Master Thesis
Electrical Engineering
Nov2011

Network Performance of HTML5 Web Application in Smartphone

Saroj Sharan Regmi
Suyog Man Singh Adhikari

School of Computing
Blekinge Institute of Technology
371 79 Karlskrona
Sweden

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

Contact Information:
Author(s):
Saroj Sharan Regmi
E-mail: sashar18@gmail.com
Suyog Man Singh Adhikari
E-mail: suyog55@hotmail.com

University advisor(s):
David Erman
School of Computing
Blekinge Institute of Technology
371 79, Karlskrona, Sweden
E-mail: der@bth.se

University Examiner(s):
Patrik Arlos
School of Computing
Blekinge Institute of Technology
371 79, Karlskrona, Sweden
E-mail: patrik.arlos@bth.se

School of Computing
Blekinge Institute of Technology
371 79 Karlskrona
Sweden

Internet : www.bth.se/com
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

ABSTRACT

Hypertext markup language 5 (HTML5), a new standard for HTML, enriched with additional features is expected to override all the basic underlying overhead needed by other applications. By the advent of new extension, HTML5, the web's basic language is transplanted from a simple page layout into rich web application development language. Furthermore, with the release of HTML5, traditional browsing is expected to change and modify accordingly and on the other hand the potential users will have an alternative rather than sticking in platform and OS dependent native applications.

This thesis deals with the readiness assessment of HTML5 with regard to different smart phones- Android and Windows. In order to visualize the fact, we analyzed different constraints like DNS lookup time, page loading time, memory and CPU consumption associated with two applications-Flash and HTML5 running right through the smart phones.

Furthermore, the comparative analysis is performed in different network scenarios- Wi-Fi and 3G and user experience is estimated based on network parameters. From the experiments and observations taken, we found that android phones provide better support for HTML5 web applications than windows mobile devices. Also, the HTML5 applications loading time is limited by the browser rendering time rather that the content loading time from the network and is also dependent on hardware configuration of device used.

Keywords: HTML5, Flash, DNS lookup time, CPU, Memory, Page loading, 3G, Wi-Fi.

ACKNOWLEDGEMENT

We are grateful to our parents for their enormous support, encouragement and blessing which strengthen our ability to complete this task.

We would like to express our heartfelt gratitude to our supervisor Dr. David Erman, School of Computing, Blekinge Institute of Technology, for his advice and support. Without his guidance and instruction this thesis work would not reach so far.

At last but not the least, we would also like to thank Dr. Patrik Arlos for granting us scenario for the thesis work and Ravichandra Kommalapati for his technical support and friendly environment. We would also like to thank Mr. Seth Fogie of AirScanner for providing us with free two years license for the use of MobileSniffer tool for our purpose of study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

In the early stage of computing models, most of the applications were distributed among the server and the client. In fact the application load was visualized as a collective pair of server code and client code. Any upgrade in the server code required simultaneous upgrade in client side codes installed on each client terminal. By the advent of web applications, universal client as known as web browser was responsible for interpretation and display of pages. Yet another terminology like client side scripting, JavaScript, evolved which override the earlier convention of sending all the data to server for processing. Most of the applications in the internet are dependent on the server side logic and clients use browsers to view web content. Previous deployment of web application was carried out in the server side and not much effort was carried out that provide service logic on the client applications. The vector animation player, such as flash, introduced some provision of adding plug-in in the browser, which literally required no intervention or communication with the server. But this added extra overhead in client side. Developer started to look for paradigm shift that can enhance and add some additional features, which ultimately led to the evolvement of new standard of HTML, HTML5 which practically avoided the plug-in demanded by browser.

To address the needs of modern Web applications deployment, the latter version of HTML and XHTML is outdated and requires amendment. The new standard, HTML5 when finalized, is expected to include tags and APIs for multimedia and interactivity. In its current stage, HTML version 5 is only supported in a few browsers, however it is expected and browser developers are providing rapid support for the HTML5 capabilities which is expected to free the web browser from propriety plug-in like flash, QuickTime and Silverlight. While in the development phase, HTML5 is not yet supported by all the major browsers. Although it takes time, but the web surfing is expected to traverse the path led by HTML5 and mitigation towards HTML5 compatible browsers will be felt in coming days [30][34].

With the development of mobile devices, web based mobile applications are growing widely in the world. Having wide options regarding Operating System (OS) in mobile phone market, platform and OS dependent native applications are being hard to design and deploy for multiplatform mobile devices. So, Web based applications are good choice for mobile application developers because of its simplicity and availability in a wide range of platform and OSes.

1.1 Aims and objectives

The main aim of this Masters' thesis is to analyze performance of Smartphone with regard to HTML 5 and flash based web applications in two network scenario- 3G and Wi-Fi.

- To investigate user experience in terms of page loading time and DNS lookup time while accessing HTML5 based web application.
- To investigate the behavior of networks like 3G and Wi-Fi while accessing HTML5 based web application.
- To investigate readiness of HTML5 by comparing network performance of Smartphone with regard to different web applications- HTML5 and flash.
- To investigate the compatibility issues, browser support issues associated between HTML5 application and Smartphone pair.
- To present comparative analysis of Smartphone based on HTML5 application.

1.2 Scope of thesis

This thesis report mostly deals with HTML5 related issues, browser issues, comparison and readiness of html5 with regard to other application like flash and network performance of two different version and OS of smart phones while accessing html5 canvas and flash canvas application over 3G and Wi-Fi networks. This thesis also presents some different viewpoint of users when exposed to different applications, which either qualifies an act of conformity or indirectly provides some valuable feedback to working group for further rectification or enhancement. The experiment conducted on different Smartphone is limited to canvas application mostly because of it's highly expected usability in games and graph plotting applications including(using) richer graphical user interface (GUI) design and also due to insufficient browser support for other HTML5 elements and tags. The in-depth browser requirement, support and other issues are not considered in this thesis. Furthermore, the experiment conducted in 3G network scenario is limited to only one operator due to probable proxy cache issue seen in other carrier. The in-depth analysis of issues seen in operators is not dealt in this thesis work.

1.3 Problem Statement

HTML has already been a language of web [1]; it has been preliminary used to represent scientific documents in the internet. To cope with the increasing demand of potential users for richer content from web, the previous version of HTML, HTML4, is outdated and not sufficient enough to address the needs. To address this lagging hole, different third party plug-in were installed in the browser itself. The survey conducted by adobe also depicted that about 99% of world computer has flash player installed in it [36]. However, not all devices and operating systems support flash plugins. Also, plugins add extra overhead in universal client side. So to rectify this issue, there should be some paradigm shift. HTML5 attempts to make universal language standard for web which provides richer additional tags and elements thereby practically avoiding the need of third party proprietary plug-in. So the readiness of the HTML5 application should be assessed in its infancy stage in other to visualize its capability in the current point of time as well as to figure out its potentiality in near future.

1.4 Research questions

The following research questions are dealt in our thesis work.

- How much are HTML 5 applications dependent of connected networks (3G and WLAN) in smart phones?
- Does HTML5 Web applications provide better user experience in terms of page loading time and DNS lookup time?
- To what extend will the performance of Smartphone be affected, in terms of memory and CPU consumption, when accessing HTML 5 based web application?

1.5 Research Methodology

In this section we provided the methodology that we have applied to investigate networks behavior against HTML5 and flash based applications. Different parameters and methodology were applied to carry out the tests which is explained in this section.

1.5.1 Selection of Papers

Literature review and selection of papers are important factors in contributing to any research work. For the purpose of our thesis we followed deductive approach as described by Saunders *et.al.* [31]. Although Saunders *et.al.* specifically deals with modern research methodology and techniques for business students, other fields of studies and even scientific research can also be benefited as it clearly explains modern research methodology and techniques on their behalf as well. HTML5 is relatively a new technology which is still under development phase; however it is related to the previous versions that points toward better promises and additional features. We deduced the hypothesis on the basis of the previous work, and accordingly research questions were formulated. For the selection of papers on the previous work, we applied critical analysis approach by generating keywords and then finding relevant research papers on the subject. Smartphone web applications and HTML5 being relatively new, not much research work is carried out. We generalized the trends in Smartphone application development and capabilities of the HTML5 technologies to finally deduce our research questions and thus try to answer the questions by formulating relevant hypothesis. For the justification of research questions, we conducted experiments, collected data set and then analyzed them thereby linking it to our laid hypothesis to check and verify whether the laid hypothesis actually qualifies high degree of agreement or not.

1.5.2 Network Behavior study

To study the network study of the Smartphone, we installed different packet sniffing tools in the phone. Mobile Scanner is a professional mobile packet sniffer tool that we installed in windows phone. Shark root, which is built atop tcpdump [18], was installed in android phone to capture the network data packets from the android phone. Each sniffing tools were used to measure the network behavior of the Smartphone while accessing HTML5 and flash based web content. The experiments were repeated 30 times each for HTML5 and flash web content as well as for 3G and Wi-Fi connections. The results of sniffing tools were saved in files and later analyzed using wire shark. We focused on DNS lookup time to study the network behavior of Smartphone in 3G and Wi-Fi connection.

1.5.3 Page Loading Time

Different mobile phones were used during the process of the experiments. Though both mobile phones were the product of same vendor, HTC, they differed in model and operating system. For the measurement of page loading time, we approached two methodologies considering the fact that the page loading time on the browser will also be dependent on hardware of mobile phone used. First we measured page loading time by calculating the time taken by the first HTTP GET request that is initiated by device browser to the last reply received from the server while accessing web content.

In the next approach, we developed two animations for HTML5 canvas and adobe flash that has a provision of displaying time elapse after the animation is completed. This method is used to investigate the application rendering time taken by the individual device browser which in a way helps in predicting the hardware limitation of Smartphone.

1.5.4 CPU and Memory Utilization

Task manager [28] was used to measure the CPU and memory utilization in the windows mobile phone whereas Process Monitor [29] was used in android phone.

While conducting the experiment, both the tools were used in the respected devices on that very moment when the web application contents started to load.

1.6 Motivation

The main motivation of this study is to visualize the user experience with respect to performance of HTML5 web applications in smart phones in two networks (3G and Wi-Fi). With the development and migration to the client based applications, we believe that HTML5 would play a major role for client-side web application development with JavaScript backend. The research is intended to provide some feedback to the developers and mobile vendors as well as cellular network providers with the performance evaluation of content aware applications in Smartphone.

Our thesis is the outcome from the inspiration of the work done by Junxian Huang *et.al.* [9] who performed the network behavior study of the Smartphone in different carrier network with respect to content service provided. While they performed the study of 3G cellular network behavior, our study is to estimate the performance of HTML5 web content and perform and evaluate the difference while browsing the HTML5 versus the flash plug-in based web contents. Also, [11] [12] and [13] have performed the performance on desktop while using the 3G data cards and phones for the internet connectivity. But their work is limited to the desktop computers and does not consider the performance of Smartphone itself. Our study also focuses on the performance of the Smartphone itself in the dimensions of memory consumption and CPU usage statistics with respect to HTML5 and flash based web contents which in a way give device reliability and user's perception.

1.7 Main contribution

This thesis gives in-depth understanding of HTML5 deployment scenario, browser deficiency or dependency and users experience while accessing the application right through the Smartphone. Furthermore, two network scenarios- 3G and Wi-Fi are dealt separately which give clear viewpoint of network behavior and probable parameter shift which can affect the user experience. In addition to that, the readiness of HTML5 can also be visualized, which can ultimately provide feedback to respective working group.

1.8 Thesis Outline

This section describes some basic outline of the thesis work.

- Chapter 2 provides a detailed description of web service scenario and evolution of web applications in mobile phones.
- Chapter 3 provides a detailed description of experiment setup we used.
- Chapter 4 deals with thesis result and analysis.
- Chapter 5 provides conclusion of the thesis work and future extension of our work.

2 LITERATURE REVIEW AND STATE OF THE ART

2.1 Mobile Phones

Mobile phones were developed to fulfill the need of wireless telephone in a larger degree of mobility for the users [19] and have evolved to an advanced minicomputers that what is now known as Smartphone. With the advancement of the telecommunication service, major computer and operating system development companies like Apple, Microsoft and other web service companies like Google have invested in the development of the mobile devices and mobile operating system. Today, mobile phones are not only capable of making voice calls, but one can run desktop like applications and advanced services from their mobile phones.

2.2 HTML5

HTML has always been the World Wide Web's markup language [1]. HTML found its primary use in the field of semantic description of scientific documents; however its design and adaptation over the years have made it capable of describing other types of documents [1].

HTML was first published as an internet draft in 1993. Throughout the decade, an enormous amount of activities were carried out around HTML which in turn led to the evolvement of different HTML release versions 2.0, 3.2 and 4.0. After the release of HTML version 4.0, the World Wide Web Consortium (W3C), the governing body responsible for the control of HTML specifications stopped further development on HTML specification [14].

However, feeling the need for updates and upgrades in the HTML specifications, a new working group called WHATWG was formed jointly by Apple, Mozilla and Opera, that started working on the elaboration of the HTML specifications. By 2006, W3C joined with WHATWG to work on further development of the HTML specification named HTML5 [8].

HTML5 is not only the name of new specification for the newer version of HTML. It is a merger of HTML, CSS, ECMA Script (JavaScript) and advanced DOM layer APIs, which provides higher and richer possibilities for web applications. With this major release, now the application can be delivered using plain HTML tags, CSS styles and JavaScript backend programming logic which were not possible previously without the use of third party plug-in.

HTML5, however already making a great influence to the web application development process, is still a work in progress with major development promises. With the new feature promises like <device> tags and file and local storage APIs, HTML5 is expected to grow beyond the limitation of services that were previously only capable to be delivered using native applications.

2.3 Survey of Related Works

The kind of data traffic generated by mobile users while accessing the web content is presented in [3]. User perception regarding the performance of network applications on Smartphone in several dimensions such as carrier network, device capabilities and server configuration are presented in [9] [2] [3] which shows the network behavior of Smartphone with respect to service provider and service content. The actual use of mobile device while browsing web content and the service provided is presented in [15]. Although the model proposed in[15] , used for visualizing the actual essence of mobile web browsing is limited to Korean market, it presents that the content quality is positively related to user's attitude towards mobile web browsing service and has potential of delivering practical implication to IT vendors if tested in global market.

Mobile applications comparison on different runtime environments including flash [6] are presented in [10]. The possibilities of providing the web services using HTML5 for rich web contents like learning management system are also possible, which is presented in [16]. In [17], deals with compatibility issues associated with user equipment which makes difficult in delivering new services from service providers. However, it shows that HTML5 along with the new Web Sockets API can be used which allows access to IMS-based services. Also, since prior work [22] shows that application performance often significantly differs from the protocol layer performance, we focus on the application layer performance of Smartphone to HTML5 web applications.

A survey organized by adobe Inc. [7] reports that user prefers using web and web based services over native applications while accessing consumer application including secure bank transactions.

Recent study and development in the field of web applications, as presented in [34], Antero Taivalsaari *et. al.* claims that the trend towards web-based software will cause a paradigm shift in the software industry from binary applications to dynamically delivered web applications. Author further claims that in the near future, the use of binary software will be limited in the realm of servers only while end users will be using web-based applications, where we may witness promising trend of HTML5 in turning the web into a real software platform.

Other recent development in the field of this new technology as illustrated in [32], [33] and [35] claims that the addition of new element tag like <canvas>, and new APIs like Websockets in the HTML5, will provide an efficient use of data visualization and graphical environment.

While most of the research on mobile web applications and services are focused on the measurement of network behavior and content accessed by users, the actual performance of Smartphone and user experience are yet to be perceived. Also, the emerging HTML5 based mobile web services are supported by major mobile OS and device manufacturers, the actual performance of HTML5 based applications with respect to the traditional flash and other plug-in based services on Smartphone is still unknown. As the web is developing into a real software development platform with the promising new technologies, it is thus essential to study the client side performance of the new emerging HTML5 based applications in Smartphone and to compare the performance of same with respect to traditional plug-in and add-on based applications which could provide the application developers as well as the mobile OS and device manufactures to support and focus on the performance of such web application in Smartphone.

3 EXPERIMENT SETUP

3.1 Experimental Environment

An experimental environment was set up by creating a simple auto line drawing animation in HTML5 canvas and adobe flash. As described in §2.3, the canvas element of HTML5 is widely been accepted as a web-based application platform to develop different graphical and visualization applications. This is currently been deployed using the additional client side add-on applications including flash and active-x technology which is costly in terms of service side deployment and propriety costs. Also, since we propose the investigation of open technology, HTML5, over propriety based browser add-ons, the insight and in depth study of HTML5 canvas element in a way helps in predicting the performance of HTML5 applications. The hardware limitation will be more prominent only in inter-device comparison, and will be devoid of when only one device is taken into consideration. Since we are dealing with two applications running on same device, the performance will be solely depended on browser efficiency rather than hardware efficiency.

Two animations were developed in HTML5 and flash canvas using respective programming languages JavaScript and Action Script with similar programming logic and time interval setups. The applications were placed on university computer running Apache 2.2.9 web server on a Debian 5.0.8 (with Linux Kernel 2.6.27-2-686) server. For the consistency of browsing in different phones, opera mobile browsers were installed in both phones. Packet sniffers were installed in mobile phones to monitor any packet level differences while accessing flash and HTML5 web contents. Fig. 1 shows a screenshot of line drawing application created in HTML5 canvas which has a provision of displaying load time in the browser after the completion of line drawing process. An exact replica of application is created in flash for measuring the performance of flash application.

Two Smartphone: one HTC HD2 running Windows Mobile 6.5 Professional and another HTC Desire running android 2.2 were used during the period of experiments. Both devices are equipped with IEEE 802.11 b/g Wi-Fi. Further details of mobile devices, browser and applications running on mobile device background during the period of this experiment are provided in appendix.

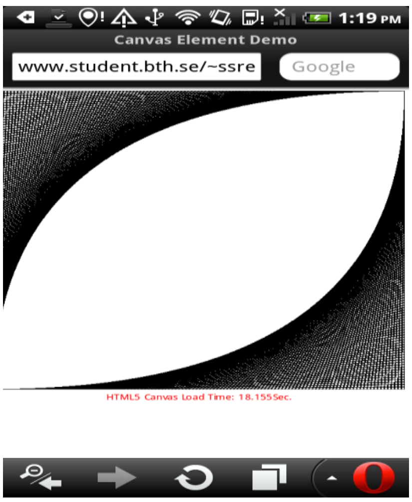

Fig. 1 HTML5 canvas line drawing application screenshot from android phone showing load time

3.2 Network Performance Experimental Setup

A basic network layout of the experimental setup is presented in Fig. 1, where the access point can be viewed as Wi-Fi access point or 3G node B terminal. Two smart phones: android and windows were used during the course of experiment. Fig. 2 shows the page request flow diagram where page requests made by mobile browser from the web server are captured by sniffer tools installed in mobile phones.

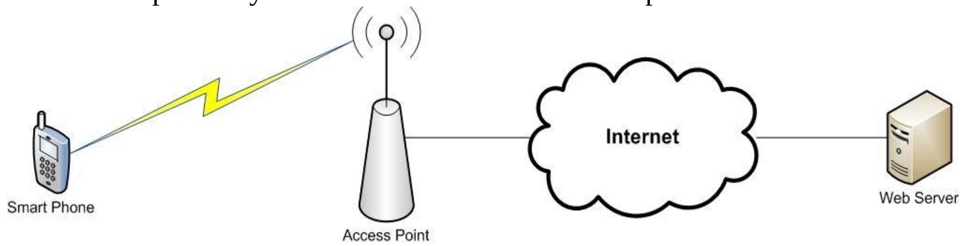

Fig. 2 Network Layout of Experiment Setup

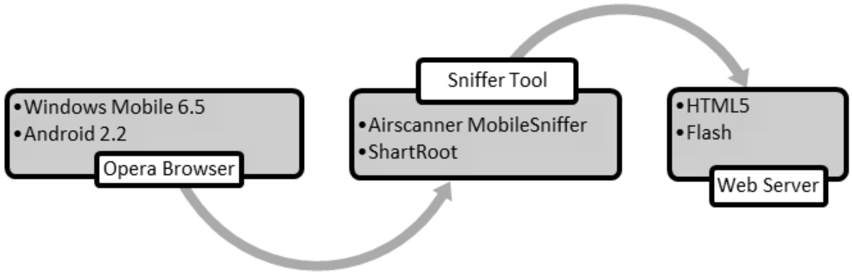

Fig. 3 Page Request Flow Diagrams

To study the network performance, we installed Mobile Sniffer on windows mobile and shark root on android mobile. Mobile Sniffer is a professional sniffer tool developed by Air scanner that was used to monitor traffic from HTC HD2. Shark root is similar network sniffer tool for android phones and devices. Shark root is built atop tcpdump which is a widely used and deployed UNIX and Linux sniffing tool. The application is then retrieved from the server and packets were captured via respective sniffing tools.The validity of tools were performed by connecting mobile device as wireless router and tethered device to our laptops thereby monitoring the captured packet through laptop. Both the captured packet were then analyzed which showed high degree of agreement between the collected data sets.

A detailed study of DNS latency and layered protocol packet would provide an insight of whether HTML5 based web applications performance is affected by the browser engines and/ or the operating system. We chose the DNS latency and application layer packet analysis method to measure time taken by devices to actually receive the packets and then finally to measure page loading time as described in §3.3 by the browser. This way we could significantly identify if the performance of HTML5 web applications could be affected by the performance of JavaScript engines in different operating system and browsers.

3.3 Page loading Time Experimental Setup

Page loading times were calculated using two approaches: one approach, that is irrespective of browser, is used to measure the loading time by analyzing the first http GET request packet form mobile device to the last FIN ACK respective response from the server while browsing the web content. For other part of the loading time of HTML5 and flash based contents, two separate web applications were created using HTML5 canvas and Flash action script. Both animations were prepared using same logic of time interval and equal numbers of lines drawn. The application would draw lines consecutively at an interval of 50 milliseconds. After all lines are drawn, a time difference between beginnings of first line drawn to the end of last line drawn is evaluated. The HTML5 and Flash execution time was calculated and displayed in the browser which was then noted manually after the complete load of the animation. Both contents were put on the same server without any change. Since flash applications are run in a different process environment, that is flash player, inside a web browser as opposed to HTML5 canvas which is rendered by the browser engine itself, this method will help us identify the performance of HTML5 in mobile device as oppose to the hardware limitations.

We chose page loading time to check the performance of HTML5 web applications in Smartphone because this is the major factor that affect the user experience while browsing in the internet. A significantly high page loading time means a bad user experience which is a great threat to any web application developer and we expect our results would provide an insight to the developers whether or not to choose HTML5 for web application development in the given context of the experiment carried out.

3.4 Smartphone Performance Experiment Setup

The performance of Smartphone was investigated by comparing CPU and memory consumption in individual Smartphone while accessing HTML5 and flash contents from the server. In order to measure CPU utilization and memory consumption in Smartphone, task manager and process monitor were used for windows and android phone respectively.

3.5 On the Validity of Tools Used

3.5.1 MobileSniffer

Mobile Sniffer is a professional packet sniffer tool developed by AirScanner which is a company that specializes on windows mobile security products. The tool is built atop winpcap [24] which is a collection of library files that is widely developed and used in most research tools for capturing windows platform packet traces including wireshark [25]. On the validity of tool used, we conducted two independent experiments. In the first experiment, we installed mobile sniffer in windows mobile device and captured packet traces thereby accessing the contents from the server. Alternately in other experiment, we configured windows mobile device as a router and was then connected to laptop through Wi-Fi. We installed wireshark in laptop to capture the packet while accessing the content from the server. Two sets of packet traces were compared, which confirmed high degree of agreement. This in a way states an act of conformity, a well fitted appropriate choice, for implementing in our experiment scenario.

3.5.2 Sharkroot

Sharkroot is a freely available tool on android market that is developed atop tcpdump [26] and libpcap. Tcpdump is a widely used tool on Unix and Linux system and libpcap [27] is a widely used set of library files for packet capturing on Unix and Linux machines. Since android is built on a Linux kernel, the tool is ideal for packet capture and we confirmed the validity of tool thereby repeating the same validation technique as described in §3.5.1.

3.5.3 Task Manager

Task manager [28] is a system resource monitoring tool widely used in windows desktop system. For windows mobile, HTC HD2, we have a similar tool named Task Manager that displays individual process and their CPU and memory utilization. Since the tool is shipped by the major mobile device vendor and accepted by the OS and is also widely been deployed for general monitoring of process resource utilization, we propose no doubt on the validity of tool.

3.5.4 Process Monitor

Process monitor [29] is a freely available tool for monitoring processes resource utilization on android devices. The tool provides a list of running processes and the resource utilization by individual processes as well. The validity of the tool was checked by using the top command which is a widely used process monitoring tool on Linux systems. Process monitor was used for its simplicity and user friendly graphical user interface.

4 RESULT AND ANALYSIS

In chapter 3, the experimental setup for the thesis work was explained. In this chapter, results obtained from experiments will be discussed and analyzed for different network scenario with regards to constraints: DNS lookup time, page loading time, and memory and CPU consumption. The experiments were carried out on the smart phones using packet capture tools while clearing cache each time to avoid any effect on the browser page loading time. The experiment was repeated from 30 to 35 data sets independently for all the constraints used. Mean value is taken as a reference for comparison in our case and variance is considered for determining the spread of data set. All experiments were taken during the working hour, between 9 AM to 5 PM, when the network load is expected to be high.

Section 4.1 covers the experimental results and analysis of DNS lookup time obtained in Smartphone while retrieving html5 and flash application in two different network scenario- 3G and Wi-Fi.

Section 4.2 presents results and analysis for network performance constraints- page loading time in two network scenario- 3G and Wi-Fi.

Section 4.3 deals with results and analysis of memory resource consumed by different applications on Smartphone.

Section 4.4 deals with results and analysis of applications running on different Smartphone with regard to constraint- CPU consumption.

Finally, detailed results and analysis of device dependent constraint- page rendering time while accessing different applications- Flash and HTML5 is presented in **Section 4.5**.

4.1 Network performance constraint-DNS lookup time

DNS lookup time, in internet browsing, refers to the total time taken by servers to resolve the IP address of the requested URL. It is generally expressed in millisecond (ms). Lower DNS lookup time means faster discovery of the requested server thus providing quicker response time from the requested server.

In our experiment we measured DNS latency to identify the server lookup time in our selected two network scenario: Wi-Fi and 3G.

4.1.1 Wi-Fi Network scenario

The HTML5 application requested by Smartphone passes via Wi-Fi network having an average of 24.39Mbps download and 21.16Mbps upload speed. The experiment was conducted for 30 numbers of times. The obtained results are presented in Fig. 4. The average DNS lookup time for windows mobile was found to comparatively less as opposed to that of android mobile. Also the scattering of data was observed to be significantly less. The mean and variance values are presented in Table 1. We also witnessed some variations in lookup time which is because of network load and congestion in the network itself. In case of android phone we observed one peculiar instant where the DNS lookup time increased exponentially to a peak value of about 1 second. Unlike, the usual case where every DNS request is followed by response, in that particular instant the DNS request were followed by two consecutive TCP resets with another DNS query to Google server before DNS response. This anomaly also affected the mean value of the lookup time that we presented above. DNS and other TCP request to Google is common in android phones for synchronization however TCP resets are common but not clearly explained in Smartphone. During our experiments, we observed several TCP resets during the page requests, which requires further research for the cause and is out of the scope of this thesis.

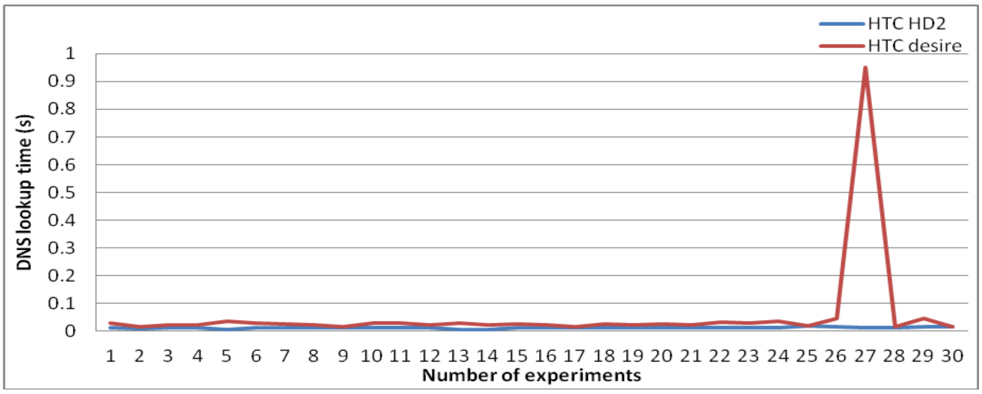

Fig. 4 DNS lookup time for Smartphone in Wi-Fi network. X-axis represents no of experiments and Y-axis represents DNS lookup time (sec)

Table 1: DNS lookup time for Wi-Fi network

Device	Average DNS lookup time(ms)	Variance
Android	56.35	0.027
Windows	12.05	$9.408x10^{-6}$

4.1.2 3G Network Scenario

We took the measurements using the 3G connection of Tele2 and Telia mobile network service provider of Sweden. The average network speed observed was found to be 4.87 Mbps download and 1.26 Mbps upload average speed. The results are presented in Fig. 5, where we observe that 3G network has significantly varying DNS lookup latency. The average DNS lookup time for android mobile was found to be significantly low as compare to windows mobile. The mean value and variance are presented in Table 2. We witnessed dynamic fluctuation in the DNS latency. This is quite obvious as the application requested surpasses through 3G network, where the application experience additional switching operations in order to reach DNS servers. So the time is expected to be of increasing nature. Also, during the period of experiments we observed some unexpected behavior from one of the service provider, which looked like a use of caching proxy server that downloaded all the static web contents from the web server and later served the client requests from the proxy cache server. This behavior however is not further discussed as it is out of the scope of this paper and requires additional study tools and techniques.

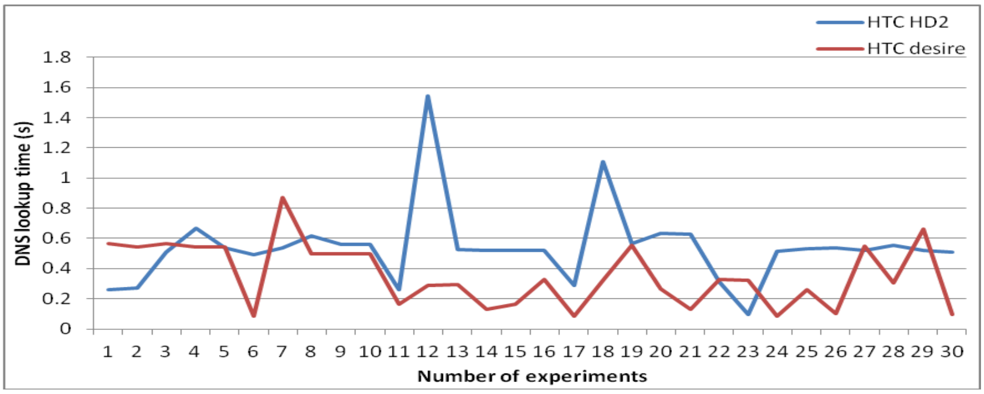

Fig. 5 DNS lookup time for Smartphone in 3G network. X-axis represents no of experiments and Y-axis represents DNS lookup time (sec)

Table 2: DNS lookup time for 3G network

Device	Average DNS lookup time (ms)	Variance
Android	33.35	0.044
Windows	540.83	0.066

4.1.3 Summary

Regarding the DNS latency, we observe that the Wi-Fi network has a low latency values however in case of 3G network the DNS latency is high, unpredictable and fluctuating. Also, since the experiments were carried out in the closed environment of university network the DNS lookup response in case of Wi-Fi was from university name server. This also contributes to the low DNS latency in case of Wi-Fi network. While in case of 3G, Swipnet backbone name resolver of Tele2 were responsible for DNS responses. The detail of name resolver servers are provided in appendix.

Also, from the experiments we observed that the DNS lookup time in Wi-Fi network is almost linear which makes it predictable however 3G networks has a fluctuating and high DNS latency which makes it more unpredictable and slow. This is due to the fact that unlike Wi-Fi network, there will be additional switching process going on in 3G network and the requested DNS server is also positioned in different network which obviously increases the time thereby collectively making the DNS latency go high.

4.2 Network performance constraint- Page loading time

Page loading time is calculated by deducting the time taken by first HTTP GET request that is initiated through the device with the last respective ACK, FIN response generated from the server. It is irrespective of content rendered in the phone browser. This provides an insight behavior of network against two applications- HTML5 and Flash.

4.2.1 Wi-Fi Network Scenario

The page loading time observed in Wi-Fi network are presented in Fig. 6. The load times were almost similar in windows and android mobile devices regarding HTML5 and flash contents. For both phones, the applications took almost the same load time but the spread of data set in case of flash application was significantly high as oppose to HTML5 application. The mean value and variance are presented in Table 3.

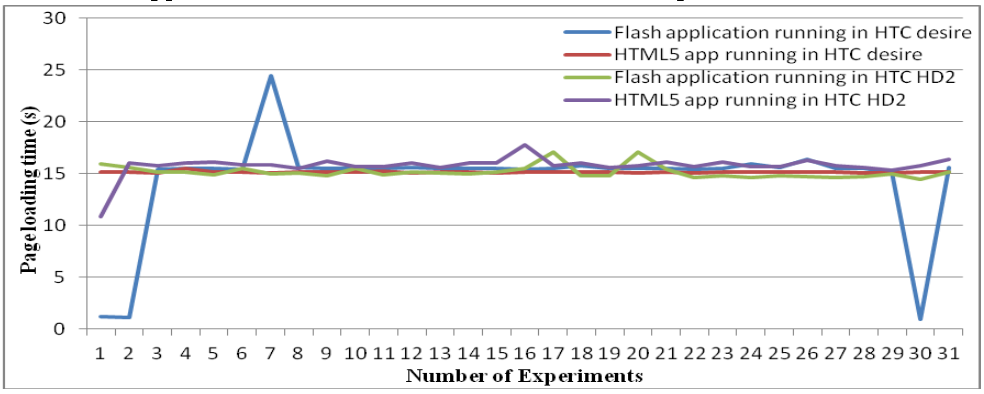

Fig. 6 Page loading time for applications running on smart phones in Wi-Fi network. X-axis represents number of experiments and Y-axis represents page loading time (sec)

Table 3: Page loading time in Wi-Fi network

Device	Content	Average loads time (s)	Variance
Android	HTML5	15.14	0.012
	Flash	14.55	19.88
Windows	HTML5	15.79	0.91
	Flash	15.09	0.37

From the captured packets, we observed that mobile devices in most cases make requests to the time server (daytime protocol), SSL certificate and other TCP requests to Google or Microsoft servers during the time of page browsing.

In Fig. 6 we observed some anomalies regarding flash application load time in android phones. The in depth analysis of such anomalies is done here under :

- The data results of 1, 2 and 30 as shown in Fig. 6, showed no TCP or SSL requests to Google servers, but only one TCP request to daytime server without any interruption on the load time thus decreasing the load time.
- The dataset obtained at 7, had TCP retransmissions with requests to daytime and Google servers during the process of page loading thus increasing the load time.

4.2.2 3G Network scenario

The page load time of HTML5 and flash canvas contents on both mobile devices is presented in Fig. 7. For android phone, there is no significant difference between the page loading time taken by both the applications, but we witnessed high spread of data set in case of HTML5 application as oppose to flash application. While for windows phone, HTML5 application loaded significantly faster than flash application. Also we observed uniform distribution of data sets in case of HTML5 application as opposed to flash application. The mean value and variance are presented in Table 4.

The load time for both flash and HTML5 application in android phone was observed to be similar to some extend except for one instant where there is exponential increase in loading time for HTML5 application, which can be seen in Fig. 7. Also from the captured files, we found that file 22 and 23 showed larger load time in android phone while accessing HTML5 application. This is because of the TCP retransmission and reset signals found during the browsing of the web content.

In case of windows mobile, we observed that the load time for HTML5 application was considerably very low as compared to that of flash application. From the analysis of the capture files, we came to a conclusive statement that the irregularities as seen in the Fig. 7 for flash application is due to regular occurrence of reset signal during the period of flash web content download.

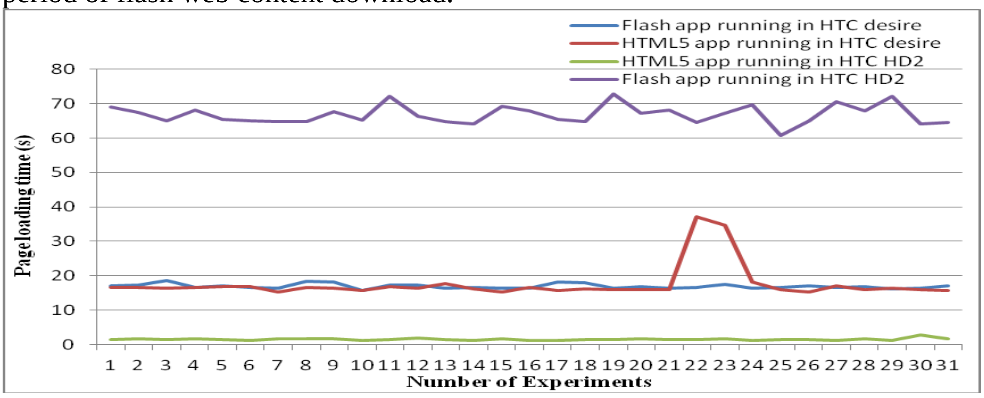

Fig. 7 Page loading time for applications running on smart phones in 3G network. X-axis represents number of experiments and Y-axis represents page loading time (sec)

Table 4: Page Loading Time in 3G network

Device	Content	Average loads time(s)	Variance
Android	HTML5	17.43	21.68
	Flash	16.92	0.49
Windows	HTML5	1.53	0.08
	Flash	66.72	6.96

4.2.3 Summary

From the experiment, we observed that in both mobile device and network connection, the HTML5 web application is loaded significantly faster than the flash content. Also, from the captured files and the results illustrated above, we can make a conclusive statement that if the mobile device synchronization to the respective device OS vendor servers and daytime servers are discarded, then the load time of the HTML5 based web applications would be even much less as compared to the load time of flash based web application.

4.3 Device performance constraint -Memory Consumption

Memory consumption in our case is the total amount of memory resource consumed by applications- HTML5 and flash, which is devoid of memory consumed by mobile browser. It is expressed in megabyte (Mb).

4.3.1 Wi-Fi Network Scenario

From the graph illustrated in Fig.8, we observe that the memory resource consumed by HTML5 application is less compared to that of flash application irrespective of smart phones. From the device point of view, memory resource consumed by both applications for windows phone (HTC HD2) is much lesser than that of android phone (HTC desire). The mean value and variance are presented in Table 5.

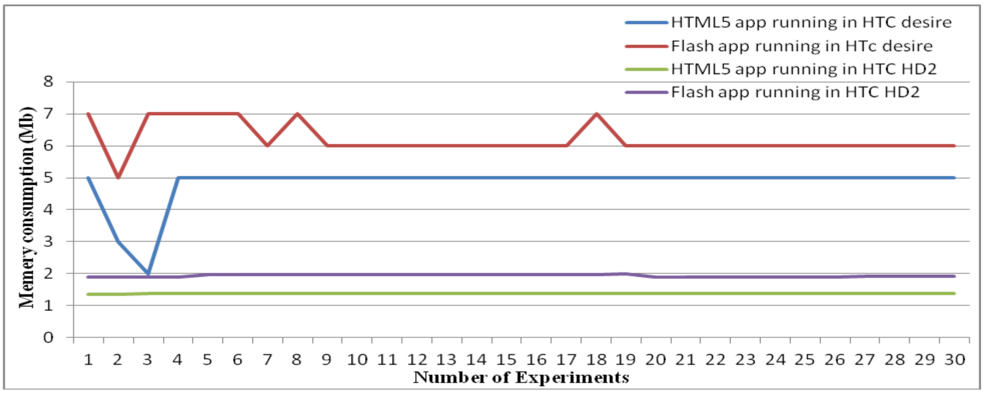

Fig. 8 Memory consumed by applications running on smart phones in Wi-Fi network. X-axis represents number of experiments and Y-axis represents memory consumption (Mb)

Table 5: Memory Consumed by applications in Wi-Fi network

Device	Content	Average Memory Consumption (Mb)	Variance
Android	HTML5	4.83	0.42
	Flash	6.2	0.23
Windows	HTML5	1.37	1.28×10^{-05}
	Flash	1.93	0.001

4.3.2 3G Network Scenario

From the graph illustrated in Fig.9 shows that HTML5 consumes less memory compared to flash application irrespective of smart phones used. Mean and variance were calculated and are presented in Table 6. From the table, we can observe that there is no significant spread of data sets for both the applications. The HTML5 application consumed relatively less memory.

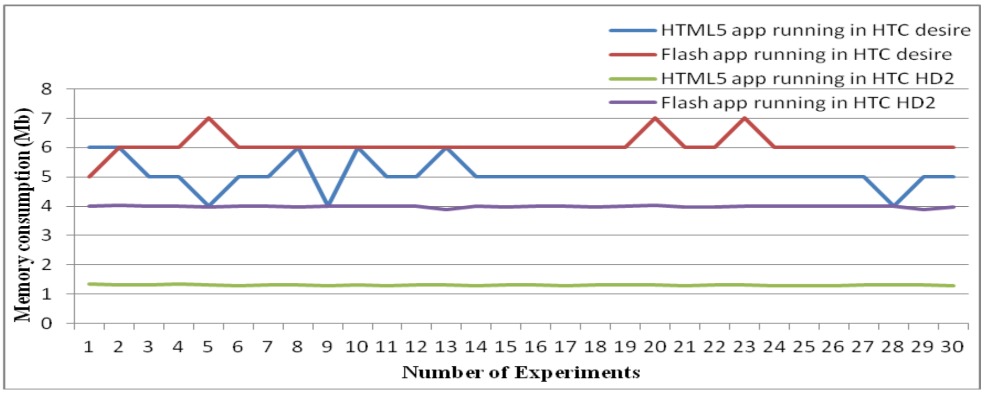

Fig. 9 Memory consumed by applications running on smart phones in 3G network. X-axis represents number of experiments and Y-axis represents memory consumption (Mb)

Table 6: Memory Consumed by applications in 3G network

Device	Content	Average Memory Consumption (Mb)	Variance
Android	HTML5	5.07	0.27
	Flash	6.07	0.13
Windows	HTML5	1.30	0.0002
	Flash	3.98	0.0012

4.3.3 Summary

From the above graphs Windows phone outperforms android phone irrespective of any network used. For both mobile phones the memory resource consumed by flash application is comparatively high. This is because of the fact that, unlike HTML5, while accessing flash application through the browser additional process of flash player will also be invoked. Despite of less memory consumption, particularly in case of flash application, the performance of windows phone was heavily degraded thereby limiting access to other applications.

4.4 Device performance constraint -CPU Consumption

CPU cycle percentage consumed by Smartphone is measured at the starting point of application load. It is expressed in percentage (%). It comprises collective effect of browser and application.

4.4.1 Wi-Fi Network Scenario

Fig.10 shows the CPU consumed by Smartphone while running different applications. From the graph, we can observe that android phone (HTC desire) outperforms windows phone by huge percentage irrespective of any application used. There is some fluctuation in the data set which is obvious as we can't measure exact CPU percentage in real time scenario. To visualize the scale of fluctuation, we calculated variance. Both the mean values and variance are presented in Table 7.

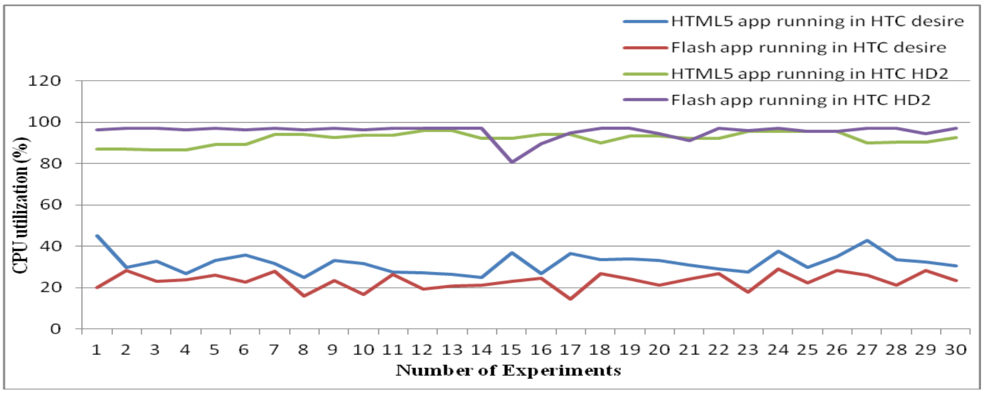

Fig. 10 CPU consumed by applications running on smart phones in Wi-Fi network. X-axis represents number of experiments and Y-axis represents CPU Utilization (%)

Table 7: CPU Consumed by applications in Wi-Fi network

Device	Content	Average CPU Consumption (%)	Variance
Android	HTML5	32.10	22.68
	Flash	23.29	14.81
Windows	HTML5	92.20	8.44
	Flash	95.58	11.33

4.4.2 3G **Network Scenario**

Fig.11 shows the CPU consumed by smart phones while running applications in 3G network. We observe high percentage of CPU usage in case of windows mobile irrespective of application used while for android phone the CPU consumption is minimal. The mean values and variance are presented in Table 8. From the value represented in the table, we can figure out that there is no significant variation in data sets for windows mobile, but in case of android mobile, we witnessed some fluctuation in data sets.

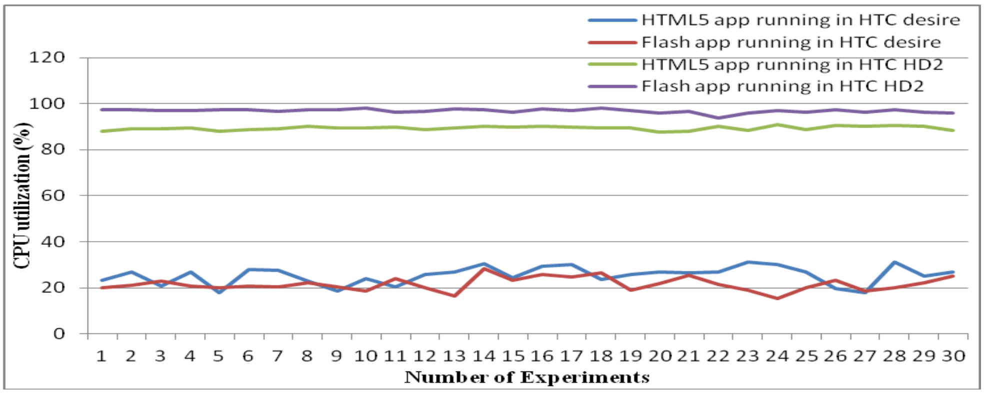

Fig. 11 CPU consumed by applications running on smart phones in 3G network. X-axis represents number of experiments and Y-axis represents the CPU utilization(%).

Table 8: CPU Consumed by applications in 3G network

Device	Content	Average CPU Consumption (%)	Variance
Android	HTML5	25.43	15.16
	Flash	21.62	8.80
Windows	HTML5	89.39	0.78
	Flash	96.82	0.70

4.4.3 Summary

From the performance measurement of Smartphone while accessing the HTML5 and flash based web applications, we observe that windows phones falls behind android by consuming a huge amount of CPU utilization. Also, during the experiment we observed that though windows phones consumed less memory as compared to android phones, the phone was unresponsive to any other user interaction while accessing the flash based contents. This clearly explains the demerit of huge CPU utilization of the flash based web applications that have a direct impact on user experience while accessing the flash based web contents. However the animation in flash based application was smoother and more elegant than the HTML5 canvas animation which was sloppy and dull.

4.5 Device performance constraint –page rendering time

Page rendering time is the time taken by the application to load. The JavaScript code included in the application itself is responsible for displaying execution time, which is different to the page loading time that is calculated from network point of view. It is generally expressed in seconds (s).

4.5.1 Wi-Fi Network Scenario

Fig.12 shows page rendering time for applications running in Smartphone for Wi-Fi scenario. From the figure, we observe that unlike windows phone, android phone (HTC desire) takes less time to load the applications. From application point of view, HTML5 takes much more time to load than that of flash application irrespective of Smartphone. Despite of additional overhead (plug-in) used by browser to load flash content, flash application outperforms HTML5 application with regard to page rendering time in case of android phones. The mean value and variance are presented in Table 9. From the values presented in table, we can visualize that the data sets for both the applications are uniformly distributed.

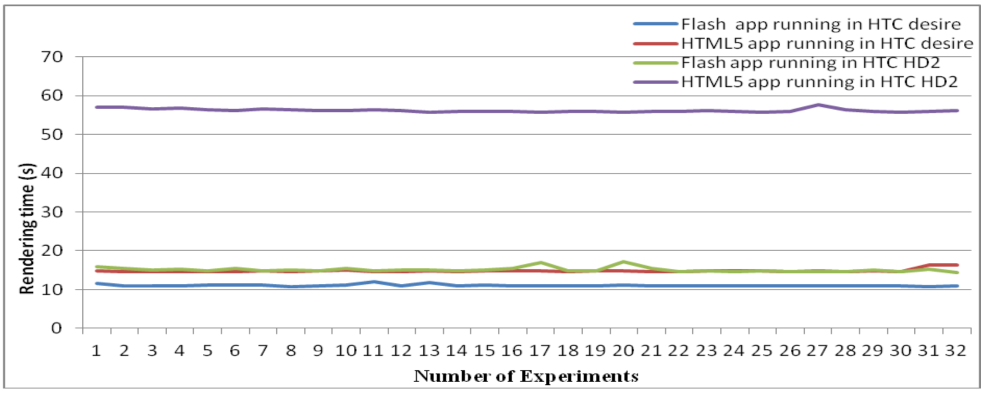

Fig. 12 Page rendering time taken by applications running on smart phones in Wi-Fi network. X-axis represents number of experiments and Y-axis represents page rendering time(seconds).

Table 9: Page rendering time taken by applications in Wi-Fi network

Device	Content	Average Rendering Time (s)	Variance
Android	HTML5	14.82	0.16
	Flash	11.06	0.06
Windows	HTML5	56.18	0.20
	Flash	15.14	0.37

4.5.2 3G Network Scenario

Fig.13 shows page rendering time for applications running in Smartphone for 3G scenario. From the figure, we observe that irrespective of any application used, android phone (HTC desire) takes less load time than that of windows phone (HTC HD2). Furthermore, from application point of view, flash application outperforms HTML5 application irrespective of any Smartphone used. The mean value and variance were calculated and are presented in Table 10. From the values presented in the table 10, we can witness that despite of extra overhead present in flash application, it took significantly less time to render the content. Moreover the variations seen in the data sets are almost uniformly distributed.

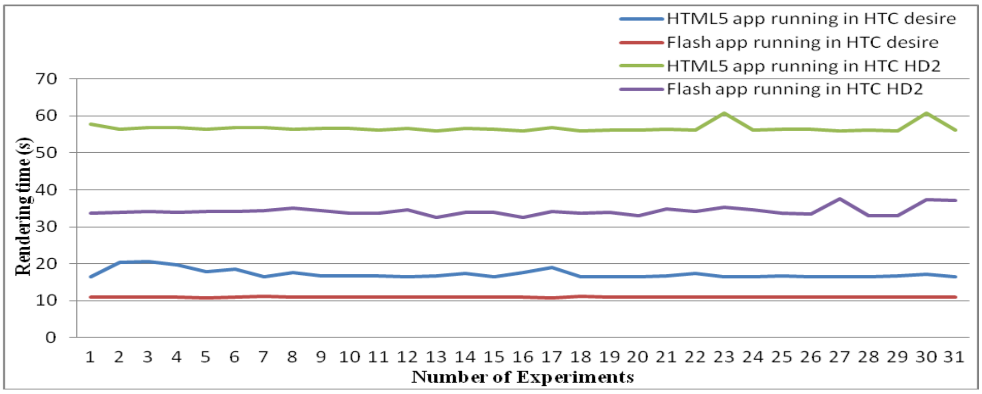

Fig. 13 Page rendering time taken by applications running on smart phones in 3G network. X-axis represents number of experiments and Y-axis represents page rendering time(seconds).

Table 10: Page rendering time taken by applications in 3G network

Device	Content	Average Rendering Time (s)	Variance
Android	HTML5	17.18	1.31
	Flash	10.96	0.006
Windows	HTML5	56.63	1.17
	Flash	34.30	1.55

4.5.3 Summary

Although we have seen from the network performance of Smartphone in § 4.3 that the page load time from network point of view is rather faster, the browser rendering to the applications differ. Since HTML5 is a collection of HTML5 tags, CSS and JavaScript backend programming, the JavaScript execution benchmark plays a vital role in the performance of HTML5 applications in Smartphone. We observe that the performance of HTML5 canvas applications, which is a collection of JavaScript codes to draw images inside of HTML5 canvas tag, is rather slow and sloppy in case of windows mobile as compared to android. This is because of the slow execution of the JavaScript engine designed in the mobile browsers. If the mobile browser JavaScript engines could be designed and optimized for quicker execution of JavaScript codes, the performance of HTML5 based applications can be expected to provide rich user experience with quicker response and loading time in contrast to plug-in and add-ons based applications.

5 CONCLUSION

We conducted the experiments to evaluate the performance of two different Smartphone with respect to HTML5 and flash plug-in based applications. To visualize user experience and readiness of HTML5 web applications we conducted experiments considering different constraints: page loading time, CPU and memory consumption as presented in §4. The observations were made by analyzing the packet traces while loading the web applications and by visualizing the application rendering within the browser. The application rendering time however is a constraint that is dependent on and limited to the mobile device hardware and operating system. Although the HTML5 pages are responded back from the server faster, the limitations in the JavaScript page rendering time in windows mobile phones makes the applications run slower. Despite this fact, we expect a better user experience in terms of mobile phone performance while accessing HTML5 web applications because of the less CPU and memory consumption of the mobile phones while accessing the HTML5 based web contents.

If mobile device operating systems and mobile browsers have a better JavaScript execution time, the performance of HTML5 is expected to take over to legacy plug-in based applications. Also, with new mobile device operating systems like android and their support for most of the HTML5 capabilities in the native browsers as well the performance of HTML5 based web applications are better and smoother.

From the experiments conducted and observations taken, we came to an conclusion that although the performance of HTML5 based web applications were providing better user experience in terms of network loading time, CPU and Memory consumption thus increasing the possibility of switching between the applications within the mobile devices, the flash based applications were smoother and faster in terms of rendering in the browsers. In terms of single application access by the user, the flash based applications, as we observed were effective in terms of rendering and display but would limit the multi-tasking feature of the mobile device with low effective utilization of the application switching capability.

HTML5 is in the development process and developers are trying to make it better with more features. However, adding more new tags and capabilities to the DOM layer and making the universal standard for all browsers will not still make HTML5 better than other legacy plug-in based applications until the developers can provide a better JavaScript engine for mobile devices for less page rendering time in the client device. Also the device hardware limits the performance of HTML5 web applications.

In the beginning of the thesis we sorted out three research questions whose solutions are explained here under.

5.1 Research Questions and Answers

R1: How much are HTML 5 applications dependent of connected networks (3G and WLAN) in smart phones?

From Fig. 6 and 7 we observed that HTML5 applications are less resistant to connected networks. There was no any abrupt change in the performance of HTML5 application irrespective of any smart phone being used. While we witnessed performance degradation of flash application in 3G network, where we found regular occurrence of RESET signal. As there is no any significant alteration and degradation in HTML5, we can make a statement that switching to HTML5 application may eradicate the regular reset problem because it is less susceptible to networks being used.

R2: Does HTML5 Web applications provide better user experience in terms of page loading time and DNS lookup time?

From Fig. 6 and 7, we observed that HTML5 web application provides better user experience as compared to flash application when page loading time is taken as decision constraint irrespective of any device used. Although HTML5 confirms better user experience when viewed from network point, the other behavior as seen in Fig. 12 and 13 should also be accounted, as it contradicts the conclusive visualization illustrated in Fig. 6 and 7. In Fig. 12 and 13, we observe that despite of extra overhead and additional process demanded by flash application, it is executed more faster compared to HTML5 application. If the browser developers and OS developer could provide better JavaScript rendering engines within the browser, HTML5 application can be expected to run even faster than other proprietary applications.

R3: To what extend will the performance of Smartphone be affected, in terms of memory and CPU consumption, when accessing HTML 5 based web application?

In relation to the CPU and memory consumption, we observed, as shown in Fig. 8, Fig. 9, Fig. 10 and Fig. 11, that HTML5 web applications definitely consumed less CPU and memory thus providing better efficiency within the mobile phones. The mobile phone performance would thus be highly affected in terms of CPU and memory consumption thus providing high efficiency in terms of battery life and phone performance.

6 FUTURE WORK

During the course of the experiments, we observed that one of the cellular network service provider responded to the static web page requests made from the Smartphone through its proxy server. This kind of behavior is new and unexpected from the network services provider. We would like to take this matter as a further research and study plan. Also, the limitations of HTML5 capabilities due to Operating System versions and browser incompatibility, more features of HTML5 were limited to one. We would like to further investigate the performance of Smartphone to other capabilities of HTML5 web applications in future.

HTML5 being new and on the development phase, more capabilities are being added to it. We would like to further investigate on the new possibilities of the HTML5 applications on mobile devices and the performance of client side programming capabilities.

7 REFERENCES

1. www.w3.org, HTML 5, *[Online]*, Available http://www.w3.org/TR/html5/
2. S. J. Vaughan-Nicholas, Will HTML 5 Standardize the Web, *in Computer*, April 2010, pp. 13.
3. F. Hossein, L. Dimitros, M. Ratul, S. Kandula, E. Deboran, A First Look on Traffic on Smartphone, *[Online]* Available http://www.cs.ucla.edu/~falaki/pub/imc153s-falaki.pdf, Date Visited: 2011-01-07
4. L. Plissonneau, G. Vu-Brugier, Mobile Data Traffic Analysis: How do you Prefer Watching Videos, *in 2010 22nd Intl. Teletraffic Congress (ITC)*, Sept. 2010, pp. 1.
5. K. Wac, M. Fiedler, R. Bults, H. Hermens, Estimations of Additional Delays for Mobile Application Data from Comparative Output-Input Throughput Analysis, *in 2010 IEEE Network Operations and Management Symposium (NOMS)*, April 2010, pp. 171.
6. Flash Platform Benefits, *[Online]*, Available: http://www.adobe.com/flashplatform/benefits/, Date Visited: 2011-01-11
7. D. Ross, Adobe Mobile Study Reveals Consumer Preferences for Accessing Consumer Products and Shopping, Financial Services, Media and Entertainment and Traces from Mobile Devices, *in Adobe Survey*, *[Online]*, Available: http://www.adobe.com/aboutadobe/pressroom/pressreleases/pdfs/201010/101310AdobeMobileExperienceSurvey.pdf, Date Visited: 2011-01-08
8. www.whatwg.org, HTML Living Standard, [*Online*], Available http://www.whatwg.org/specs/web-apps/current-work/
9. J. Huang, Q. Xu, B. Tiwana, Z. M. Mao, M. Zhang, P. Bahl, Anatomizing Application Performance Differences on Smartphone, *[Online]*, Available: http://research.microsoft.com/pubs/131377/MobiSys10-3GTest.pdf, Date Visited: 2011-02-04
10. D. Gavalas, D. Economou, Development Platforms for Mobile Applications: Status and Trends, i*n IEEE Software*, Jan.-Feb. 2011, vol. 28pp. 77
11. X. Liu, A. Sridharan, S. Machiraju, M. Seshadri and H. Zang, Experiences in a 3G Network: Interplay between the Wireless Channel and Applications, *in Proceedings of ACM MOBICOM*, 2008
12. R. Mahajan, M. Zhang, L. Poole and V. Pai. Uncovering Performance Differences in Backbone ISPs with Netdiff, *in Proceedings of NSDI*, 2008
13. K. Mattar, A. Sridharan, H. Zang, I. Matta and A. Bestavros, TCP Over CDMA2000 Networks: A Cross-Layer Measurement Study, *in PAM*, 2007
14. P. Lubbers, B. Albers, F. Salim, Pro HTML5 Programing, Apress, 2010
15. B. G. Kim, H. Yun, C. C. Lee and W. L. Kettinger, Determinants of Actual Use of Mobile Web Browsing Services in Korea: An Empirical Examination, *in 2010 Ninth International Conference on Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR)*, June 2010, pp 124
16. R. M. Ijtihadie, Y. Chisaki, T. Usagawa, H. B. Cahyo, A. Affandi, Offline Web Application and Quiz Synchronization for e-learning Activity for Mobile Browser, *in TENCON 2010 – 2010 IEEE Region 10 Conference*, 2010, pp 2402.
17. L. Hai, Y. Chadli, B. Fourestie, M. Faye; Moving the Entire Service Logic to the Network to Facilitate IMS-Based Services Introduction, *in 2010 14th International Conference on Intelligence in Next Generation Networks (ICIN)*, 2010, pp. 1
18. Tcpdump, [*Online*] Available: http://www.tcpdump.org
19. AT&T, 1946: First Mobile Telephone Call, [*Online*] Available: http://www.corp.att.com/attlabs/reputation/timeline/46mobile.html, Visited: 2011-03-21
20. Cisco, Cisco Visual Networking Index: Global Mobile Data traffic Forecast Update, 2010-2015, [*Online Whitepaper*], Available:

http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf

21. C. Heilmann, Beginning JavaScript with DOM Scripting and Ajax : From Novice to Professional, 2006, Apress

22. Z. Zhuang, Tae-Young Chang, R. Sivkumar, A. Velayutham, Application-Aware Acceleration for Wireless Data Networks: Design Elements and Prototype Implementation, *in IEEE Transactions on Mobile Computing*, Sept. 2009, pp. 1280.

23. Apache HTTP Server Project, [*Online*], Available: http://httpd.apache.org/, Visited: 2011-02-10.

24. WinPcap, [*Online*], Available: http://www.winpcap.org/, Visited: 2011-02-15

25. Wireshark, [*Online*], Available: http://www.wireshark.org/, Visited: 2011-02-02

26. Tcpdump, [*Online*], Available: http://www.tcpdump.org/, Visited: 2011-01-25

27. LibPcap, [*Online*], Available: http://www.tcpdump.org/, Visited: 2011-01-25

28. Microsoft, Task Manager, [*Online*], Available: http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/taskman_whats_there_w.mspx?mfr=true, Visited: 2011-02-12

29. Process Monitor Widget, [*Online*], Available: https://market.android.com/details?id=ccc71.pmw&hl=en, Visited: 2011-02-18

30. Info world, HTML5 deep dive, [*Online*], Available: http://www.infoworld.com/d/html5/download-the-html5-deep-dive-456, Visited: 2011-01-01

31. M. Saunders, P. Lewis, A. Thornhill, Research Methods for Business Students, 2003 Third Edition, Pearson Education Limited.

32. W. Hu, H. Yuan, J. Wang, L. Wang, The Research and Application of Power System Visualization Based on HTML5, *in 2011 4th International Conference on Electric Utility Deregulation and Reconstructing and Power Technologies (DRPT)*, July 2011, pp. 1562-1565.

33. A. Wessel, M. Purvis, J. Jackson, S. Rahman, Remote Data Visualization through WebSockets, *in 2011 Eight International Conference on Information Technology: New Generations (ITNG)*, April 2011, pp. 1050-1051.

34. A. Taivalssari, T. Mikkonen, M. Anttonen, A. Salminen, The Death of Binary Software: End User Software Moves to the Web, *in 2011 Ninth International Conference on Creating, Connecting and Collaborating through Computing (C5)*, Jan. 2011, pp. 17-23.

35. H. Nakano, T. Nagai, J. Yunpeng, M. Wannous, T. Kita, Mashup Approach for Embedding Algebraic Manipulations, Formulas and Graphs in Web Pages, *in 2011 IEEE Global Engineering Education Conference (EDUCON)*, April 2011, pp. 691-694.

36. www.adobe.com, Statistics: PC penetration, [*Online*], Available: http://www.adobe.com/products/flashplatformruntimes/statistics.html

APPENDIX A

Smartphone Details:

1. HTC HD2

    CPU: 1GHz Snapdragon Processor

    RAM: 448 MB

    OS: Windows Mobile 6.5 Professional

    3G: 7.2 Mbps download and 2 Mbps upload speed

    Wi-Fi: IEEE 802.11 b/g

2. HTC Desire:

    CPU: 1 GHz

    RAM: 576 MB

    OS: Android 2.2

    3G: 7.2 Mbps download and 2Mbps upload speed

    Wi-Fi: IEEE 802.11 b/g

DNS Resolver Details

1. Wi-Fi:
    a. Resolver: hihat.bth.se
2. 3G:
    a. Telia resolver: resolver1-g-fo.skanova.com
    b. Tele2 resolver: dns1.swip.net