Performance Measurement of Web Services from Wireless Local Area Network using KITE

Shoyeb Al Mamun Shohag
MS in Computer Networks, Dept. of ICT
KTH-The Royal Institute of Technology
SE-100 44, Stockholm Sweden

Md. Ashaduzzman Ashad
MS in Computer Networks, Dept. of ICT
KTH-The Royal Institute of Technology
SE-100 44, Stockholm, Sweden

ABSTRACT
Web service is the most common and popular application to exchange information over internet. It makes tasks easy, cost-effective, and comfortable from single user to corporate levels. Integration of different organizations or sub divisions of an organization can be done using web service with most efficient and cost effective way. To do this type of task, just enable the web services or configure web server does not mean that it will run well during peak hour or busy time. The most important issue is the performance of those services that means how much efficiently it will run or serve the purpose. To achieve the perfect service, the key concern things are response in shortest period of time and availability. The purpose of this study was to take a test based performance measurement of web services from Wireless Local Area Network (WLAN) with Keynote Internet Testing Environment (KITE) features those includes Instant Test, Long Term Measurement and Script Testing. The main reason for taking this test with various locations of wireless local area network was to observe and identify how delay variation of response time of web services occurred with the changing of each network components and it has been done with the available feature of KITE.

General Terms
Computer Networks

Keywords
Web Services, Response time, Availability, KITE (Keynote Internet testing Environment), Wireless local area network (WLAN), Instant test, Long term measurement, Scrip test, Network components.

1. INTRODUCTION
Web services standardized three ways those includes Enterprise web service, Internet web service, XML (Extensible Markup Language) base web services. Enterprise web services works with XML (Extensible Markup Language) and Web Services Description Language (WSDL) that defines the protocol and format for web services. Enterprise web service may use transport protocols or proprietary based application messaging. Suppose a service sends a message-based Simple Object Access Protocol (SOAP) message over IBM MQ Series messaging systems using Java Messaging Services (JMS). That employs XML syntax to send text commands across the Internet using HTTP (Hyper Text Transfer Protocol). Internet web services is also enterprise web service but use only open application messaging or transport protocol such as a service that sends XML messages over HTTP. XML web service is subset of Internet web service that must use W3C's (World Wide Web Consortium) adopted XML-based messaging protocol over a narrow range of transport protocols. Specifically, XML Web services will only send SOAP messages, and only send them over HTTP, SMTP (Simple Message Transport Protocol), or raw TCP/IP connections [1]. Hypertext Transfer Protocol (HTTP) is an application layer request–response protocol standard for client-server interaction on the World Wide Web (WWW). The client - server interaction indicates the sharing of program for an application where server gives services to one or more clients and the client's initiates request for such kind of services. For example, a browser running on an end user computer sends a request message to web server to get access to the stored data on that web server and server uses response message to reply to client's request. In general, there are two kinds of HTTP messages those are request messages and response messages [2].

Keynote Internet Testing Environment (KITE) is one of the top tier free web-based tools for measurement and test of web services. Keynote has been developed in the year of 1995. When a web site is in under development that is in test phase, it is very difficult to predict how that web site, content and application will run on actual browser, network and mobile devices [3]. KITE used few network components to trace the exact transaction between browser and WinInet. WinInet stands for Windows Internet Application Program Interface and its function is to connect the browser to internet via TCP/IP protocol suite to work with HTTP, FTP (File Transfer Protocol) and Gopher protocol [4]. Network component consist of Estimated cache time, DNS_time, Initial connection, SSL (SSL_handshake_time), Redirection, Request time, Time_to_first_byte, Base page download, Content download, Client time, Total measured time and Total bytes downloaded[5].

The next part of this paper is structured as follows. An overview of related research works have been point out in section-2. Section-3 provides the case study of this entire paper and section-4 holds the findings and analysis of our case study. A conclusion is remarks in section-6.

2. RELATED WORK
So many works on performance measurement of web services has been done. Few of them have been studied and presented with this section. The methodology for testing the scalability and performance of a specific web services application has been presented in [6]. Author presents an E-learning web services portal framework and implemented architecture of the UbiLern system for web based distributed learning portal system. He tested the scalability, performance of specific web service under

two different circumstances, namely LAN and DSL environment. Finally, the author suggests a proxy-based approach that accepts connections and schedules them accordingly to support huge concurrent requests.

According to [7], the authors defined and implemented a generic architecture applicable for different services within different business activities. They also implemented a complete prototype of a service-oriented Conference Management System (CMS) using web service technology. Writers experienced their monitoring and measurement architecture using the implemented application and assessed successfully the scalability of web service. They showed how QoS vary under many load conditions, and how to prevent service from QoS degradation. Their opinion is to limit the number of concurrent requests addressed to a service to less than 50 requesters.

In [8], the authors focused how packet loss effect on response time of web service and transport layer behaviour. To cover the packet loss effect, they defined modified retransmit calculation and focused on early retransmit, smart framing and adaptive duplicate acknowledgement.

The authors in [9], present the architecture and design of the SAMGRID web services, and describe current functionality of that service. They also discussed various dataset and cataloguing services, the techniques used for delivering data files to end users, service testing and performance measurements, and deployment plans. Finally, their results show very good performance of web services under various load conditions and with different usage patterns.

An overall architecture and the evaluation of a middleware infrastructure which provides quality-of-service differentiation among classes of communication-bound processes are presented in [10]. They have proposed an architecture that supports different classes of service, each with different quality attributes concerning the network data delivery performance. The authors has showed, their proposed architecture is able to provide a class of service, namely guaranteed service class, which is suitable for increasing the service availability for a group of premium users, especially in overloaded servers.

3. CASE STUDY

The case study for this entire paper has been carried out using a free web based tool named KITE with respect of three categories those are Instant test, Long term measurement and Script test.

3.1 Test Diagram

In this part, a network diagram have been demonstrated which is used to take the test and measurement for this entire paper. The network that has been used is an existing system of KTH at Haninge Campus. The network diagram has been illustrated in the above Fig 1 that shows how we were connected to KITE and an overall idea about the used devices with this case study. An URL named http://www.kth.se has been chosen for this measurement and is running on the web server of KTH.

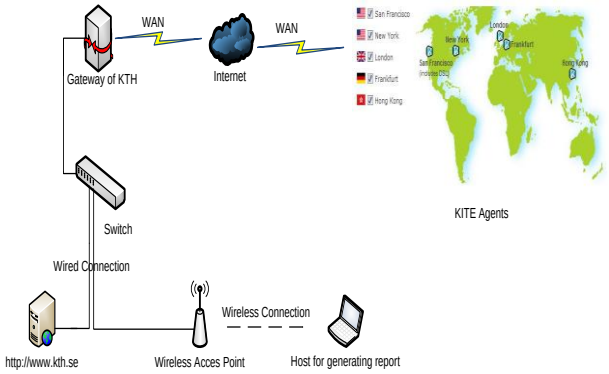

Fig 1: Test diagram

The devices that have been used with this test, were few wireless access points those are pre-installed and pre-configured with different locations in each floor of Haninge campus at KTH, a web server which is also pre-configured, and a host computer (Pentium(R) dual-core CPU T4200 @2.00 GHz, RAM 3.00 GB, HDD 250 GB) on which KITE is installed to run the test and to collect the data from this measurement.

3.2 Instant Test

Three instant tests have been taken from three different location of wireless local area network. As three tests have been taken same way, summary and graph of one test has given bellows. Fig 2 bellows is an instant test graph that we took at 5th floor. This graph represents the user experience time in second and how much byte downloaded from six global agents of KITE individually.

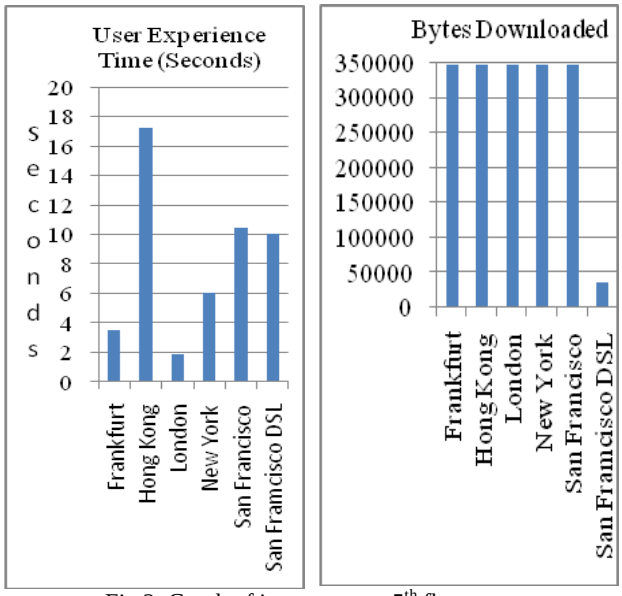

Fig 2: Graph of instant test at 5th floor

Tab 1: Summary of Instant Test at 5th floor

Locations	User Experience Time (Sec)	Network Time (Sec)	Content Error	Byte Downloaded	Status
Frankfurt	3.515	3.3	0	344858	✓
Hong Kong	17.251	17.042	0	344899	✓
London	1.907	1.704	0	344880	✓
New York	6.08	5.868	0	344899	✓
San Francisco	10.42	10.21	0	344880	✓
San Francisco DSL	10.067	9.825	0	34480	✓

Tab 2: Response time for an agent (Frankfort) at 5th floor

All Time Ranges	Frankfurt Action 1	
Network Components	Avg. Time (Sec.)	%
DNS Lookup	0.002	.06
Initial Connection	0.032	.97
SSL	-	-
Redirection	-	-
Request Time	0.00	0.00
First Byte Download	0.373	11.30
Base Page Download	0.027	0.82
Client Time	0.000	0.00
Content Download	2.866	86.85
Total Measurement Time	3.300	
Average Bytes Downloaded	344858	

A summary of instant test has been illustrated in Table 1 above that includes user experience time, network time, content error and total bytes downloaded. The network times is the summation of all network component times those includes Estimated cache time, DNS_time, Initial connection, SSL (SSL_hand_shake_time), Redirection, Request_time, Time_to_first_byte, Base page download, Content download, Client_time, Total Measured Time and Total Bytes Downloaded. Table 2 shows values for each network components that helps us to understand easily how KITE calculates the response time (network time) for each agent based on the network components. We can calculate the response time for one agent (Frankfurt) using Table-2 bellows as, Response time = (0.002+0.032+0.373+0.027+2.866) = 3.30 sec

3.3 Long Term Measurement

A continuous seven days (the time rage is absolute, from 23-JUL-10 12 AM Western European Daylight Time to 30-JUL-10 12 AM Western European Daylight Time) measurement has taken on the selected URL (http://www.kth.se) under these long term scheme of KITE.

Table-3: TCP Connection error

Name	Value
Content Error	1
Time	28-Jul-2010 06:34:21
Error _Code	TCP Connection Error
Agent	HF Transaction Seoul Kornet
Backbone	Kornet
City	Seoul, Kornet
DNS Lookup	0.002
Initial Connection	21.066
SSL	0
Redirection	0
Request Time	0
First Byte Download	0
Base Page Download	0
Client Time	0
Content Download	0
User Time	21.068
Cached Delta	21.068
Custom Component 1	0
Custom Component 2	0
Custom Component 3	0
Measurement	21.068
Total Byte Downloaded	0
Total Object Count	1

The findings from this long term observation have been mentioned bellows with various types of graphs. These continuous measurements has taken from eleven agents of KITE those are Beijing Netcom, Berlin level3, Dallas AT&T, Dallas Verizon, Delhi VSNL, London level3, New York sprint, Paris orange, Francisco sprint and Seoul Cornet.

Table-4: Transaction error

Name	Value
Content Error	0
Time	23-Jul-2010 03:44:57
Error _Code	Transaction Timed Out
Agent	HF Transaction Seoul Kornet
Backbone	Kornet
City	Seoul, Kornet
DNS Lookup	0.002
Initial Connection	0.421
SSL	0
Redirection	0
Request Time	0
First Byte Download	0.42
Base Page Download	0.018
Client Time	0
Content Download	55.761
User Time	60
Cached Delta	0
Custom Component 1	0
Custom Component 2	0
Custom Component 3	0
Measurement	56.622
Total Byte Downloaded	63126
Total Object Count	6

3.3.1 Scatter Plot Graph

It was easy to identify exactly what type of error took place with the transaction from each measurement point and what is the reason behind this error with this scatter plot graph .Fig 3 bellows (next page) represents a scatter plot graph, in which the red color triangle solid box denotes error (TCP connection error, Transaction timeout); the yellow color solid circle indicates the content error and the green color dotted point represent the successful transaction of each agents. A Snapshot of TCP connection error and connection time out has given in the Table 3 and Table 4 that comes up with this scatter plot graph

3.3.2 Agent Wise Graph

A snap shot of agent wise graph has given in the Fig 4 and Fig 5 bellows that have been taken from the continuous seven days observation of long term measurement. It gives an overall idea about total response time in second and availability in percentile basis for each agent.

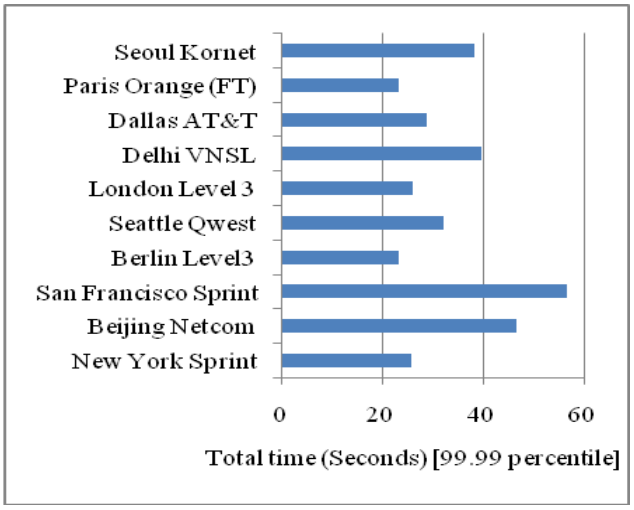

Fig 4: Response time from each agent

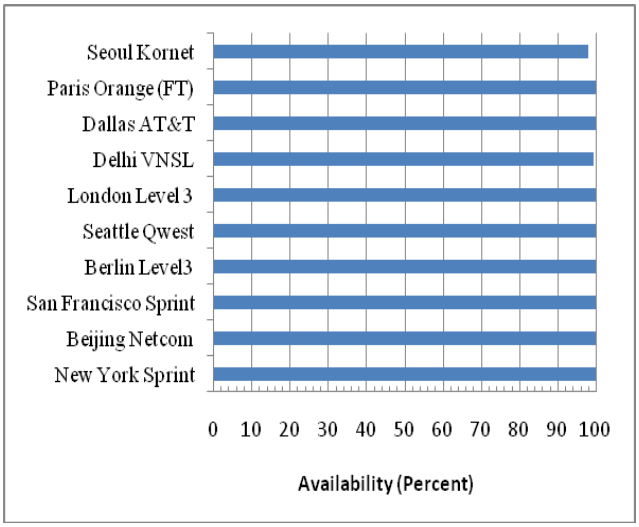

Fig 5: Availability of each agent

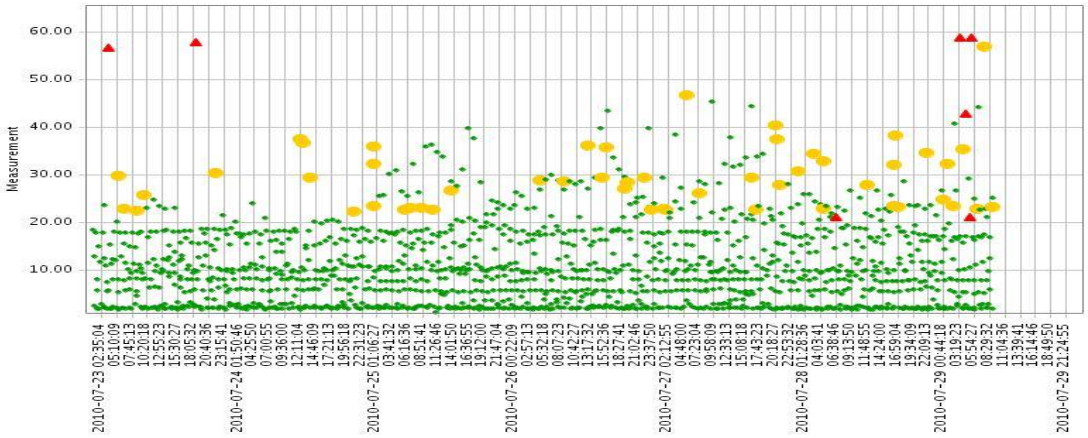

Fig 3: Scatter Plot Graph

Table 5: Response time and availability for each agent

Agent	www.kth.se(TxP)-www.kth.se]		Avail %
	DNS (Sec) [99.99 percentile]	Data Points	
New York Sprint	25.833	150	100.00
Beijing Netcom	46.731	102	100.00
San Francisco Sprint	56.640	150	100.00
Berlin Level3	23.209	156	100.00
Seattle Qwest	32.344	153	100.00
London Level 3	26.094	143	100.00
Delhi VNSL	39.725	149	99.33
Dallas AT&T	28.876	153	100.00
Paris Orange (FT)	23.435	151	100.00
Seoul Kornet	38.358	149	98.03
Percentile	www.kth.se(TxP)-[www.kth.se]		
	DNS (Sec) [99.99 percentile]	Data Points	Avail %
	0.681	1454	99.52
Trimmed Data Points	0		

A summary of response time and availability from this long term measurement has given in the Table 5 above, from this table it's seen that the agent Berlin Level holds the lowest response time with the 100 percent availability. This is because the network component of this agent takes lowest time. The San Francisco sprint agent takes longer time with 98 percent availability; this is happened due to having the longer time

Table 6: DNS time for each agent

Agent	www.kth.se(TxP)-www.kth.se]		Avail %
	DNS (Sec) [99.99 percentile]	Data Points	
New York Sprint	0.704	150	100.00
Beijing Netcom	0.479	102	100.00
San Francisco Sprint	0.107	150	100.00
Berlin Level3	0.047	156	100.00
Seattle Qwest	0.038	153	100.00
London Level 3	0.017	143	100.00
Delhi VNSL	0.014	149	99.33
Dallas AT&T	0.014	153	100.00
Paris Orange (FT)	0.013	151	100.00
Seoul Kornet	0.003	149	98.03
Percentile	www.kth.se(TxP)-[www.kth.se]		
	DNS (Sec) [99.99 percentile]	Data Points	Avail %
	0.681	1454	99.52
Trimmed Data Points	0		

for each network component. It was easy to find what values were for each network components (like DNS time, initial connection time) with each agent individually. For convenience, only a summary of DNS time for all agents has given to Table 6.

Table 8: Script test result of each action for Hong Kong

All Time Ranges	Action 1		Action 2		Action 3		Action 4		Action 5		Action 6		Action 7	
Networks Components	Avg. Time	%	Avg. Time	%	Avg. Time	%	Avg. Time	%	Avg. Time	%	Avg. Time	%	Avg. Time	%
DNS Lookup	0.002	0.01	0.000	0.00	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Initial Connection	0.309	1.62	0.309	8.23	0.327	8.57	0.306	3.56	0.328	2.63	0.00	0.00	0.00	0.00
SSL	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Redirection	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Request Time	0.000	0.00	0.000	0.00	0.00	0.00	0.00	0.00	0.000	0.00	0.00	0.00	0.00	0.00
First Byte Download	0.313	0.313	0.312	8.31	0.331	8.69	0.309	3.59	0.609	4.87	0.314	10.99	1.581	20.97
Base Page Download	0.015	0.08	0.002	0.05	0.002	0.05	0.002	0.02	0.001	0.01	0.010	0.35	0.003	0.04
Client Time	0.000	0.00	0.000	0.00	0.00	0.00	0.001	0.01	0.003	0.02	0.000	0.00	0.015	0.20
Content Download	18.409	96.65	3.130	83.40	3.151	82.68	7.988	92.82	11.552	92.47	2.532	88.66	5.941	78.79
Total Measurement Time	19.048		3.753		3.811		8.606		12.493		2.856		7.540	
Average Byte Downloaded	144589		147384		103877		103877		197390		28445		111329	

Table 7: Script test result of each action for Hong Kong

Location	User Experience Time (Sec.)	Network Time (Sec.)	Content Error	Byte Downloaded	Status
Frankfurt	13.515	12.44	0	1212840	✓
Hong Kong	60.031	58.108	2	10063246	×
London	11.437	10.614	0	1212766	✓
New York	25.105	24.201	0	1212870	✓
San Francisco	51.059	50.074	0	1212828	✓
San Francisco DSL	0	0	0	0	×

3.4 Script Test

A test script is a series of web page navigations and user actions recorded to test a site. To work with script test, a script has been created with few transactions on the selected URL (http://www.kth.se). This recorded script has been tested from two different locations (5th and 6th floor) of wireless local area network in Haninge campus at KTH. The output format of script test result is similar to instant test result that means it shows the response time, user experience time in second, content error, bytes downloaded and status of the scripts. For convenience, a summary of script test that has been taken at 6th floor has given in the Table 7.

It was easy to get what response time is coming for each action with each network component more specifically that is not possible with instant test. A detail summary for an agent (Hong Kong) given in the above Table 8. It contains data for each network component of each transaction individually with respect to a single agent (Hong Kong). Response time for Hong Kong can be calculated with the available data of Table 8 as, Response time = Summation of the total measurement time for each transaction (action) = (19.048+3.753+3.811+ 8.606+12.493+2.856+7.540) = 58.108 sec.

Table 9: Instant test

Agents	5th –floor–WLAN		6th –floor–WLAN		7th –floor–WLAN	
	Response Time (sec)	Byte Downloaded (Byte)	Response Time (sec)	Byte Downloaded (Byte)	Response Time (sec)	Byte Downloaded (Byte)
Frankfurt	3.3	344858	2.3	344880	2.575	344882
Hong Kong	17.042	344899	21.99	344898	18.149	344899
London	1.704	344880	2.008	344880	1.727	344881
New York	5.868	344899	5.885	344898	5.804	344899
San Francisco	10.21	344880	31.745	331428	10.12	344880
San Francisco DSL	9.825	344880	12.432	344880	10.86	344880

Table 10: Script test

Agents	5th-floor-WLAN			6th-floor-WLAN		
	Response Time (Sec)	Byte Downloaded (Byte)	Status	Response Time (Sec)	Byte Downloaded (Byte)	Status
Frank Fort	26.403	1212992	✓	12.44	1212840	✓
Hong Kong	58.477	1008909	☒	58.108	1063246	☒
London	24.838	1212946	✓	10.614	1212766	✓
New York	34.308	1213027	✓	24.201	1212870	✓
San Francisco	50.084	1212993	☒	50.074	1212828	☒
San Francisco DSL	42.294	1212975	✓	0.00	0.00	☒

4. FINDINGS AND ANALYSIS

An overall discussion with the findings of three test phase (instant test, script test and long term measurement) has given bellows.

4.1 Findings and Analysis of Instant Test

For the convenience of discussion and analysis, Table-9 has been creates with the available data from instant test. It consists the response time and byte downloaded for each agent with in the three different locations of wireless local area network. From these three instant tests result (Table 9), it's seen that the response time vary from each other and this is because of the changing of location for each instant test with in the wireless local area network. . It's also observed that the response time for Hong Kong is always high, the region behind this could be due to the geographical distance between the place where the website is hosted (kth.se in Sweden) and Hong Kong itself as well as the internet speed in that location.

The other things that could be behind this high response time, time takes by each network component like DNS resolve, redirection, initial connection, base page download, content download, SSL connection and estimated cache time then as usual. The second highest response time comes from agent San Francisco and San Francisco DSL, the possible cause behind this would be the Speed of the DSL connectivity that can vary based on the ISP (Internet Service Provider) in different regions (Country). For example, in USA the typical speed for Dial-up connectivity is from 1.5 Mbps to 3 Mbps that depends on the ISP company of that location [11] .Table-9 also shows two agents (New York and London) takes lowest response time to download maximum bytes although geographically these two agents are quite fare from the hosted domain (kth.se) location, this is could be of good internet link speed and low latency of those agents and this would be due to having good routing policy, less hop count and low latency of system [12].

4.2 Findings and Analysis of Long Term Measurement

The scatter plot graph which has been created from long term measurement scheme (Fig 3) for a period of seven days, and it's observed that two types of error has occurred those are TCP connection error and Transaction error (Table 3 and Table 4). A TCP error can take place when the server is temporarily down or busy too much with other types of problem like multiple requests are coming that preventing it from responding to requests. From Table-3 it's seen that this error has occurred having the longer time of initial connection that is 21.066 sec. Table-4 represents a transaction error which has occurred for the longer time of content download that is 55.761sec. The initial connection time with this transaction error was too short that is 0.421 sec that indicates the TCP connection with this transaction was established successfully. Agent wise graph of Long term

measurement (Fig 4, Fig 5 and Table 5) in the previous section (Case study) also carries response time and availability of the tested website for each agent and it was easy to identify from which agents long response time is coming.

4.3 Findings and Analysis of Script Test

Table-10 holds the script test results of two elements that are the response time and total byte downloaded for each six agents individually. This table has been creates with the available data of scripts test in the previous chapter (case study). From Table-10 above, it's found that the response time and total byte downloaded for each agent varies from each other this is because of the geographical region of agents and the other issues could be the changes of location within the wireless local area network for each test. Status option gives an idea about the minimum or acceptable value of response time, although there is no pre-defined value of response time, but it should be less as much as possible. If we look at Table-7 above, it's observed that when response time comes more than 50 sec is not acceptable and count as error. This could be for few other cause like having longer time for each network component, or server is busy too much with other task that prevent respond to request. It's possible to clarify more accurately with the available data of Table-8, it is found that response time for each action is different and the first action takes almost four times greater than other six actions, and this is because content download component takes longer time (18.409 sec) for the first action.

5. ACKNOWLEDGEMENT

We are heartily thankful to Thomas Lindh, who has given solid idea that makes us knit interest to paper writing especially on performance measurement fields. We are also thankful to our parents for their unbounded love and blessing for ever and friends for their great cooperation.

6. CONCLUSION

The main things that have been focused with this paper were to take the necessary test and measurement to observe the performance of the selected web sites (kth.se) that has been done with the case study phase. The goal of the case study was to identify how the response time and availability are affected by the network component, and how delay variation of response time occurred with the changing of location for each test phase with in the wireless local area network that have been observed by collecting data using a web based tools named KITE and discussed in the previous section (findings and analysis). The tools that we have used for this paper was a free web based tools but there was limitations with the available features. Our intention was to create a host computer as an agent that keynote has and it was fixed. It would be more interesting work, if we could create an agent by our self with in the wireless local area network but this feature is not available with KITE presently. We have contact with support centre of Keynote and an agent (Aniket Damle, aniket.damle@keynote.com) via mail and phone regarding this issues, they have creates a ticket with this issues and informed to the development teams for near future. Finally, we have tried to our best to work for this paper with the available features of KITE which were available and related to web services performance measurements.

7. REFERENCES

[1] Holt Adams (rhadams@us.ibm.com), Senior Solution Consultant, IBM j Start, Best Practices for Web services, http://www.ibm.com/developerworks/webservices/library/ws-best9/index.html

[2] Behrouz A Forouzan, 4th edition, TCP/IP Protocol suite

[3] Keynote at a glance , http://www.keynote.com/docs/datasheets/Keynote_Overview.pdf

[4] Mahesh Chad, August 2000, WinInet tutorial for beginners Components, http://www.dotnetheaven.com/uploadfile/mahesh/wininetintroduction05242005061945am/wininetintroduction.aspx%5c

[5] About Transaction Perspective and Application Perspective Components, http://keynote.com/mykeynote/help/abouttxn4components.asp#cached

[6] Abdulmotaleb El Saddik, October 2006, Performance measurements of web services based applications. IEEE Transactions on Instrumentation and Measurement, 55(5):1599–160

[7] Riadh Ben Halima, Emna Fki, Khalil Drira, Mohamed Jmaiel, 2009, 1st IEEE Workshop on Performance evaluation of communications in distributed systems and Web based service architectures, Sousse : Tunisia.

[8] Johan Garcia, Per Hurtig, Anna Brunstrom, Department of Computer Science, Karlstad University, The effect of packet loss on the response time of web service

[9] S. Veseli, FNAL, Batavia, IL 60510, USA, Samgrid web services

[10] D. Cotroneo, M. Gargiulo, S. Russo, G. Ventre, cotroneo, giorgio*g*@unina.it, mauro.gargiulo@napoli.consorziocini.it, improving the availability of web services

[11] Verizone Telephone Line Specification, http://www.ehow.com/list_7719005_verizon-telephone-line-specifications.html

[12] The real world - Speed Enemies, www.dslreports.com/speed