Throughput and Delay Analysis in a Real Time Network

Umeh O. A., Akpado K.A., Okechukwu G. N., Ejiofor H. C.

Abstract— **The aim of this paper is to evaluate the throughput and delay performance of IEEE802.11 WLANs using different network performance monitoring tools. It also shows detailed procedure on how to use these network monitoring tools (NetStress, Wireshark and Jperf) to monitor a real time network. The study was carried out to establish the effects of varying the number of work stations against the network performance in a real WLAN environment. The result shows that maximum throughput and minimum delay can only be achieved in the non saturated case, that is, it is dependent on the number of active nodes.**

***Index Terms*—WLAN, Network, Time, delay, throughput, NetStress, Wireshark, Workstations and Jperf**

I. INTRODUCTION

Managing a network without monitoring is similar to driving a vehicle without a speedometer or a fuel gauge. Network monitoring is the use of logging and analysis tools to accurately determine traffic flows, utilization, and other performance indicators on a network. Good monitoring tools give you both hard numbers and graphical aggregate representations of the state of the network. This helps you to visualize precisely what is happening, so you know where adjustments may be needed (http://wndw.net).
There are several benefits to implementing a good monitoring system for your network:

- Network budget and resources are justified. Good monitoring tools can demonstrate without a doubt that the network infrastructure (bandwidth, hardware, and software) is suitable and able to handle the requirements of network users.
- Network intruders are detected and filtered. By watching your network traffic, you can detect attackers and prevent access to critical internal servers and services.
- Troubleshooting of network problems is greatly simplified. Rather than attempting "trial and error" to debug network problems, you can be instantly notified of specific problems. Some kinds of problems can even be repaired automatically.
- Network performance can be highly optimised. Without effective monitoring, it is impossible to fine tune your devices and protocols to achieve the best possible performance.
- Capacity planning is much easier. With solid historical performance records, you do not have to "guess" how much bandwidth you will need as your network grows.
- Proper network usage can be enforced. When bandwidth is a scarce resource, the only way to be fair to all users is to ensure that the network is being used for its intended purpose.

Fortunately, network monitoring does not need to be an expensive undertaking. There are many freely available open source tools that will show you exactly what is happening on your network in considerable detail. This paper will help you identify some invaluable tools and how best to use them (http://wndw.net).

A number of researches have been done to study the performance on IEEE 802.11 especially on DCF mechanism both in real-time and using computer modeling and simulation technique. The throughput, frame discard probability, and average frame delay performance of DCF in IEEE 802.11 are analyzed and evaluated for different parameters in (Chen and Liu, 2010). In (Amjad & Shami, 2006), the authors propose a new MAC protocol that modifies the DCF protocol such that the channel utilization can be improved with successful packet transmissions, yielding higher throughput performance. Certain research papers aim to propose new analytical models for investigating the impact of some factors on the performance of IEEE 802.11. In Zheng et al., (2006), the authors provide a new analytical model that considers the channel error conditions in the unsaturated mode to analysis delay and throughput performance of IEEE 802.11 DCF under different incoming traffic conditions. Ivanov et al., (2010) consider the 802.11 networks which are defined in terms of throughput requirements and packet loss probability limitations and the influence of sizes of packets being transmitted through the network on the QoS is investigated. Yang et al., (2009) analyze the goodput of a WLAN with hidden nodes under a non-saturated condition.

This paper makes an in-depth investigation into the performance of Quality of Service parameter (Throughput and Delay) as the number of host varies. Three different tools were used to carry out real time network performance measurement, the tools include; NetSress, Jperf and WireShark

2.0 Real Time Network Performance Measurement

Manuscript received.
Umeh O. A., Electronic and Computer Engineering, Nnnamdi Azikiwe University, Awka, Anambra State, Nigeria.
Akpado K.A., Electronic and Computer Engineering, Nnnamdi Azikiwe University, Awka, Anambra State, Nigeria.
Okechukwu G. N., Electronic and Computer Engineering, Nnnamdi Azikiwe University, Awka, Anambra State, Nigeria.
Ejiofor H. C., Electronic and Computer Engineering, Nnnamdi Azikiwe University, Awka, Anambra State, Nigeria.

Three different networks sniffer software were used to carry out the measurement. The measurement of throughput and delay were taken at intervals as the number of users' increases or decreases.

2.1 Experimental Environment

This experiment was carried out at Chams Plc office (Old Secretariat Building by Arroma Junction) Awka, it's an indoor environment. Three different cases were considered in the experiment. Their network is divided into three regions namely; Data Capture Center (Ground Floor), Data Update Center, and Data upload/card center which also houses the main Server. These three regions have different ranges from the access points. Each of these centers houses between 15 to 40 fixed desktop computers and few laptops. Some mobile smart phones were also used as host. All the systems have wireless LAN features and are IEEE 802.11 compliant. The network was transmitted to the three regions, each via Nanostation M5 radio, which brings the network to the wireless Cisco Router (Cisco 871 model) and distributed via a wireless AP, LAN and WLAN switches (LinkSys), Hubs and so on. These three Centers are located in the same building but are separated with few meters way from each other.

The measurements are carried out in these three locations. In all the three regions, three different network sniffers (NetStress, Wireshark and Jperf) were used to monitor and take measurement on the network.

Netsreess: NetStress is a tool used to measure network performance, both wired and wireless. It is a simple tool that employs bulk data transfer using layer 3 protocols (TCP and UDP). Network performance is reported in terms of throughput; that is, bits (or bytes) per second.
In order to test and troubleshoot networks, there is need for tools that can generate network traffic and analyze the network's throughput performance. This is true for both wired and wireless networks. By quantifying the network's throughput performance, NetStress provides a valuable metric to assist in monitoring the health of the network or assist in troubleshooting network problems.

By comparing actual throughput with the theoretical bandwidth between the transmitter and receiver or with a measurement taken at an earlier date, it can tell whether the network is operating as expected. Variations in throughput may indicate a significant amount of other traffic, overloaded network equipment, communication errors which cause packets to be lost or, in the case of wireless networks, interference from other wireless devices. By performing tests using different machines on the network, a network engineer will begin to gain clues as to where the problem lies and which areas should be examined in greater detail (http://nutsaboutnets.com/netstress/).

NetStress includes the following features:

- single instance of the application (i.e. each instance can be used as server or server plus client)
- supports both TCP and UDP data transfers
- supports multiple data streams
- variable TCP / UDP segment size
- rate of packet transmission (Packets Per Second)
- variable Maximum Transmission Unit (MTU)
- uplink and downlink modes
- auto node discovery
- choice of display units (KBps, Kbps, MBps, Mbps)
- support for multiple network adapters

2.2 Features of Netstress

Single Instance of the Application (i.e. each instance can be used as receiver or receiver plus transmitter): The new version of NetStress follows a transmitter / receiver model, but there is no need to be concerned with this when application is launched for the first time. Now, every instance of the application includes a built-in receiver; that receiver may lie dormant or respond to transmissions from a transmitter. Also, every instance of the application can act as a transmitter. So, this provides a lot more flexibility since each instance can act as a receiver and, optionally, as a transmitter. When NetStress is launched, it will be observed that the charts have been separated as Transmitter and Receiver. When the built-in receiver receives transmissions from a transmitter then the results are displayed in the receiver's charts, and when NetStress is transmitting then the results are displayed in the transmitter's charts.

TCP and/or UDP -- plus multiple data streams: Supports TCP transmissions, UDP transmissions or both (concurrently). Furthermore, you can specify from 0 to 8 data streams for each protocol. These are configured using the Settings dialog. In addition, the timecourse results are broken down accordingly:

Total: The chart labelled Timecourse (Total) shows (a) the total throughput (both TCP plus UDP), (b) TCP alone (total of all the TCP data streams), and (c) UDP alone (total of all the UDP data streams).

TCP: The chart labelled Timecourse (TCP) shows the throughput results for each of the TCP data streams.
UDP: The chart labelled Timecourse (UDP) shows the throughput results for each of the UDP data streams.
The default setting is to test throughput performance using a single TCP data stream; this is the most common scenario.

Packet Size: Supports TCP and UDP packet size. Using the Settings dialog you can configure the packet size of the TCP and UDP data streams. Turns out this can have a slight effect on throughput.

Rate of Packet Transmission: Supports Packets Per Second (PPS) "throttle" for the transmission side. Using the Settings dialog you can configure 'packets per second' for both TCP and UDP data streams. This is useful in emulating voice and other timed applications.

Maximum Transmission Unit (MTU): Using the Settings dialog you can configure the MTU for the current interface adapter. Typically, the operating system will optimize this value when it first boots. This is not a parameter that Microsoft readily exposes since a bad value can dramatically degrade network performance -- so, experiment with it at your own risk. Though NetStress allows you to change this parameter while the application is running, it will restore the MTU to its original value when the application exits. Microsoft Windows computers default to an MTU of 1500 bytes for broadband connections and 576 bytes for dialup connections.

Display Units: Using the Settings dialog you can configure which units NetStress should use when displaying the data: e.g. KBps (KBytes per sec), Kbps (Kbits per sec), MBps (MBytes per sec) or Mbps (Mbits per sec).

2.3 Using Netstress

In order to measure throughput performance between two nodes on a network we need a transmitter and receiver. Packets are transmitted to the receiver, and the receiver, in turn, sends the result back to the transmitter. The results represent a quantifiable metric that reflects the performance of the path between the transmitter and receiver machines. By selecting the transmitter and receiver machines at various points within the network one can analyze critical portions of the data path.

Each instance of NetStress can run as both a transmitter and receiver, or just as a receiver. If you wish for a particular instance of NetStress to run just as a receiver then it is not necessary to specify a remote receiver IP address. On the other hand, if you wish for a particular instance of NetStress to run as both a transmitter and receiver, then use the 'Remote Receiver IP' button along the main menu to select or enter the IP address of a remote receiver. NetStress uses a UDP broadcast mechanism to automatically discover other instances of NetStress on the network that can act as a remote receiver. If no remote receivers are listed below this is probably because NetStress is not running on another machine. Then the IP address of the remover receiver can be entered manually. There is one requirement, that is the remote receiver must be reachable from the transmitter using ICMP (i.e. ping).

2.4 Procedure for Running NetStress

1) Launch NetStress on station A; call this "Node A". A dialog appears with a list of available network interfaces, choose one.
2) Launch NetStress on station B; call this "Node B". A dialog appears with a list of available network interfaces, choose one. Be aware that in order for the two instances of NetStress to be able to communicate then the interfaces you've selected must be on the same network. If there is only one network and each machine has only one NIC, then that is not a problem. But in the case of multiple networks and/or NICs then it is necessary to pay attention to this detail.

3) Let's say "Node A" will be a transmitter, then it is necessary to select a remote receiver node. Along the menu bar you'll see the label "Remote Receiver IP" and an IP address of 0.0.0.0. The star that moves from side-to-side is a visual cue that a remote receiver needs to be selected if this instance of NetStress is to act as a transmitter. Select the blinking '0.0.0.0' and the 'Select Remote Receiver dialog box will appear. Select the IP address of one of the remote receivers and press 'OK'.
4) Once a remote receiver has been selected, then the 'Settings' and 'Start' buttons along the top menu are activated. The 'Settings' dialog box allows you to control the following parameters: TCP vs UDP, Number of data streams, TCP Packet Size, UDP Packet Size, Rate of Packet Transmission (Packets Per Second), Data Flow, MTU, Display Units etc.
5) While transmitting, certain buttons on the main menu change state, for example the Stop button now becomes active. The Settings, 'Start', and 'Remote Receiver IP' buttons are temporarily deactivated.
6) To stop transmitting, press the Stop button.
7) Below the main menu there are two main windows , that is 'Transmitter' and 'Receiver'. When NetStress is acting as a transmitter then the timecourse charts in the upper window will display the results. When NetStress is acting as a remote receiver then the lower window will display the results (https://nutsaboutnets-documentation.s3.amazonaws.com/NetStress).

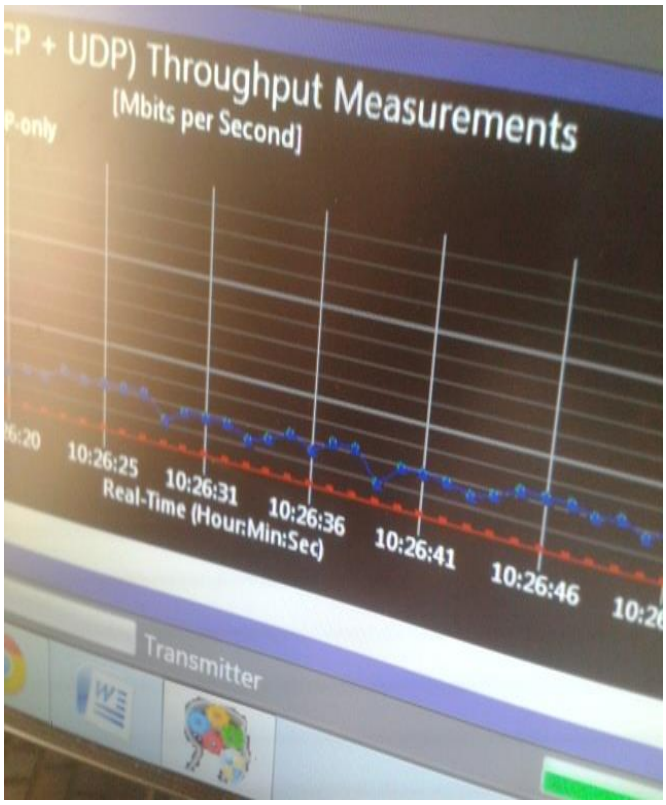


Figure 1: Pictorial Diagram of Throughput Measurement

Wireshark: Wireshark is the world's foremost open-source packet/network protocol analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. It lets you see what's happening on your network at a microscopic level. Originally named Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues.

Wireshark is cross-platform, using the GTK+ widget toolkit in current releases, and Qt in the development version, to implement its user interface, and using pcap to capture

packets; it runs on Linux, OS X, BSD, Solaris, some other Unix-like operating systems, and Microsoft Windows. Wireshark, and the other programs distributed with it such as TShark, are free software, released under the terms of the GNU General Public License (https://www.wireshark.org).
Features: Wireshark is software that "understands" the structure (encapsulation) of different networking protocols. It can parse and display the fields, along with their meanings as specified by different networking protocols. Wireshark uses pcap to capture packets, so it can only capture packets on the types of networks that pcap supports.

Data can be captured "from the wire or wirelessly" from a live network connection or read from a file of already-captured packets. Live data can be read from a number of types of network, including Ethernet, IEEE 802.11 or PPP. Captured network data can be browsed via a GUI, or via the terminal (command line) version of the utility, TShark.
Wireshark has other rich feature which includes the following:

- Deep inspection of hundreds of protocols, with more being added all the time
- Live capture and offline analysis
- Standard three-pane packet browser
- Multi-platform: Runs on Windows, Linux, OS X, Solaris, FreeBSD, NetBSD, and many others
- Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility
- The most powerful display filters in the industry
- Rich VoIP analysis

Captured files can be programmatically edited or converted via command-line switches to the "editcap" program. Data display can be refined using a display filter.
VoIP calls in the captured traffic can be detected. If encoded in a compatible encoding, the media flow can even be played. Raw USB traffic can also be captured. Wireless connections can also be filtered as long as they transverse the monitored Ethernet.

Various settings, timers, and filters can be set that ensure only triggered traffic will appear.

Wireshark's native network trace file format is the libpcap format supported by libpcap and WinPcap, so it can exchange captured network traces with other applications that use the same format, including tcpdump. It can also read captures from other network analyzers, such as snoop, Network General's Sniffer, and Microsoft Network Monitor (https://en.wikipedia.org/wiki/Wireshark).

2.5 Colour Coding

Wireshark uses colours to help you identify the types of traffic at a glance. By default, green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic, and black identifies TCP packets with problems — for example, they could have been delivered out-of-order.

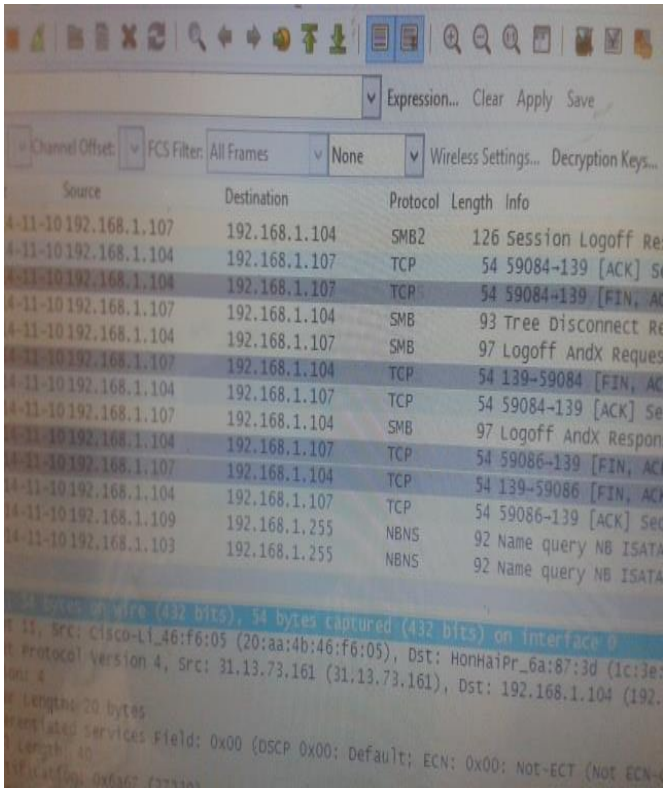


Figure 2: Pictorial diagram of Network parameter measurement

Jperf: Jperf is a tool to measure the bandwidth and the quality of a network link. Jperf can be associated with Iperf to provide a graphical frontend written in Java. The network link is delimited by two hosts running Iperf.

The quality of a link can be tested as follows:

- The bandwidth is measured through TCP tests.
- Latency (response time or RTT): can be measured with the Ping command.
- Jitter (latency variation): can be measured with an Iperf UDP test.
- Datagram loss: can be measured with an Iperf UDP test.

2.6 How to Measure Network Throughput Using Jperf

JPerf is a graphical front end for the popular network testing tool Iperf. Using JPerf you can quickly test a WAN or LAN connection to determine the network throughput and delay. The test results are automatically graphed and presented in a format that is easy to read. JPerf can also be used to detect packet loss, delay, jitter, and other common network problems.
In WLAN, JPerf testing can also be used to perform; site surveys, validate client performance, verify throughput across 802.11 bridges and access points.
JPerf provides many benefits over Iperf which is a command line only application. Jperf is reliable and easy to use. The utility is fully operational on both Windows and Linux systems.

Whether you're trying to diagnose a problem in your home network or troubleshoot the performance of a WLAN link, JPerf can handle the task (https://code.google.com/p/xjperf/).

2.7 Installing JPerf on Windows

JPerf requires that Java version 1.5 or newer is installed before it will run. You can visit Java.com to download the latest version or verify if it is correctly installed on your computer.

To get JPerf running you will need to download the jperf-2.0.2.zip file from the JPerf Google code page. There isn't an installer so simply extract the contents of the zip file to a location on your computer such as C:\JPerf. To launch the JPerf utility run 'jperf.bat'.

2.8 Setting up a JPerf Server

JPerf is designed to run as a client/server application. To run a test you will need to set up a JPerf server on your network. Then you can run a JPerf client from another location on the network which will connect to the remote server.

To start the JPerf server select the radio button labelled server then click Run IPerf. By default JPerf runs in TCP mode and listens on port 5001 (www.techrepublic.com/using-jperf-to-check-network-performance/).

2.9 Connecting a client to the server

To connect to the JPerf server to run a test there is need to first select the client radio button. In the server address field type in the IP address of the computer running the JPerf server. To begin the test click on run iPerf in the upper right hand corner of the app.

By default JPerf will run a 10 second TCP test using 1 stream. While the test is running the graph will update in real time to reflect the results.

There are several options that can be adjusted to modify the parameters of the test, these may include;

no arg. Default settings
-b Data format
-r Bi-directional bandwidth
-d Simultaneous bi-directional bandwidth
-w TCP Window size
-p, -t, -i Port, timing and interval
-u, -b UDP tests, bandwidth settings
-m Maximum Segment Size display
-M Maximum Segment Size settings
-P Parallel tests
-h help

More so, there are several TCP options that can be modified such as buffer length, window size, and Round trip Time. JPerf can also function in UDP mode, although the server must be operating in UDP mode in order for this test to work.

2.10 Application layer options in Jperf

Transmit - Run the test for a specified number of seconds, or until a certain amount of bytes have been transferred.

Output Format - Test results can be changed to display bits, bytes, kbytes, etc.

Report Interval - This adjusts how often the graph results are updated.

JPerf Tips: Below are a few useful tips for improving your JPerf results;

Use Parallel streams - The bandwidth of a single TCP session is limited by several factors. By using parallel streams you can easily saturate a very high bandwidth connection. In the JPerf client settings you can specify the number of streams to use. 10 is found to be a good number.

Run a Bi-Directional Test - By default JPerf transmits data from the client to the server. By selecting the dual testing mode under application layer options JPerf will send data in both directions at the same time.

Use a representative file - JPerf has a cool ability that allows you to select a file to be transmitted to the server during the test. This function allows you to simulate a real world data transfer across your network in a controlled manner.

Use JPerf to create iPerf commands - Since JPerf uses iPerf as a back end to run all of the tests you can use JPerf to help you build useful iPerf commands. Select the test options you want using the GUI and then copy the command it created from the box at the top of the application (www.firewall.cx).

Figure 3: Pictorial Diagram of Bandwidth Measurement

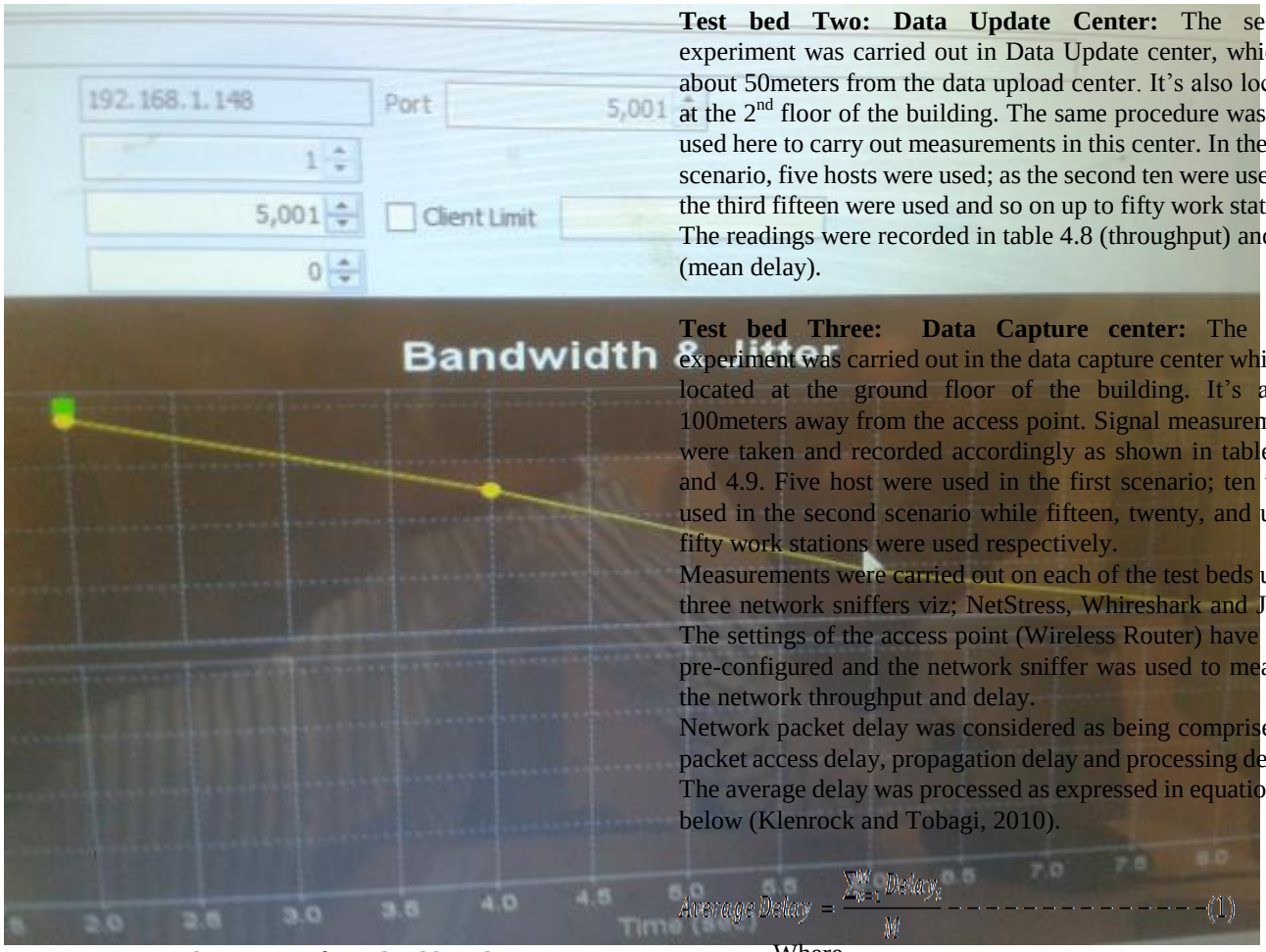
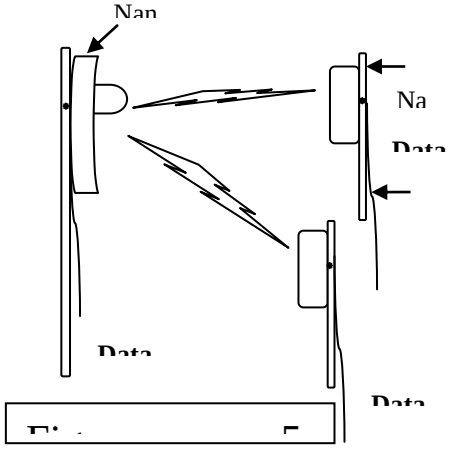


Figure 4: Pictorial Diagram of Bandwidth and Jitter Measurement

3.0 Test bed Environment

Test bed one: Data upload/card Center: The first experiment was carried out in Data upload/card Center which houses the main servers. The Data upload/card Center is located on the first floor of the old secretariat two storey building and was used for the measurements; first was with five hosts, second was with ten host, third was with 15 and it continues in that order till we have fifty work stations. The measurements of throughput and delay/jitter were carried out using the three different network sniffers one at a time.



Test bed Two: Data Update Center: The second experiment was carried out in Data Update center, which is about 50meters from the data upload center. It's also located at the 2nd floor of the building. The same procedure was also used here to carry out measurements in this center. In the first scenario, five hosts were used; as the second ten were used; in the third fifteen were used and so on up to fifty work stations. The readings were recorded in table 4.8 (throughput) and 4.9 (mean delay).

Test bed Three: Data Capture center: The third experiment was carried out in the data capture center which is located at the ground floor of the building. It's about 100meters away from the access point. Signal measurements were taken and recorded accordingly as shown in table 4.8 and 4.9. Five host were used in the first scenario; ten were used in the second scenario while fifteen, twenty, and up to fifty work stations were used respectively.

Measurements were carried out on each of the test beds using three network sniffers viz; NetStress, Whireshark and Jperf. The settings of the access point (Wireless Router) have been pre-configured and the network sniffer was used to measure the network throughput and delay.

Network packet delay was considered as being comprised of packet access delay, propagation delay and processing delays. The average delay was processed as expressed in equation (1) below (Klenrock and Tobagi, 2010).

$$Average\ Delay = \frac{\sum_{i=1}^{M} Delay_i}{M} \quad \text{----------(1)}$$

Where,

$Delay_i$
= Time taken to transmit bits from UTs to medium Access Control
+ Time taken to send the bits from Medium Access Controller to the destination
+ Time taken to retransmit bits due to collision

$$Delay_i = Time\ taken\ to\ offer\ bits - Time\ taken\ to\ serve\ bits$$

	THROUGHPUT FOR THE THREE TEST		
No of work stations	THR1	THR2	THR3
2	5.295	5.003	4.963
5	4.283	4.532	4.185
10	3.743	4.009	3.861
15	3.364	3.736	3.375
20	2.571	3.072	2.843
25	2.295	2.654	2.579
30	1.782	1.849	1.943
35	1.023	1.329	1.465
40	0.743	0.843	1.275
45	0.532	0.437	0.732
50	0.529	0.419	0.529

M = Total number of packets transmitted.

	Delay for the three regions		
Number of Work Stations	D1	D2	D3
5	0.257	0.401	0.448
10	0.474	0.504	0.621
15	0.671	0.811	0.883
20	0.894	1.16	1.207
25	1.083	1.24	1.282
30	1.16	1.237	1.294
35	1.236	1.247	1.253
40	1.252	1.269	1.301
45	1.312	1.326	1.337
50	1.354	1.377	1.398

Also recall that;

$$\text{Bite Rate} = \frac{\text{Data Size}}{\text{Transmission Time Delay}} - - - - - - - - - - (2)$$

$$\text{Transmission Time Delay} = \frac{\text{Data Size}}{\text{Bite Rate}} - - - - - - - - - - (3)$$

Note also that;

$$Data\ Size = User\ Data + Header - - - - - - - - - - - - - (4)$$

Equations (1, 2, 3 and 4) were used to compute the Transmission Delay for a particular packet. If the Date size and Bite rate are known, the Transmission Delay can be calculated.

Throughput is defined as the number of bits passing through a point in a second or it is defined as the number of packet passing through the network in a unit time (Klenrock and Tobagi, 2010), it can also be stated as;

$$\text{Throughput} = \text{Total Number of Bits offered by UTs} - \text{Number of Bits pending(unserved)} - \text{Number of Bits dropped}$$

$$Throughut = (Bits\ offered) - (Bits\ pending) - (Bits\ dropped) - - - (5)$$

Also,

$$Throughput = \frac{File\ Size}{Transmission\ Time}(bps) - - - - - - - - - - - (6)$$

$$Max.TCP\ Throughput = \frac{RCV\ Buffer\ Size}{Round\ Trip\ Time\ (RTT)} - - - - - - - - - - (7)$$

Therefore,

$$Throughput \leq \frac{RWIN\ (TCP\ Receive\ Window)}{RTT} - - - - - - - - - - (8)$$

Equations 4 to 8 are used to manually compute the Throughput value of a particular traffic or packet, be it TCP or UDP traffic pattern; this was done to verify the measured data. The maximum throughput which is sometimes regarded as the bandwidth is always less than the network performance throughput.

The reading for Throughput was recorded, transferred to Microsoft Excel spreadsheet program for more clarity as shown in table 1 below.

Table 1: Table of values for throughput against number of workstations in the three regions

From the above reading, the graph of Throughput was plotted against the number of work stations using Microsoft Excel; the graph is as shown in figure 6 below. The graph shows the effect of number of work station against network throughput.

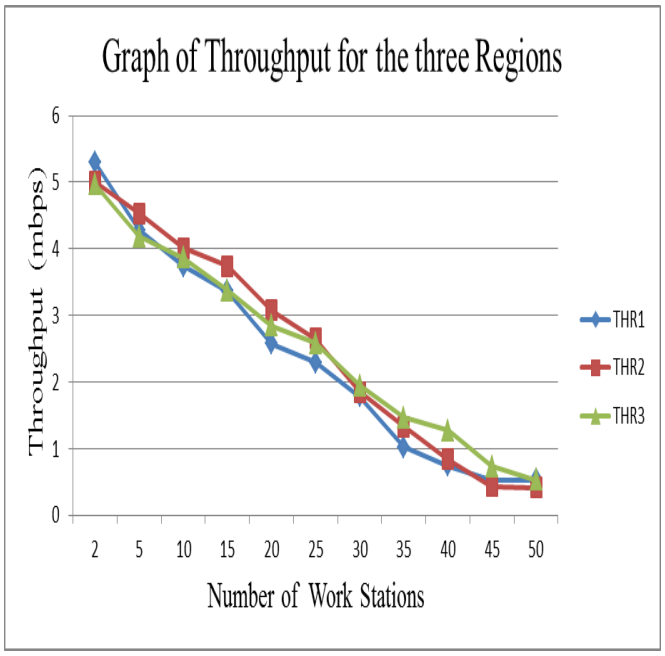


Figure 6: Graph of Throughput against Number of Work Stations in the three Regions

Delay reading was also collated and recorded, transferred to Microsoft Excel spreadsheet program for more clarity as shown in table 2 below.

Table 2: Table of Values for Mean Delay against Number of Workstations in the three Regions

From the above reading, the graph of Delay was plotted against the number of work stations using Microsoft Excel, the graph is as shown in figure 6 below. The graph shows the effect of number of work station against network Delay.

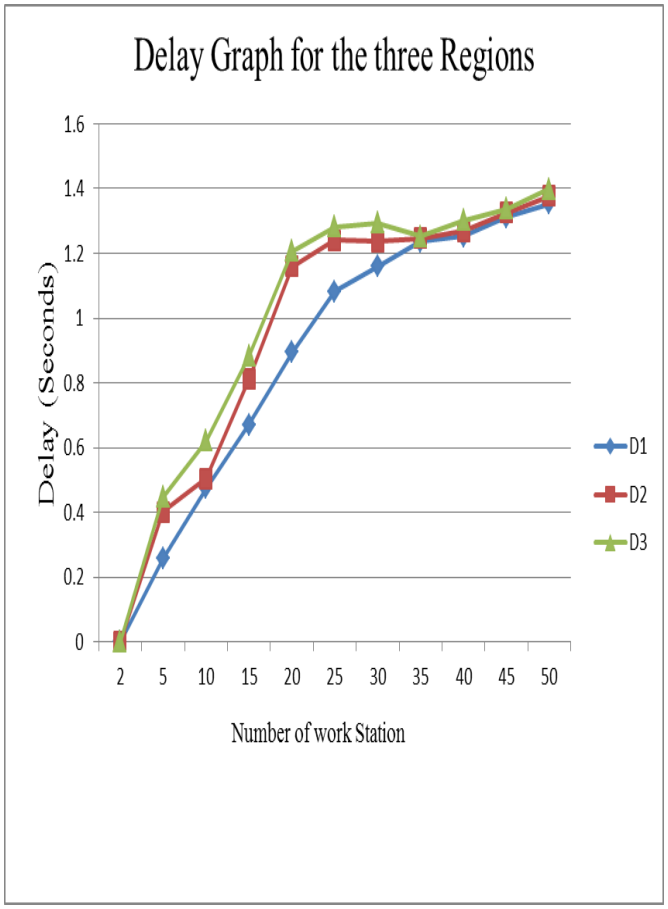


Figure 6: Graph of Mean Delay against Number of work stations in the three regions

4.0 Measurement Analysis

When we ran the software(s) on the network, the results reported a throughput of approximately 694,030 to 54,920 bytes- per-second. So, how does this compare with the theoretical maximum of 11 Mbps? 11 Mbps is equivalent to 1,441,792 bytes-per-second (11 x 1024 x 1024 bits / 8 bits / byte). So, is our actual throughput (which is roughly 48% to 3.8% of the theoretical) reasonable? The short answer -- yes. The 11 Mbps is the theoretical maximum of what the hardware and medium are capable of. There are numerous factors that can contribute to the difference -- including network traffic, interference from other wireless devices and, equally important, the overhead of the 802.11 and TCP protocols. The throughput we measure doesn't take into account that every 802.11 packet includes additional bytes besides the data payload.
Figure 6 however, shows that increase in the number of work station(s) will increase the mean delay of a wireless network.

5.0 Conclusion

The objective of this research is to carry out a real time performance monitoring of IEEE 802.11 compliant network, by investigating the effect of varying the number of work stations on the Quality of Service parameter on WLAN.
We have concluded that the throughput and delay performance strongly depends on the number of stations of the wireless network. Moreover, if the network is composed of a small number of stations and for small length frames, the network will have better performance.

In this thesis, however, if the number of workstations increases to the point of network saturation, both mean packet delay and throughput performance of the IEEE 802.11 protocol degrades significantly. Similarly, it was also shown that the mean packet delay of arrived packets increases as the number of workstations increases at saturation.
Therefore, to achieve a better enhanced network performance, it was observed that the IEEE 802.11b WLAN requires an improvement on its window back off algorithm and fairness.
More so, future work can include the throughput and delay analysis including the effect of hidden stations. Another possible area of research is a consideration of an erroneous wireless channel.

References

[1] Amjad, M. K., and Shami, A. (2006). Improving the Throughput Performance of IEEE 802.11 Distributed Coordination Function. 23rd Biennial Symposium on Communications, pp. 182 – 185.

[2] Chen, R., and Liu, X. (2010). Performance Evaluation over IEEE 802.11 Wireless. 2010 Second International Conference on MultiMedia and Information Technology: 2010 (2), pp. 31 – 34.

[3] https://code.google.com/p/xjperf/

[4] https://en.wikipedia.org/wiki/Wireshark

[5] http://nutsaboutnets.com/netstress/

[6] https://nutsaboutnets-documentation.s3.amazonaws.com/NetStress

[7] http://wndw.net/pdf/wndw3-en/ch16-network-monitoring.pdf

[8] https://www.wireshark.org

[9] Ivanov, S., Botvich, D., and Balasubramaniam, S. (2010). Joint Throughput and Packet Loss Probability Analysis of IEEE 802.11 Networks. 2010 IEEE symposium on Computers and Communications, pp. 673-676.

[10] Klenrock L. and F. Tobagi, "Packet Switching in Radio Channel: Part 1, Carrier Sense Multiple – Access Modes and their Throughput – Delay characteristics," IEEE Trans. Comm, Vol. 23, no. 12, Dec. 1975, Pp. 1400 – 16.

[11] www.firewall.cx/Downloads/Administrator Utilities

[12] Yang, J. W., Kwon J. K,. and Hwang, H. Y. (2009). Goodput Analysis of a WLAN with Hidden Nodes under a Non-Saturated Condition. IEEE Transactions on Wireless Communications, 8(5), pp. 2259 -2264.

[13] Zheng, Y., Lu, K., Wu, D., and Fang, Y. (2006). Performance Analysis of IEEE 802.11 DCF in Imperfect Channels. IEEE Transactions on Vehicular Technology, pp. 1648 - 1656.