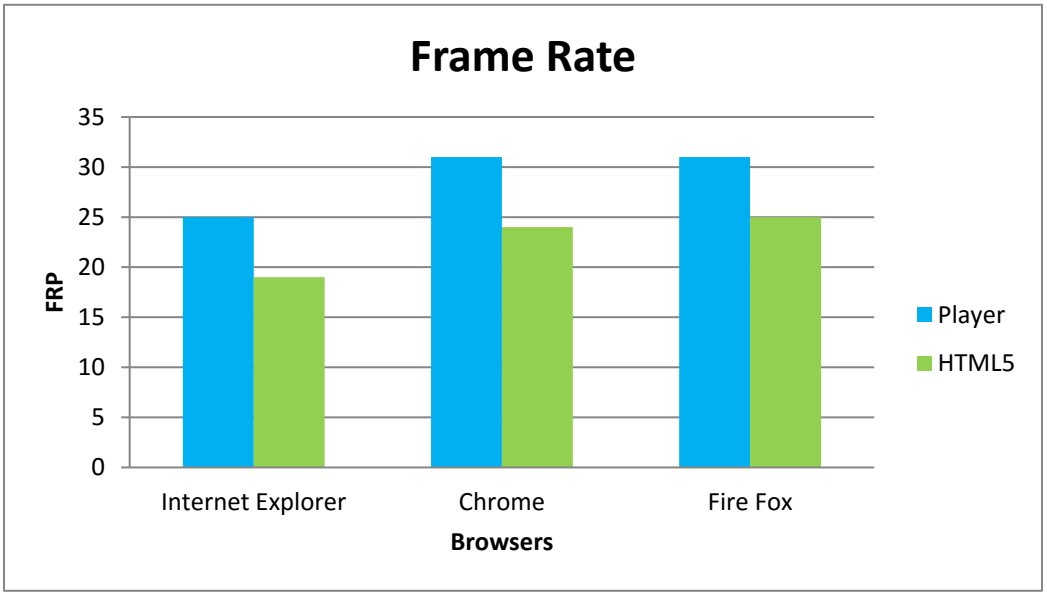

Figure 27: FRS in wireless connection

In our analysis almost in all browsers same difference between frame rates is same. Table 3 shows tabular form of result whereas figure 27 is representing graphical representation of data. In wireless network we have captured FPS

- In IE browser FPS is 25 where as in HTML5 rate is 19 which is quite better than player technology.
- Where as in Chrome browser FRS is 24 in HTML5 and 24 in player.
- In Firefox HTML5 is at 31 FRS and player is 35.

HTML5 is better in IE and Chrome where as Player is better at Fire Fox in wireless connection.

We have also study the comparison from different resources where high resolution multimedia applications are analyzed. In table 4[25], the HTML5 and Flash player has been tested in term of FRS. It is analyzed the more FPR rate there are more chances of crash. In table4 [26], at user blogs there are more than 77tests results of FRS where flash frame rate is more 100 and as result it crash the browser.

Table 4: Comparison of FRS online-I

Browser	Frames	
	Player	HTML5
Internet Explorer	56	-
Chrome	56	28
Fire Fox	56	19

Table 5: Comparison of FRS online-II

Browser	Frames	
	Player	HTML5
Internet Explorer	51	-
Chrome	51	20
Fire Fox	51	27

As conclusion from our results and results taken from other online available sites that HTML5 is better in FRS in DSPACE its frame rate is lower than player which means video plays smoothly and there are less chances of jerk and fast motion. But rate should between 25 to 30 frames not less than this required rate. High speed motion and Frame Rates are considered good in fast motion animated movies or online games where as for institutional information short and smooth motion videos are recommended so that contents in video would be listened and viewed properly.

5.4 CPU Utilization:-

CPU Utilization is referred to as processing of video at a user end. In online streaming through streaming server CPU processing power is less as videos are processed at the server side where as in Plug-in and HTML5 technology video processing is done at client's end. More CPU power means less processing at server side and processing at a client side which can be viewed as advantage at server side as at the client side at time only or few threads are being processed where as at server side there are many requests to access same or other object.

Table 6, table 7 and table 8 show tabular form of the result. It shows the percentage of the CPU utilization in both technologies, rendering frames of video on DSPACE. Figure28, figure29 and figure 30 show the graphical form of results. Green bar are result of HTML5 file rendering and blue is result of player technology.

Table 6: CPU Utilization in cable network

Browser	CPU Utilization	
	Player	HTML5
Internet Explorer	59%	96%
Chrome	95%	76%
Fire Fox	75%	89%

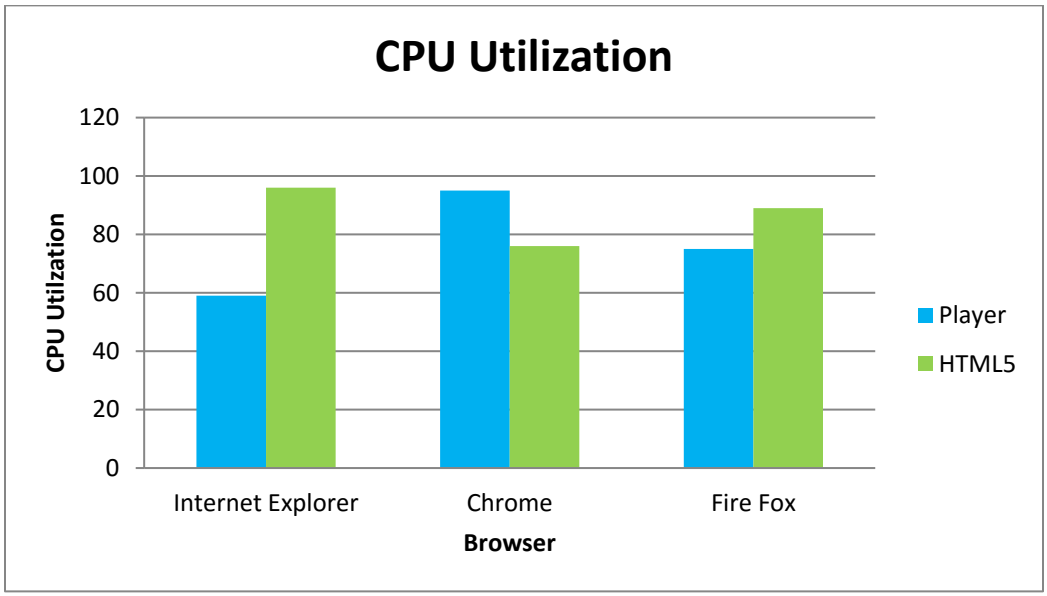

Figure 28: CPU Utilization in cable network

In figure28, utilization of CPU is plotted using cable network.

- In IE browser CPU utilization 59 % (player)as it allows GPU where as in HTML5 percentage is 96 % the difference is 37%, which shows player is 37 times better than HTML5 technology.
- In Chrome browser CPU utilization 95 %(player) as it allows GPU where as in HTML5 percentage is 76 % the difference is 19%, which shows player is 19 times better than HTML5 technology.

- In Firefox CPU utilization 75 %(player) as it does not allows GPU where as in HTML5 percentage is 89 % the difference is 14%, which shows HTML5 is 14 times better than player technology.

Table 7: CPU Utilization in wireless network

Browser	CPU Utilization	
	Player	HTML5
Internet Explorer	55%	96%
Chrome	81%	78%
Fire Fox	85%	92%

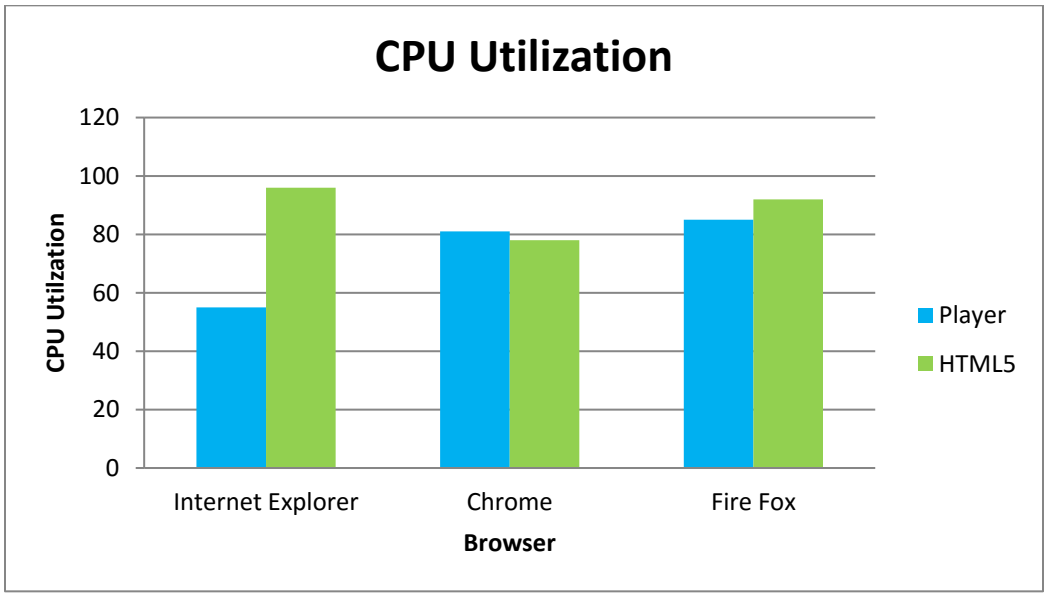

Figure 29 : CPU Utilization in wireless network

In figure29, utilization of CPU is plotted using wireless network.

- In IE browser CPU utilization 55 % (player)as it allows GPU where as in HTML5 percentage is 96 % the difference is 41%, which shows player is 41 times better than HTML5 technology.
- In Chrome browser CPU utilization 81 %(player) as it allows GPU where as in HTML5 percentage is 78 % the difference is 3%, which shows HTML5 is 3 times better than player technology.

- In Firefox CPU utilization 85%(player) as it allows GPU where as in HTML5 percentage is 92 % the difference is 7%, which shows player is 14 times better than HTML5 technology.

Table 8: CPU Utilization without using video card devices

Browser	Frames	
	Player	HTML5
Internet Explorer	98%	96%
Chrome	98%	78%
Fire Fox	98%	92%

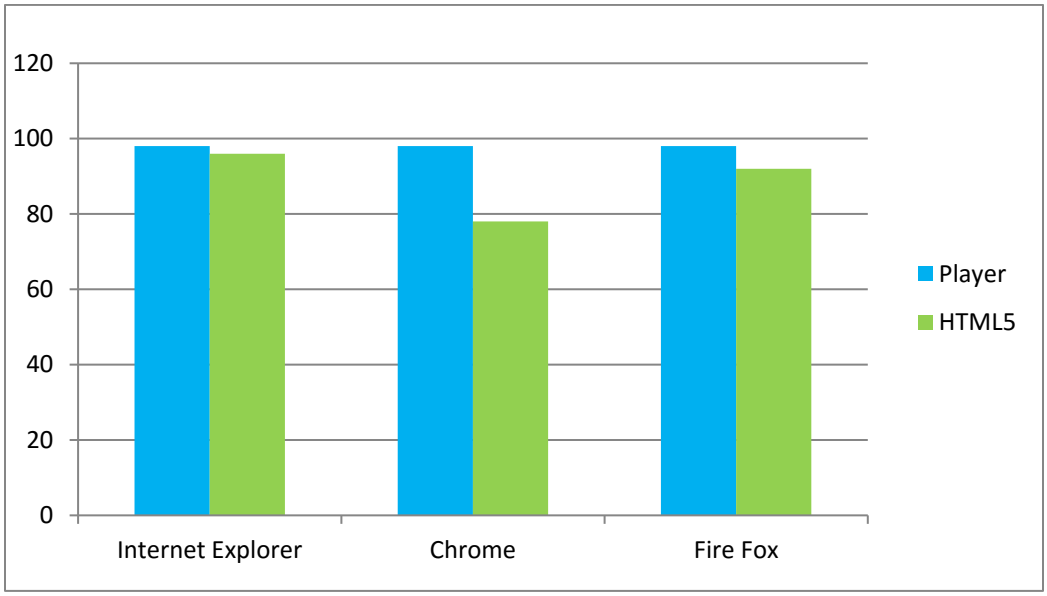

Figure 30: CPU Utilization without using Video Card Device

Figure 30 shows utilization of CPU without using GPU(video card)

- In IE browser CPU utilization 98 % (player) as it allows GPU where as in HTML5 percentage is 96 % the difference is 2%, which shows HTML5 is 2 times better than player technology.
- In Chrome browser CPU utilization 98 %(player) as it allows GPU where as in HTML5 percentage is 78 % the difference is 20%, which shows HTML5 is 20 times better than player technology.

- In Firefox CPU utilization 98%(player) as it allows GPU where as in HTML5 percentage is 92 % the difference is 6%, which shows HTML5 is 7 times better than Player technology.

We have compared both technologies with respect to processing power of CPU there are different results at different browsers. In our results IE uses 59% with the comparison of the HTML5 whereas in chrome Video Graphic Card is disabled in Google Chrome so utilization of CPU is more in HTML5 as compare to Player. In Fire Fox CPU utilization is 89% with comparison of 75%. Figure 25 shows the result of CPU utilization without using video card. The figure shows that utilization of CPU accelerates when there is no card rendering in player based plug-in.

It is cleared that plug-in based technology uses the system graphic card acceleration which can store the processing power of the system and use less CPU power consumption so the systems which allows plug-in like flash player; video is more efficient than HTML5 video the basic reason is that it's accelerated by the graphics hardware of the PC. Jan Ozer [24] says "So a system where graphics card is installed flash is efficient and without graphic card it is inefficient."

We have tested the processing power with graphic card on system so flash is efficient at IE

MacBook Pro - YouTube	Safari - HTML5	Safari - Flash	Chrome - HTML5	Chrome Flash	FireFox Flash
CPU utilization Flash Player 10.0	12.39	37.41	49.89	50.39	40.25
Flash 10 vs HTML5		202%		1%	
CPU utilization Flash Player 10.1	12.39	32.07	49.89	49.79	42.07
Flash 10.1 vs HTML5		159%		0%	
Change from 10.0 to 10.1		-14%		-1%	5%

HP 8710p/Windows	Safari - HTML5	Safari - Flash	Chrome - HTML5	Chrome Flash	FireFox - Flash	IE - Flash
CPU utilization Flash Player 10.0	Did not Play	23.22	25.66	19.55	22.00	22.41
Flash 10 vs HTML5		NA		-24%	NA	NA
CPU utilization Flash Player 10.1	Did not Play	7.43	25.66	10.73	6.00	14.62
Flash 10.1 vs HTML5		NA		-58%	NA	NA
Change from Flash 10.0 to Flash 10.1		-68%	NA	-45%	-73%	-35%

Figure 31: Jan Ozer's analysis at Streaming Learning Center

Jan Ozer at streaming learning center [24], has tested CPU utilization at both MAC and Windows environment. As shown in figure31, Mac use 12.39% with compared 37.41 %. According to survey Ozer believes that in Mac HTML5 is 202% more efficient than Flash. While

on other hand in Windows HTML5 is 24% behind the utilization of the CPU. They also compared the videos using different version of the flash where CPU utilization is improving gradually in new versions. From these analysis it is shown that flash is better in term of utilization of CPU where there are multimedia supportive tools embedded.

Comparing these analyses with DSPACE we conclude that as contents of DSPACE are not achieved and processed for high accelerated machines so HTML5 would works. These contents are stored on the Web so that people would have access to the contents online regardless highly multimedia and accelerated machine. Our aim is contents should be available at any device whether it is mobile or simple computer system even in low bandwidth areas.

CHAPTER 6

CONCLUSION AND FUTURE DIRECTIONS

This chapter includes conclusion and future work of this thesis.

6.1 Conclusion

We have compared two technologies of Streaming; Plug-in based and HTML5.In plug-in a player is embedded at client side as well as at server side where as in HTML5 technology there is no need of third party plug-in. Both technologies have different advantages over each others. None of them is perfect in term of performance, quality and availability but in some particular situation HTML5 is better than plug-in and vice versa.

In DSPACE, HTML5 is better in terms as data is not generated at single point. It is retrieved from several resources in different formats. So if one player is installed at DSPACE all incoming and outgoing data should match with existing installed plug-in which is quite challenging task. Accessing data in one format and changing into another format require adding of new layer to perform processing. Secondly configurations of the player according the requirement of IR need licensing.

In our results we have seen HTML5 videos are working better than plug-in based technology in wireless network. It means that multimedia data stored on DSPACE can be accessed on low bandwidth. It is also required because we need online lectures or videos through mobile devices when we are away from institutions and have not access to high bandwidth. This can be helpful in distance learning systems in far from rural areas which have low band width access.

HTML5 takes less timing in loading contents from the web server as compare to plug-in technology. As it downloads objects as the texts files are loaded to the browser. Further, HTML5 do not use GPU systems so it can process same results of video processing on high quality graphic enabled devices as well as on the machines which do not have graphic acceleration devices.

Finally we have compared the frame rate rendered of both technologies in browser. Flash has high frame rate as compare to the HTML5. But 25 to 30 frames rendering is considered as best choice. In our results we have seen in plug-in based technology frame render rate exceeds to 60 frames per second which become the cause of system crash. So by using HTML5 there are fewer chances of DSPACE crashes while rendering the frame of video objects.

As conclusion we can say no technology is inefficient both are working well in their own domain. But in environment like DSPACE, HTML5 can be implemented efficiently. It can integrate the access of all formats of data onto the single platform.

6.2 Future Directions

HTML5 is proving to be an emerging technology in the world of the web application development. Even dominating multimedia applications like YouTube, Microsoft, Google, IBM and Face book have also started to implement this technology into their applications. There are following future directions for IRs to implement HTML5 technology.

We can create HTTP streaming server using this technology for live video lectures and online video conferences instead of implementing cost effective streaming server. In this way there would be no need of the third party plug-in to play the videos. In this way videos can be stored and retrieved as textual contents are stored and retrieved over HTTP. We can implement real time chat functionality of HTML5 to interact with community. Instead of using blogs or mailing list people would be able to discuss their problems online. They do not need to wait for reply of the post on mailing list.

Furthermore, there are also many other functions which we can be used at DSPACE. As we have added abstract with the video and then made index on the abstract to retrieve the semantics of the video whereas in HTML5 there is semantic Tag which can be used for indexing and metadata storage purposes. It also offers the offline storage capabilities. It means storing data locally within a client side browser. It works like cookies where data persists even after we navigate site, close browser tab, and exit browser etc. It is not transmitted to remote server automatically we have to do it manually. So this feature can be implemented in DSPACE for offline storage and retrieval of contents. DSPACE also provides mobile device interface and HTML5 is also providing efficient services on mobile devices so in future HTML5 would be the finest option for DSPACE mobile interface application.

References

1. F. Kozamernik “Media Streaming over the Internet.” October, 2002
2. B. Urgaonkar. “Dynamic Resource Management in Internet Hosting Platforms”. PhD. Thesis, University of Massachusetts Amherst, India, September 2005.
3. S. Acharya and B. Smith. “An Experiment to Characterize Videos Stored on the Web”. In Proceedings of the ACM/SPIE Multimedia Computing and Networking (MMCN), pages 166 – 178, San Jose, CA, Jan. 1998.
4. M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon. “I Tube, YouTube, Everybody Tubes: Analyzing the World’s Largest User Generated Content Video System”. In Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, pages 1–14, San Diego, California, USA, 2007.
5. M. Chesire, A. Wolman, G. Voelker, and H. Levy. “Measurement and Analysis of a Streaming Media Workload”. In Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS), pages 1–12, San Francisco, CA, USA, Mar. 2001.
6. F. Duarte, F. Benevenuto, V. Almeida, and J. Almeida. “Geographical Characterization of YouTube: a Latin American View”. In Web Conference, 2007. LA-WEB 2007. Latin American, pages 13–21. IEEE, 2007.
7. P. Gill, M. Arlitt, Z. Li, and A. Mahanti. “YouTube Traffic Characterization: A View from the Edge”. In Proceedings of the ACM Internet Measurement Conference (IMC), San Diego, CA, USA, Oct. 2007.
8. M. Li, M. Claypool, R. Kinicki, and J. Nichols. “Characteristics of Streaming Media Stored on the Web”. ACM Transactions on Internet Technology (TOIT), 5(4):601–626, Nov. 2005.
9. E. Rescorla, and N. Modadugu, N. “Datagram Transport Layer Security Version 1.2.”RFC 6347 (Proposed Standard), Jan. 2012.
10. M. Castro, P. Druschel, A. M. Kermarrec, Nandi, A. Rowstron, and A.Singh. “Split stream: High-bandwidth content distribution in cooperative environments” In Proceedings of IPTPS03, Feb. 2003.
11. A. J. Meyn. “Browser to Browser Media Streaming with HTML5”. Master’s Thesis, Technical University of Denmark, DTU Informatics, Denmark, 2002.

12. E. Veloso, V. Almeida, W. Meira, A. Bestavros, and S. Jin. “A Hierarchical Characterization of a Live Streaming Media Workload”. In Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment, pages 117–130. ACM, 2002.
13. M. Zink, K. Suh, Y. Gu, and J. Kurose. “Watch Global, Cache Local: YouTube Network Traces at a Campus Network -Measurements and Implications”. In Proceedings of the SPIE/ACM Multimedia Computing and Networking (MMCN), 2008.
14. Y. Bechqito.” High Definition Video Streaming Using H.264 Video Compression” Master’s Thesis. Metroplia Information Technology Multimedia Communications Finland. 2009
15. T. Roscoe. B. Lyles. “Distributing Computing without DPEs: Design Considerations for Public Computing Platforms”. In *Proceedings of the Ninth ACM SIGOPS European Workshop, Kolding, Denmark. Sep, 2000.*
16. M. Aron. “Differentiated and Predictable Quality of Service in Web Server Systems” Tech. rep., Rice University, October 2000.
17. M. Lin “Techniques And Protocols For Distributed Media Streaming” Phd. Thesis, University of Singapore, 2007.
18. I. S. Ilias, S. B. Munisamy, N. Z. Rahman. “A Study of Video Performance Analysis between Flash Video and HTML 5 Video” In Proceeding of 7th International Conference on Ubiquitous Information Management communication, NY, USA, 2013.
19. R. Karki. “Fresh Analysis of Streaming Media Stored on the Web”. Msc. Thesis, Worcester Polytechnic Institute, Jan. 2011.
20. “Flash VS Html 5 Analysis” Internet: http://www.sbp-romania.com/Articles/flash-vs-HTML5-analysis.aspx, accessed , May.2010 [July, 27, 2013].
21. S.Jobs. “Thought of Apple” Internet: http://www.apple.com/hotnews/thoughts-on-flash/ ,April,2010[Aug.,2,2013]
22. V. Nichols. “Opinion: HTML5: Less than it's cracked up to be”. Internet: http://www.computerworld.com/s/article/9174323/Opinion_HTML_5_Less_than_it_s_cracked_up_to_be?pageNumber=1, Mar. 29,2010[Aug.6,2013]

23. J. James “HTML 5 Editor Ian Hickson discusses features, pain points, adoption rate, and more”. Internet:http://www.techrepublic.com/blog/software-engineer/html-5-editor-ian-hickson-discusses-features-pain-points-adoption-rate-and-more/ ,Dec.22,2010[Aug.12,2013]
24. J. Harman. “HTML5 vs. Flash: The Video Benchmark Death match”. Internet: http://gizmodo.com/5490205/HTML5-vs-flash-the-video-benchmark-deathmatch, Mar. 10,2010[Aug.14,2013]
25. E. L. Guitard. “Flash Vs HTML5 Performance”. Internet:http://pacoup.com/2011/02/03/flash-vs-HTML5-performance/, Feb. 3, 2011[Aug.14,2013].
26. "HTML5" versus Flash: Animation Benchmarking”. Internet: http://themaninblue.com/writing/perspective/2010/03/22/, Mar. 22, 2011 [Aug.19, 2013].