



# Empirical Study of Achievable Bandwidth Capacity of VoIP Infrastructure in an Intranet with Open Source Tools

M. B. Soroyewun

School of Science and Technology,  
 National Open University of Nigeria,  
 Kaduna Study Center, Kaduna, Nigeria

A. A. Obiniyi

Department of Mathematics,  
 Ahmadu Bello University, Zaria,  
 Kaduna State, Nigeria

## ABSTRACT

This paper presents an empirical study of achievable bandwidth/concurrent calls capacity of an open source Voice over Internet Protocol (VoIP) Server deployed on an Intranet with open source tools. Network readiness in IPv4 and IPv6 environments was evaluated using Iperf while VoIP calls were simulated with open source SIPp (Session Initiation Protocol performance-tester). Voipmonitor, another open source software, was used to measure the bandwidth capacity and maximum concurrent calls handling capacity of the Asterisk Server over fast Ethernet Local Area Network. It can be deduced that the concurrent calls capacity of a VoIP server is dependent on a number of factors such as the VoIP readiness / bandwidth of the network and hardware specifications of the VoIP server. The result from experimental values shows that the achievable bandwidth otherwise called TCP throughput (for the 100Mbps LAN) for IPv4 and IPv6 traffics in dual stack configuration is 86.8Mbps and 87.4Mbps respectively. While that of UDP bandwidth measurement gives 50Mbps, jitter value of 1.794ms and packet loss of 0.59% for IPv4 traffic and 49.7Mbps bandwidth, 0.592ms jitter with packet loss of 0.22% for IPv6 traffic. At the maximum concurrent calls handling capacity, the processor of the VoIP server was observed to have 100% loading or utilization measure. It is recommended that the VoIP server should not be loaded continuously at this maximum processor load for optimum performance and longevity of life.

## General Terms

Networking, Performance Analysis, Voice over Internet Protocol

## Keywords

Achievable bandwidth, Throughput, Empirical Study, VoIP, FreePBX, Concurrent calls capacity

## 1. INTRODUCTION

Voice over Internet Protocol (VoIP) bandwidth consumption over a network is one of the most important factors to consider when deploying a VoIP infrastructure. Failure to account for VoIP bandwidth requirements will severely limit the reliability of the VoIP system and place a huge burden on the network infrastructure and VoIP Server in question [1].

In computer networks, bandwidth is often used to describe data transfer rate – the amount of data that can be transferred from one point to another in a given time (usually a second). More recently, the term has been popularly used to refer to the capacity of a digital communications line, typically measured in some number of bits per second. It is usually expressed in

bits (of data) per second (bps) and occasionally as bytes per second (Bps). Generally, a link with a high bandwidth is one that may be able to carry enough information to sustain the succession of images in a video presentation and high volume of audio packets or VoIP calls [2].

As in data networks, bandwidth congestion can cause packet loss and a host of other Quality of Service (QoS) problems. Hence, proper bandwidth reservation and allocation is essential to VoIP quality. Sharing data and voice on the same wires, which is one of the great attractions of VoIP, is also a potential challenge for implementers who must allocate the necessary bandwidth for both networks in a system normally designed for one [3].

## 2. BANDWIDTH AND THROUGHPUT

There is a clear difference between theoretical bandwidth or available bandwidth and throughput which is the achievable bandwidth of a link. According to [21 and 22], among the different metrics peculiar to traffic monitoring, one of the most significant is available bandwidth. From available literature, a single link is characterized by its capacity,  $C$ , defined as the maximum rate at which it can transfer data. Available bandwidth is usually related to the unused capacity of a link. Specifically, since at a given instant of time  $t$  a link is either transmitting at a rate equal to its capacity  $C$  or idle, the definition of instantaneous available bandwidth is meaningless, and a certain time-averaging is needed [21]. The available bandwidth of a network path, made of  $N$  links,  $L_1, \dots, L_N$ , is, invariably, defined as

$$A(t, \Delta t) = \min_{i=1, \dots, N} \frac{1}{\Delta t} \int_t^{t+\Delta t} C_i (1 - u_i(\tau)) d\tau, \dots (\text{eqn. 1})$$

where  $C_1, \dots, C_N$ , are the link capacities and  $u_1(t), \dots, u_N(t)$  are the percentages of link utilization. Estimating the available bandwidth of a network path represents an important task for many applications like VoIP whose performance and effectiveness can be increased by properly exploiting the output of such activity. For example, available bandwidth evaluation plays a fundamental role in traffic engineering algorithms. For such algorithms, the available bandwidth is one of the major parameters which determine the choice of the best connection path [23].

It must be noted that not only is the available bandwidth of the system affected by the introduction of security mechanisms, but in addition, the effective bandwidth of the VoIP system is significantly depreciated as shown in equation 2. Where  $N_d$  = number of packet received at destination,  $P_s$  = packet size in



unit of byte, and  $t$  = total time of simulation in unit of second. The unit of bandwidth is usually given in Mbps [24].

$$\text{Bandwidth} = \frac{N_d \times P_s \times 8}{t \times 1 \times 10^6} \quad (\text{eqn. 2})$$

On the other hand, Throughput is the rate at which bulk data transfer can be transmitted from one host to another over a sufficiently long time period. The throughput of a connection can be characterized by the following formula. The unit of throughput also is Mbps.

$$\text{Throughput} = \frac{N_d \times P_s \times 8}{(T_{sp} - T_{st}) \times 1 \times 10^6} \quad (\text{eqn. 3})$$

Where,

$N_d$  = number of packet received at destination,

$P_s$  = packet size in unit of byte,

$T_{sp}$  = stop time of each traffic flow in unit of second,

$T_{st}$  = start time of each traffic flow in unit of second.

Packet loss is the percentage of the number of packet that gets lost in a connection. Normally, when a device/path is overloaded and cannot accept any incoming data at a given time, then packet loss occurs. Heavy packet loss has a great impact on VoIP, more especially on video quality in either continuous or real-time perspective [24]. In this paper, an attempt is made to empirically evaluate the achievable bandwidth /concurrent call capacity of VoIP infrastructure. Section 3 presents the overview of VoIP protocols, section 3.1 discussed VoIP Servers and Services while section 4 considers empirical measurement. Related works, materials and method, experimental setup, results and discussion, conclusion and recommendation form the remaining part of the paper.

### 3. OVERVIEW OF VOIP PROTOCOLS AND SERVICES

Voice over IP also known as IP Telephony or broadband telephony allows the routing of voice conversations over IP-based networks including the Internet. It has created a world of possibilities for businesses to realize cost savings by utilizing their existing IP networks to carry voice, video and data; especially where such networks have underutilized capacity that can carry VoIP at no additional cost on their Local Area Networks or Intranet [4].

In order to provide useful services, VoIP requires a set of control protocols for connection, establishment, capabilities exchange and conference control. In recent time, there are two popular protocols that exist to meet this need. The first is ITUT-T (International Telecommunication Union, Telecommunication Standardization Sector) H.323 and the second is the IETF Session Initiation Protocol (SIP). Others are Media Gateway Control Protocol (MGCP) and ITUT-T H.248. These signaling protocols are used to set up and tear down calls while signaling packets carry information required to locate users, negotiate capabilities, create call establishment and set up call features [3].

SIP has been mainly adapted for recent developments [5] and will be the major focus of this work. SIP is reasonably modular; in the sense that it encompasses basic call signaling, user location and registration. Advanced signaling is part of SIP but within a single extension. QoS directory accesses, service discovery, session content description and conference

control all reside in separate protocols. This made it possible to use other protocols like H.245 capability description elements in SIP, with no changes to SIP at all. On the other hand, H.323 is less modular. It is defined by a vertically integrated protocol suite for a single application. This may be responsible for its lack of popularity in recent time [6].

#### 3.1 VoIP Servers and Services

Essentially, Asterisk is one of the software that is installed to turn any computer into a VoIP Server [7]. It provides many services like call conference, voice-mail, video call, Interactive Voice Response (IVR), Internet Protocol Private Branch eXchange (IP PBX), Music on Hold (MOH).

Asterisk PBX is a cost free and open source software that is bundled on Linux. It supports multiple protocols (SIP, SCCP-Signaling Connection Control Part, H.323, IAX2-Inter-Asterisk eXchange version 2 - and others). Asterisk allows both the use of software and hardware based phones [8].

The components of SIP server can be divided into two elements; user elements and network elements. The User Agent comes in two flavors – the User Agent Client (UAC) and the User Agent Server (UAS). A UAC sends SIP requests and receives SIP responses while a UAS receives SIP requests and sends SIP responses. The major network elements are SIP registrar, SIP Proxy, SIP Redirect Server and Presence Server [9].

#### 4. EMPIRICAL MEASUREMENT

The three techniques mostly employed in performance evaluation are analytical (or mathematical) modeling, simulation modeling and measurements [10]. The choice of the techniques used depends on a number of criteria. When the system under investigation already exists as when designing an improved version of a product, measurement technique can be used. On the other hand, when the system does not exist or is too difficult to deal with, a performance model must be developed. Modeling may be an abstract mathematical representation of the system behavior or an operational abstraction implemented as a simulator [11].

Measurement-based performance analyses have become an increasingly important component of network research. Measurements are carried out by using software and hardware monitoring devices. Hardware monitoring devices include probes and sensors, counters, as well as devices for data display and recording. In addition, software monitoring records certain events and information about the system state. This is otherwise called operational analysis which covers measurement and estimation of the real system. Analysis results can then be used for improving performance and setting up new bounds of the system [12]

Measurement is necessary for understanding current system behavior and how new systems will behave. Measurement is meaningless without careful analysis. Measurement and analysis enable models to be built which can be used to effectively develop and evaluate new techniques. It is important to determine what to measure before any measurements take place. Many tools have been developed to measure/monitor general characteristics of network performance. All network performance measurement tools can be categorized into two broad categories: (1) non-intrusive measurement or passive tools for example tcpdump and SNMP (Simple Network Management Protocol) tools and (2)



intrusive measurement or active tools like traceroute and ping [13, 14].

Non-intrusive or passive measurement measures the network behavior by observing the packet arrival rate at an end system and making some deduction on the state of the network, thereby deducing the network performance on the basis of these observations. In contrast, intrusive or active measurement refers to the controlled injection of packets into the network and the subsequent collection of packets [14]. In this work, open source tools like Voipmonitor and Iperf/Jperf (Internet Performance tester and its Java Graphical User Interface part) will be used for the measurements while SIPp (Session Initiation Protocol performance-tester) will generate the simulated calls.

Invariably, measured data need to be analyzed and the results should be properly interpreted. Analyzing such data in networks is usually done using statistical methods like averaging, dispersion (variability), t-test, ANOVA (Analysis of Variance), correlations, regression analysis, distributional analysis, frequency analysis, principal-component analysis and cluster analysis with necessary graphs and charts [13].

## 5. RELATED WORKS

A review of related literatures is presented in this section. In [24], Asterisk Server performance as a SIP Server as well as of bandwidth consumption in multiple scenarios was presented. The work made use of SIPpv3.1 and X-Lite v3.0 to evaluate the performance of Asterisk PBX v1.4.18 on Debian 4.0 on a test bed. The authors concluded that there is a non-linear relationship between the call rate and the processor load the Asterisk server with the given configuration.

In [25], the authors implemented some commonly used VoIP CODECs (Compressor/Decompressor) on Windows Operating Systems to evaluate their performance on IPv4 and IPv6. Using a test-bed setup, their results showed that there is only a slight difference in performance between TCP IPv4 and IPv6 networks while VoIP throughput, jitter and latency values are significantly different depending on the CODEC, OS and the IP version.

The authors in [21] looked at performance comparison of the techniques available for bandwidth measurement in IP networks. Their focus was on three different techniques using a test-bed with MGEN (Multi-Generator) as the traffic generator, IGI (Initial Gap Increasing), pathChirp and Pathload as measurement tools. Their conclusion showed the strengths and weaknesses of the three tools.

This paper used a different methodology to evaluate the achievable bandwidth and concurrent call capacity of FreePBX server in both IPv4 and IPv6 networks. In this work, open source and freeware tools – Voipmonitor, SIPpv3.3, X-Litev4.5 and Jperf – were used.

## 6. MATERIALS AND METHOD

For the experiments and practical deployment, relevant computer hardware and software were used. The hardware and software configurations of the computer used for FreePBX (VoIP) Server comprise of Intel Pentium Core 2 Duo CPU with 3.0 GHz processing speed, 2 GB RAM, 180 GB Hard drive, DVD/CD-ROM drive, Fast Ethernet Network Card and Audio Card installed with FreePBX 2.10.1.10 running on CentOS 6.2, Linux Kernel 2.6.32-220.13.1.el6.i686, Asterisk Ver. 10.11.0, Apache /2.2.15, PHP 5.3.3 and MySQL 5.1.61.

The hardware and software specifications for the computer on which Iperf/Jperf was installed is made up of Intel Pentium Core 2 Duo CPU with 3.0 GHz processing speed, 4 GB RAM, 150 GB Hard drive, DVD/CD-ROM drive, Realtek Fast Ethernet Network Card, Linksys 802.11b/g Wireless Network Card and Audio Card with Windows 7 Ultimate (32-bit) Operating System (OS). That of the laptop used consists of Intel Pentium Core i5 (4 CPUs) @ 2.67 GHz processing speed, 2 GB RAM, 280 GB Hard drive, DVD/CD-ROM Drive, High Definition Audio Controller, Intel(R) 82577LC Gigabit Network Card, Broadcom 4313 802.11b/g/n Wireless Network Card with Windows 7 Professional 32-bit (6.1, Build 7600) OS. All the machines are connected to the Intranet through Cisco Switches (Catalyst 2960 series).

The open source and free software tools used are Voipmonitor version 6.1301, SIPpv3.3 (Session Initiation Protocol-performance tester) and Iperf/Jperf 2.0.2. X-Lite softphone 4.5, a proprietary freeware product was used to receive the simulated calls from SIPp.

## 6.1 Installations and Configurations

In this section, the installations and configurations of systems, software and tools used in this work were discussed in detail.

### 6.1.1 Installing and Configuring FreePBX (VoIP Server)

FreePBX is an open source VoIP Phone system packaged in a full-featured PBX web application that controls and manages Asterisk (PBX). It is built on the LAMP Stack (Linux, Apache, MySQL, PHP and Asterisk). It is a modular system with click-to-install plugins downloadable over the Internet from the online module repository [15, 16]

There are two ways to install FreePBX. It can be installed as a package on an existing Linux distribution or installed as a FreePBX distribution on old or new hard drive. The hard way to install FreePBX is to set-up a machine running Linux, Asterisk and all other required applications before installing FreePBX manually. The easy way is to download and install the FreePBX Official Distro from their website and install on a dedicated machine.

After the FreePBXDistro has been successfully installed, the first steps after installation are:

- (1) Login to the PBX Graphical User Interface (GUI) from another machine in the same network.
- (2) Set a static IP Address for the IP PBX Server if you have not done so and configure DNS.
- (3) Check for FreePBX updates.
- (4) Configure Asterisk SIP Settings.
- (5) Configure E-mail Settings (Email Server and E-mail Alert Destination Address).
- (6) Explore the Modules. One of the important modules to explore and configure is the Extensions Module where one can setup the extensions that will be used on the IP PBX Server. In this module, different extension numbers with their passwords and voice-mails (if desired) are set up.





- (7) Then configure the phones – either softphones like X-lite, Linphone, Ekiga or IP Phones like Cisco and Avaya.
- (8) Configure Backups

In this scenario, the extension lines for the softphones were configured by using the bulk extension template from the Bulk Extensions module. About 150 SIP extension numbers were configured on the FreePBX Server. X-lite softphone is used for simplicity of configuration and operations.

### 6.1.2 Installing and Configuring Voipmonitor

Voipmonitor is an open source live network packet sniffer VoIP monitoring tool for SIP and RTP (Real-Time Transport Protocol) VoIP protocols running on Linux. As at the time of this report, it has not supported IPv6. It was used only to monitor the bandwidth and concurrent calls capacity of the IP PBX Server deployed on the Intranet. It was installed on the FreePBX Server running on CentOS 6.2 Linux OS by following the step by step guide available at [18].

To install VoIPmonitor, WEB GUI and License key file must first be downloaded (even for the 30 days free trial version). For complete and successful installation it is important to read the sniffer manual and WEB GUI manual and follow the step by step installation guide both available at <http://voipmonitor.org/doc>. There is need to select and follow the step by step installation guide for the specific Unix / Linux variant. VoIPmonitor supports Debian, Ubuntu, CentOS and FreeBSD. The knowledge of the Linux Distro and the version or nearest version will aid a stress free installation process. After following the step by step guide, from another computer web browser type <http://yourserveripaddress/> and finish the installation. In this case, the address used is <http://10.1.24.11/voipmonitor>. (The license key is needed and obtainable by registering at <https://www.voipmonitor.org/whmcs/register.php> for the specific version downloaded. The license key obtained is copied in to the voipmonitor directory).

By login in to the voipmonitor from the Server IP address/voipmonitor, some important features from Settings were configured.

### 6.1.3 Installing and Configuring SIPp

SIPp is a performance testing tool for the SIP protocol. It includes a few basic SipStone user agent scenarios (UAC and UAS) and establishes / releases multiple calls with INVITE and BYE methods. It can also read XML scenario files that describe any performance testing configuration. SIPp can be used to test many real SIP infrastructures like SIP proxies, B2BUAs (Back-to-Back User Agents), SIP Media servers, SIP Gateways and SIP PBXes like FreePBX. It is also very useful to emulate thousands of user agents calling the SIP server [20]

Installing SIPp is pretty straight forward. Like many other open source projects, there are two versions of SIPp: a stable release and unstable (or BETA) release. It is always advisable to install the stable release. Information abounds on the Internet on how to install SIPp on different platforms (Cygwin-windows and Linux distributions). It was installed on Linux for this work.

The focus here is to actually test the concurrent calls capacity of the FreePBX server. This was achieved by running the SIPp performance testing tool to send SIP requests to X-Lite

softphone through the FreePBX server. Concurrent calls statistics was available in the Tools Menu. Other menus apart from Settings and Tools are CDR (Call Detail Record), Messages, Charts, Active Calls, Live Sniffer, Dash board, Users and Audit, Capture Rules, Alerts (Anti-Fraud), Reports, Call Summary and Report generator, Upgrade and Logout. Essentially, QoS Report can be accessed from the Report menu.

### 6.1.4 Installing and Configuring Iperf/Jperf

Iperf 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 quality of a link can be tested as follows:

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

Essentially, the difference between TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) is that TCP uses processes to check that the packets are correctly sent to the receiver whereas with UDP the packets are sent without any checks but with the advantage of being quicker than TCP. Iperf uses the different capacities of TCP and UDP to provide statistics about network links. Finally, Iperf can be installed very easily on any UNIX/Linux or Microsoft Windows system. One machine must be set as client, the other one as server. To install Jperf on Windows 7 or Linux, Java and Iperf must have already been installed on the machine. [19].

## 7. EXPERIMENTAL SETUP

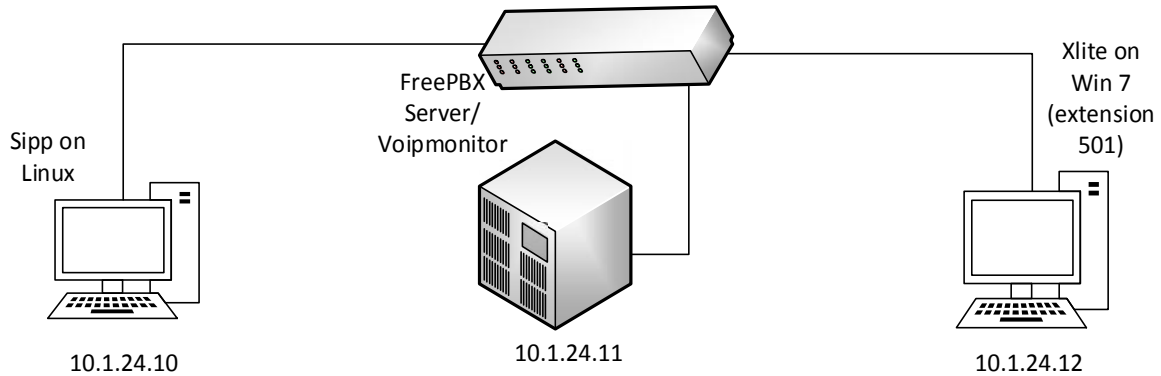
In this setup, the VoIP (FreePBX) Server that has already been installed was configured with about 150 user accounts or SIP extension numbers. Also the voipmonitor software was installed and configured with the trial version license on the FreePBX Server. At this stage the FreePBX was already connected to the network for Internet connection in order to download relevant packages and updates. The SIPp traffic simulator (or generator) was installed on another Linux machine running CentOS 6.2 while X-lite softphone was installed on Windows 7 machine and configured with a SIP user account.

The SIPp was then used to generate continuous calls to the VoIP Server using the command line:

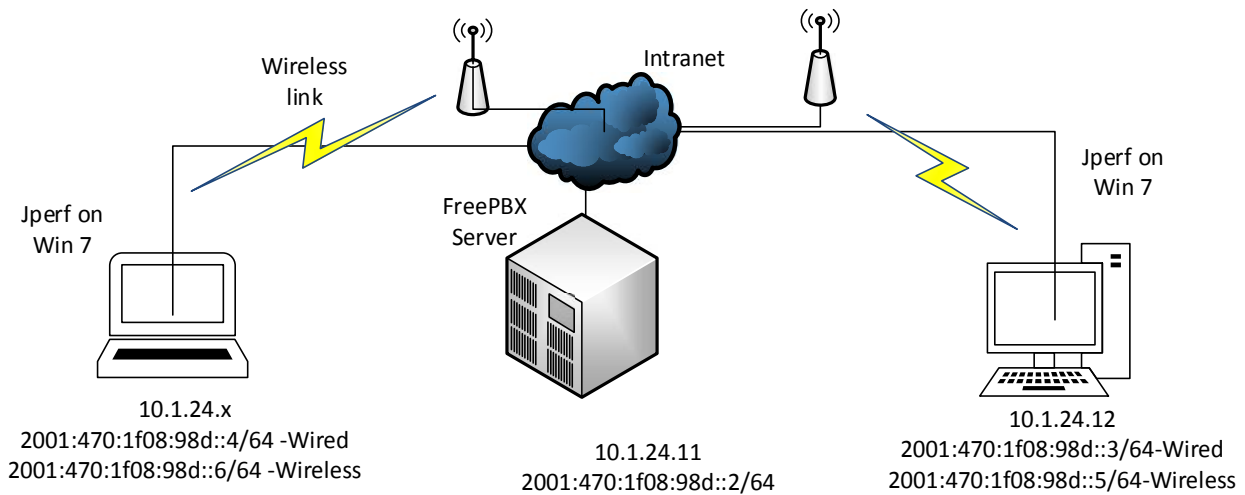
```
$ ./sipp -sn uac -d 10000 -s 601 10.1.24.11 -l 500
```

This command will execute 500 concurrent calls (with the -l 500 parameter) with each call lasting 10s (the -d parameter is in ms) to extension 601(sip user account) on the VoIP Server with the default voice codec G.711ulaw. Figures 1 and 2 illustrate the practical setup.

The voipmonitor on the FreePBX recorded the maximum number of concurrent calls on the three occasions. Iperf/Jperf was used to determine the VoIP readiness and measure the effective or available bandwidth and the throughput on the network for both IPv4 and IPv6.



**Figure 1: Experimental setup for bandwidth requirement of VoIP Server using SIPp/Voipmonitor**



**Figure 2: Experimental setup for bandwidth requirement of VoIP Server using Jperf**

## 8. RESULTS AND DISCUSSION

The VoIP readiness of the network infrastructure was determined by using Iperf/Jperf which gave the following outputs as shown in figures 3a to 5b.

The theoretical bandwidth of the fast Ethernet network of the Intranet under consideration is 100Mbps while that of the 802.11g wireless standard used in the Intranet is 54Mbps. The result from experimental values shows that the average effective/achievable bandwidth otherwise called TCP throughput (for the 100Mbps LAN) for IPv4 and IPv6 traffics in dual stack configuration is 86.8Mbps and 87.4Mbps respectively. While that of UDP bandwidth measurement gives 50Mbps, jitter value of 1.794ms and packet loss of 0.59% for IPv4 traffic and 49.7Mbps bandwidth, 0.592ms jitter with packet loss of 0.22% for IPv6 traffic. It can be observed that the TCP throughput and UDP bandwidth value for IPv4 and IPv6 are nearly the same, while jitter value of IPv4 (1.794ms) is approximately twice the value of that of IPv6 (0.592ms). Similarly, the packet loss

for IPv4 traffic (0.59%) doubles that of IPv6 traffic (0.22%). From the aforementioned, it can be deduced that the jitter and packet loss statistics are within the permissible limit of 5ms and 1% respectively. However, a paired samples t-Test in Excel 2010 conducted for IPv4 and IPv6 wired TCP Throughput reveals that the difference in the mean is significant with the two-tail p-value < 0.05 (0.032). While for the UDP Bandwidth the test shows that there is no significant difference between the mean value of IPv4 and IPv6 wired at two-tail p-value > 0.05 (0.246) as can be seen in table 1. Conversely, the test shows that for wireless there is no significant difference between the mean values of both TCP Throughput and UDP Bandwidth of IPv4 and IPv6 respectively as shown in table 2.

The achievable bandwidth capacity of the VoIP Server shown by the number of concurrent calls it can handle at a given time was determined by using SIPp and Voipmonitor. The outputs from the experimental setup and configuration are shown in figure 6.

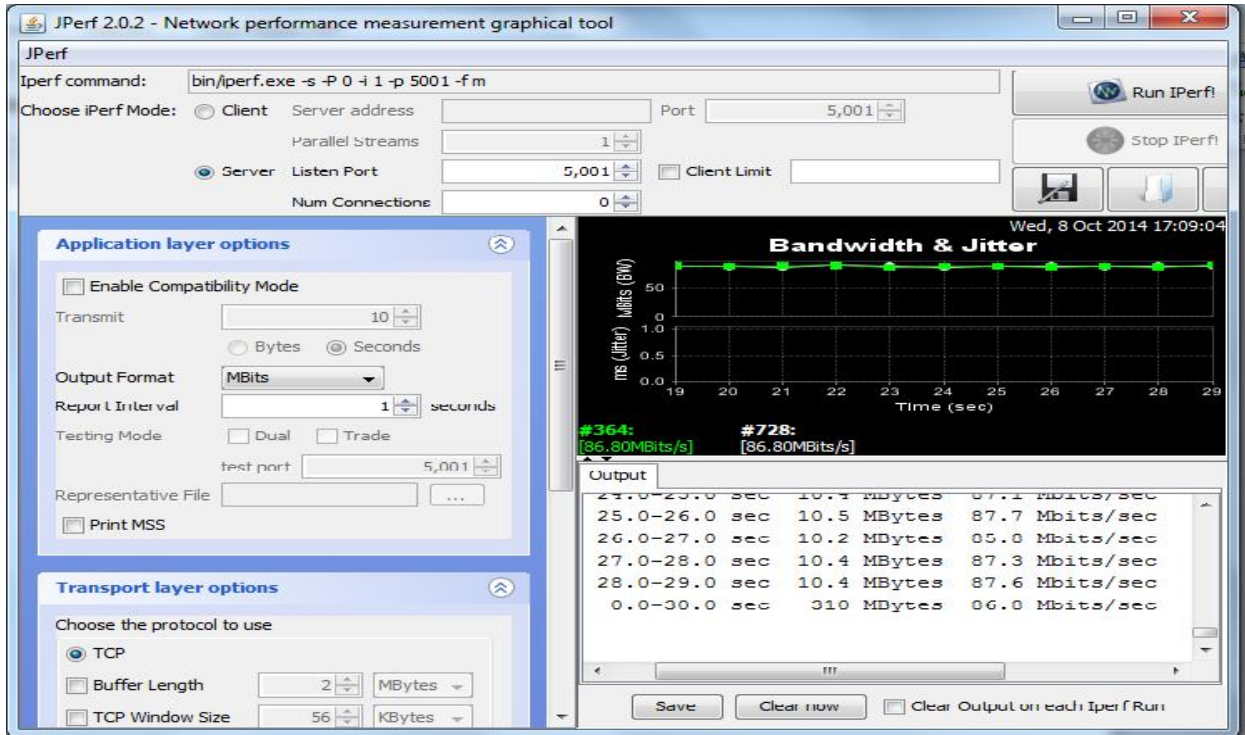


Figure 3a: TCP Throughput measurement with IPv4 server side

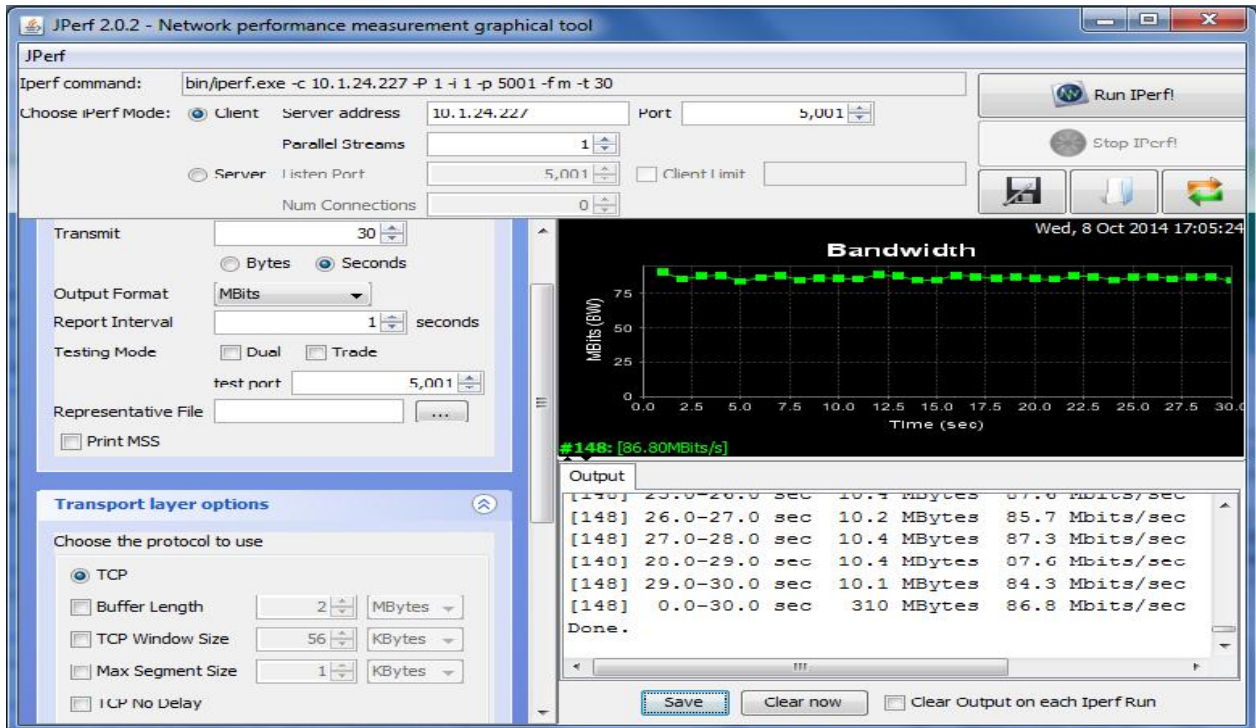


Figure 3b: TCP Throughput measurement with IPv4 client side



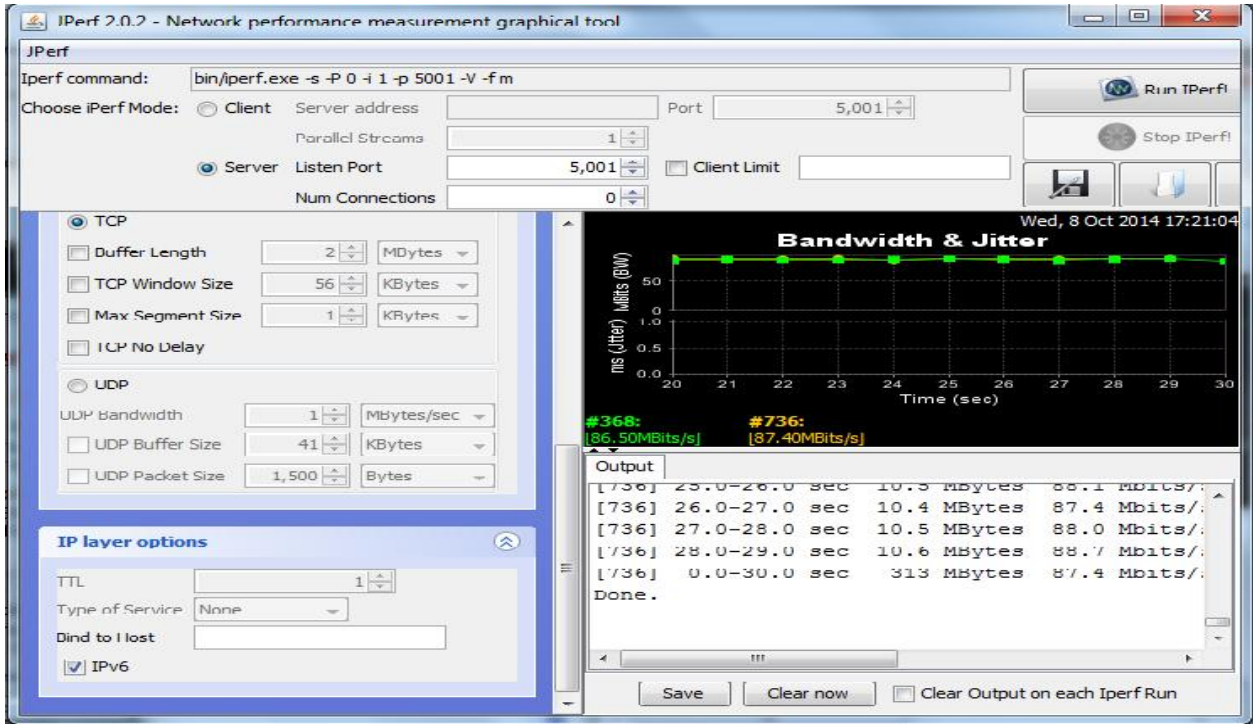


Figure 4a: TCP Throughput measurement with IPv6 server side

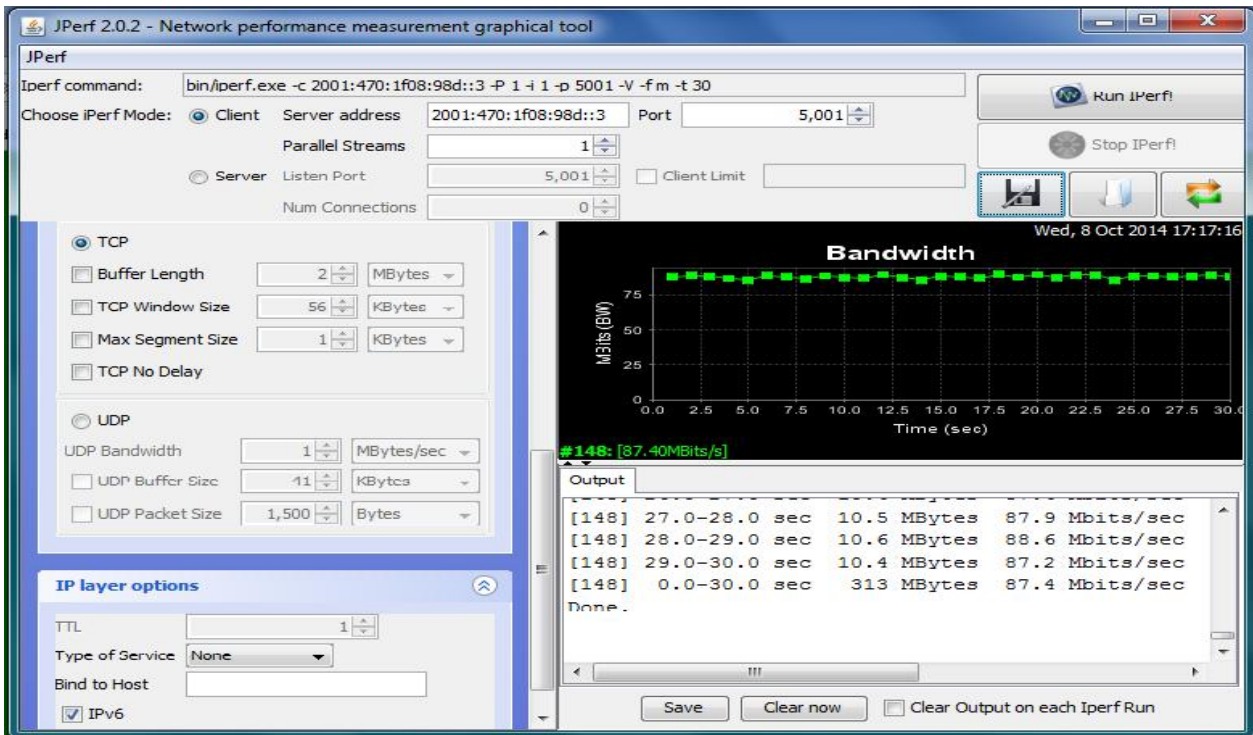


Figure 4b: TCP Throughput with IPv6 client side

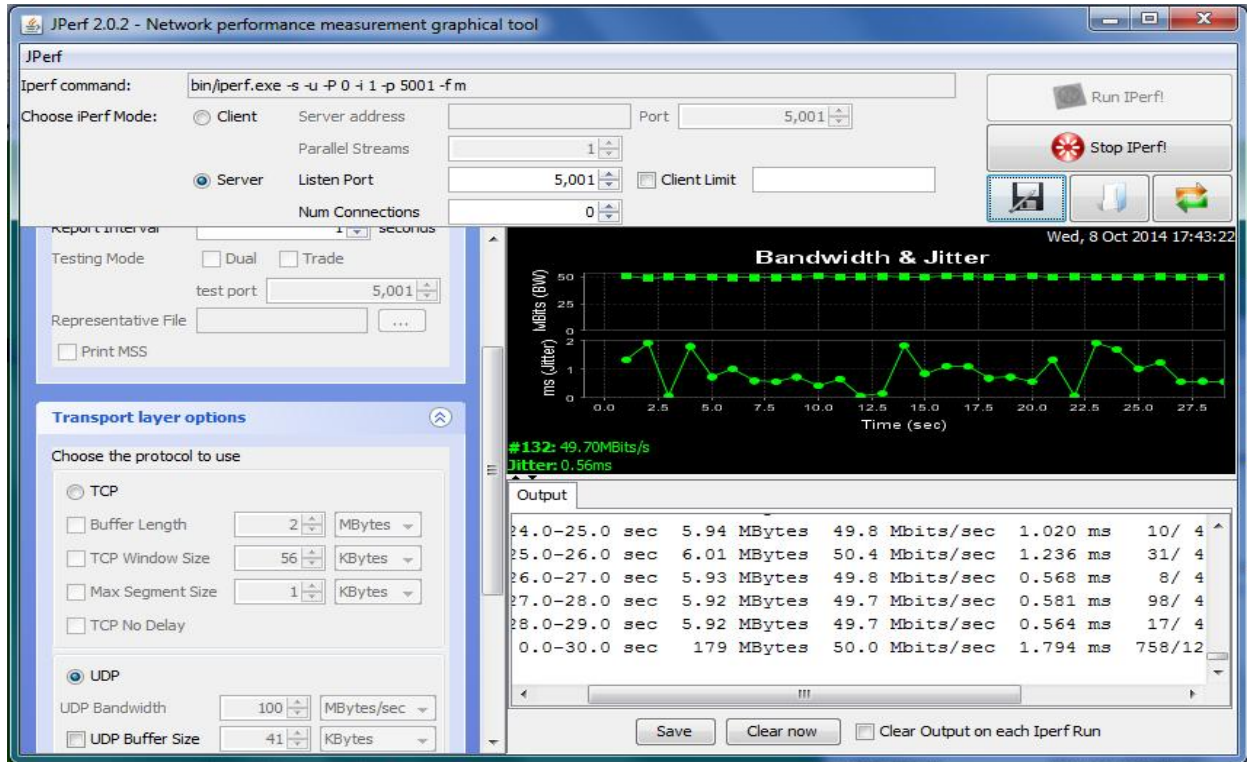


Figure 5a: UDP Bandwidth measurement with IPv4

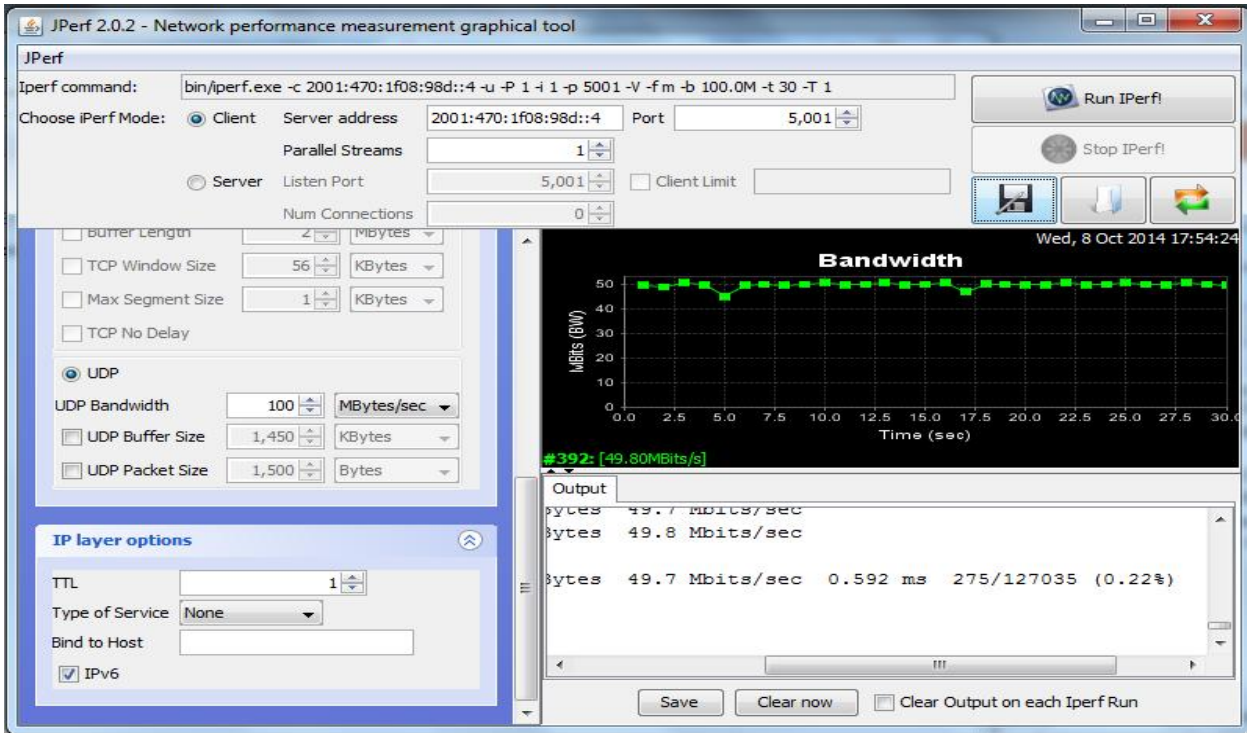


Figure 5b: UDP Bandwidth measurement with IPv6



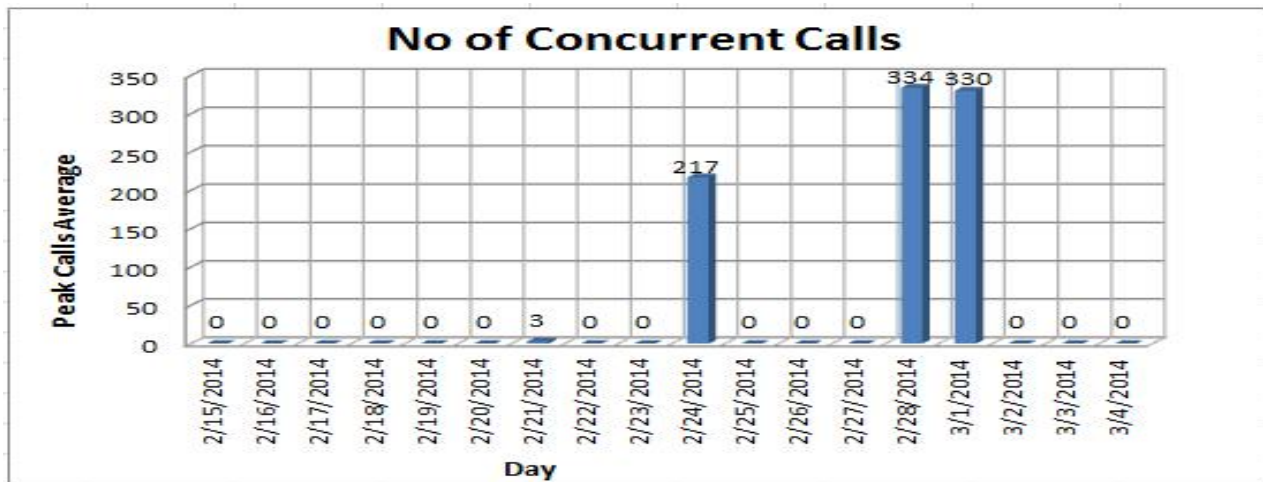


**Table 1. Paired Samples t-Test for difference in IPv4 and IPv6 wired TCP and UDP means**

t-Test: Paired Two Sample for Means for Wired Network					
		TCP Throughput		UDP Bandwidth	
		IPv4 TCP	IPv6 TCP	IPv4 UDP	IPv6 UDP
Mean		86.77	87.38	49.97	49.7
Variance		2.374586207	1.176137931	0.275197044	1.331428571
Observations		30	30	29	29
Pearson Correlation		0.402400077		0.123902607	
Hypothesized Difference	Mean	0		0	
Df		29		28	
t Stat		-2.249631942		1.18472555	
P(T<=t) one-tail		0.016116311		0.123042419	
t Critical one-tail		1.699127027		1.701130934	
P(T<=t) two-tail		0.032232622		0.246084838	
t Critical two-tail		2.045229642		2.048407142	

**Table 2. Paired Samples t-Test for difference in IPv4 and IPv6 wireless TCP and UDP means**

t-Test: Paired Two Sample for Means for Wireless Network					
		TCP Throughput		UDP Bandwidth	
		IPv4 TCP	IPv6 TCP	IPv4 UDP	IPv6 UDP
Mean		3.59	4.048	1.502	2.961
Variance		2.439882299	0.88751092	0.465385747	1.229501609
Observations		30	30	30	30
Pearson Correlation		0.060744198		-0.07309736	
Hypothesized Difference	Mean	0		0	
Df		29		29	
t Stat		-1.412699814		-5.94594741	
P(T<=t) one-tail		0.084193458		9.23796E-07	
t Critical one-tail		1.699127027		1.699127027	
P(T<=t) two-tail		0.168386916		1.84759E-06	
t Critical two-tail		2.045229642		2.045229642	



**Figure 6: Average number of concurrent calls per day during the period of 23-02-2014 to 01-03-2014**



From figure 6, it can be seen that during the given period (February to march), the maximum number of calls is about 668 while the minimum is 434. However, the highest average number of concurrent calls is 334 which translate into two simultaneous channels giving rise to one established call (i.e.  $334 \times 2 = 668$ , total calls). It was observed from the VoIP server (freepbx) system status interface that CPU utilization statistics reached 100% with red alert. It is deduced that operating the server continuously at this maximum will result in poor quality of calls and endanger the system health. It is therefore strongly advised not to operate the VoIP server continuously at this maximum to ensure longevity and optimum performance of the system. A load average of 50% of the maximum is recommended at any given time. This will result in about 167 concurrent calls ( $334/2$ ) which will be adequate for the total number of concurrent users of the intranet put at about 150 so that the total calls will not exceed 300. Hence, it can be deduced that the VoIP server bandwidth which is a function of the hardware and software specifications is adequate for the purpose it is to be deployed.

This result agrees with the measurement results using Jperf from figures 3a to 5b. Given that the default voice codec is G.711 which uses 64Kbps of the available bandwidth with some overhead load leading to a real bandwidth usage of 87.2Kbps [26]. If this bandwidth is used in both directions for a concurrent call, then  $87.2\text{Kbps} \times 2$  will equal 174.4Kbps. Taking IPv4 scenario with average TCP throughput of 86.8Mbps and UDP bandwidth of 50Mbps, hence  $(86.8 \times 1024)\text{Kbps}/174.4\text{Kbps} = 509.6$  concurrent calls for TCP.

Also  $(50 \times 1024)\text{Kbps}/174.4\text{Kbps} = 294$  concurrent calls for UDP.

Focusing on UDP for VoIP calls, half of 294 will give approximately 147 concurrent calls which is much closer to 150 concurrent calls arrived at from empirical study with the total calls not exceeding 300.

## 9. CONCLUSION AND RECOMMENDATION

This paper presents an empirical study of bandwidth capacity measurement of an open source VoIP Server deployed in an Intranet. The VoIP Server under consideration is a FreePBX Asterisk Server installed on a workstation in an Intranet setting. The Intranet is made up of Windows Server 2010 as the Domain Controller, CBT (Computer-Based Test) Server, this FreePBX (VoIP) Server and about 300 systems comprising of desktops and laptops as at the time of this study. Open source network monitoring software – voipmonitor and free software-Iperf/Jperf were used to measure the bandwidth capacity and maximum concurrent calls handling capacity of the FreePBX Asterisk Server over fast Ethernet LAN (100MB) with the default G711 codec. The VoIP calls were simulated with SIPp an open source VoIP server performance and stress /load tester software. It can be deduced that the concurrent calls capacity of a VoIP server is dependent on a number of factors such as the VoIP readiness/bandwidth of the network and system configuration of the VoIP server (CPU, memory and disk storage capacity). At the maximum concurrent calls handling capacity, the CPU of the VoIP server was observed to have 100 percent loading or utilization measure. Hence, it is recommended that the VoIP server should not be loaded continuously at this maximum for optimum performance and longevity of life.

## 10. ACKNOWLEDGMENTS

The authors thank the management of Institute of Computing and ICT, Ahmadu Bello University, Zaria, Kaduna State, Nigeria for permission to use the computer hardware for this work.

## 11. REFERENCES

- [1] TechTarget, <http://searchunifiedcommunications.techtarget.com/tip/Whats-in-a-VoIP-packet-An-introduction-to-bandwidth-for-VoIP> retrieved on 27/10/2014
- [2] Flickenger, R *et.al* (2006), How to Accelerate your Internet retrieved from <http://bwmo.net/bwmo-ebook.pdf> on 27/10/2014
- [3] Kuhn, D. R., Walsh, T. J. and Fries, S. (2005), Security Considerations for Voice Over IP Systems, NIST Special Publications 800-58, retrieved from <http://csrc.nist.gov/publications/nistpubs/800-58/SP800-58-final.pdf> on 4/6/2014
- [4] Naoum, R. S. and Maswady, M. (2012), Performance Evaluation for VOIP over IP and MPLS, World of Computer Science and Information Technology Journal (WCSIT) ISSN: 2221-0741 Vol. 2, No. 3, pg. 110-114, 2012
- [5] VoIP Overview and Infrastructure, <http://what-when-how.com/voip/voip-overview-and-infrastructure> retrieved on 27/10/2014
- [6] Schulzrinne, H. and Rosenberg, J. (1998), A comparison of SIP and H.323 for Internet Telephony retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.850> on 27/10/2014
- [7] Asterisk, <http://www.asterisk.org/index.html> retrieved on 27/10/2014
- [8] Luís Sousa, Performance Evaluation of the Asterisk PBX retrieved from [https://fenix.tecnico.ulisboa.pt/downloadFile/395137829643/resumo\\_avaliacao\\_asterisk\\_52871.pdf](https://fenix.tecnico.ulisboa.pt/downloadFile/395137829643/resumo_avaliacao_asterisk_52871.pdf) on 27/10/2014
- [9] Prokop, A. (2013), SIP Servers and Services, retrieved from <http://andrewjprokop.wordpress.com/2013/06/21/sip-servers-and-services/> on 27/10/2014
- [10] Jain R., (1991), Art of Computer Systems Performance Analysis Techniques for Experimental Design Measurements Simulation and Modeling. John Wiley & Sons, 1991.
- [11] Puigjaner, R. (2003), Performance Modeling of Computer Networks, IFIP/ACM Latin America Networking Conference, October 3 - 5, 2003, La Paz, Bolivia, [http://staffweb.cms.gre.ac.uk/~pa10/ResMeth/network\\_performance\\_modelling.pdf](http://staffweb.cms.gre.ac.uk/~pa10/ResMeth/network_performance_modelling.pdf) retrieved on 16th January, 2013
- [12] Martinovic, G., Petrisevac, B. and Zagar, D. (2010), Monitoring and measurement of computer network performance, Technical Gazette 17, 3, pp. 317-326
- [13] Dept. of Computer Science, University of Wisconsin-Madison Lecture Note,



- <http://www.cs.wisc.edu/~pb/640/perform.ppt> retrieved on 27/10/2014
- [14] Fu, H., Lin, L. and Jia, W., Efficient Wireless Link Bandwidth Detection for IEEE 802.11 Networks, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.151.6587&rep=rep1&type=pdf> retrieved on 27/10/2014
- [15] FreePBX, <http://schmoozecom.com/video/freepbx-overview.php> retrieved on 27/10/2014
- [16] Voip-info Wiki, <http://www.voip-info.org/wiki/view/freePBX> retrieved on 27/10/2014
- [17] FreePBX distro download, <http://www.freepbx.org/freepbx-distro> retrieved on 27/10/2014
- [18] Voipmonitor Centos 6.3 Fast Installation, [http://www.voipmonitor.org/doc/Centos\\_6.3](http://www.voipmonitor.org/doc/Centos_6.3) retrieved on 27/10/2014
- [19] Reeves, S., TechRepublic, Linux and Open Source, <http://www.techrepublic.com/blog/linux-and-open-source/using-jperf-to-check-network-performance> retrieved on 27/10/2014
- [20] SIPp Documentation, <http://sipp.sourceforge.net/doc/reference.html#Installing+SIPp> retrieved on 27/10/2014
- [21] Angrisani, L., D'Antonio, S., Esposito, M. and Vadursi, M. (2006), Techniques for available bandwidth measurement in IP networks: A performance comparison, *International Journal of Computer and Telecommunications Networking*, Vol. 50, Issue 3, Elsevier B.V., pages 332-349 available online at <http://www.sciencedirect.com>
- [22] Ahmed, H. A. S., Sulaiman, N., Mohammed, M. N., Performance Analysis of VoIP Quality of Service in IPv4 and IPv6 environment, retrieved from [www.aicit.org/JDCTA/pp1/JDCTA3640PPL.pdf](http://www.aicit.org/JDCTA/pp1/JDCTA3640PPL.pdf) on 24/10/2014
- [23] Kodialam, M., Lakshman, T.V. (2003), Dynamic routing of restorable bandwidth-guaranteed tunnels using aggregated network resource usage information, *IEEE/ACM Transactions on Networking* 11 (3) (2003) 399–410.
- [24] Razvan, R. and Cristian, I. (2008), A Practical Analysis of Asterisk SIP Server Performance, 7th International Conference RoEduNet 2008 in Cluj-Napoca, Romania, retrieved from <http://conference.cluj.roedu.net/papers/25.pdf> on 27/10/2014
- [25] Narayan, S., Gordon, M., Branks, C. and Li, F. (2010) "VoIP Network Performance Evaluation of Operating Systems With Ipv4 And Ipv6 Network Implementations". In: 3rd IEEE International Conference on Computer Science and Information Technology (Iccsit), pp. 669-673, 2010.