# Performance Testing, and Evaluation for the Voipv6 Network Related Functions, (Sendto and Receivefrom)

## (An Applied Networking Research Approach)

Mr. Asaad Abdallah Yousif Malik Abusin
Faculty of Computing and Informatics Multimedia University
Cyberjaya, Malaysia

Dr. Junaidi Abdullah
Faculty of Computing and Informatics
Multimedia University, Dr
Cyberjaya, Malaysia

Dr Tan Saw Chin
Faculty of Computing and Informatics Multimedia University, Dr
Cyberjaya, Malaysia

***Abstract*—(The network related functions (Sendto, and Receivefrom) in VoIPv6, are needed to obtain the communication socket in both UDP, and TCP before the communication can take place between the sending and receiving ends. The intent of testing and evaluating the network related functions in Voice over Internet Protocol (VoIPv6) in this research work is not to provide a comprehensive benchmark, but rather to test how well TCP (Transmission Control Protocol), and UDP (User Datagram Protocol) perform in sending and receiving VoIPv6 traffic and bulk data transfer, Part of this, due to the cumulative nature of VoIPv6 performance, can be achieved by testing the network related functions which are the Sendto and Receivefrom socket calls. This is because the sending concept in UDP IPv6 is based on the best effort sending of packets not a guaranteed sending as in TCP IPv6. In this context, performance enhancement techniques are needed to be applied in VoIPv6 due to the fact that there is no dedicated line between the sending and receiving ends. This is actually the putty and the drawback at the same time of VoIP. This is also the reason for IPv6 to take longer time yet to reach its full maturity (Recommendation G.711 of the ITU expectation is by the year 2050) when fully deploying real time applications due to their time sensitivity.**

***Keywords*—(VoIPv6 (Voice over Internet Protocol V6) Performance; Voice over Internet Protocol V6 Performance testing; Voice over Internet Protocol V6 Performance analysis; VoIPv6 quality testing in the protocol and application layers; Internet Measurement Research; and VoIPv6 performance improvements***

## PROBLEM STATEMENT

The problem statement in this research work in a précised way is that obtaining the Sendto and Receivefrom Socket calls in VoIPv6 contribute to both UDP, and the round trip delay as in TCP, and to the overall delay in VoIPv6 traffic, which is of a cumulative nature. Before any communication session could be established between any (TCP or UDP) client and server, the communication socket must be obtained first then only the sending and receiving of VoIPv6 packets can take place. This research work objective clearly here is to obtain and quantify this amount of delay so that improvement methods can be suggested and applied accordingly based on this. Additionally also to specify the contribution of the network related functions delay amount in the overall less possible delay limit in VoIPv6. This overall less possible amount of delay was stated by the International Telecommunication Union (ITU) and Internet Engineering Task Fours (IETF), as in the G.114 ITU recommendation.

G.114 is the International Telecommunication Union (ITU) standard recommendation that addressed acceptable delays for voice applications. It is oriented to national telecommunications and is more stringent than what is normally applied in private voice networks. One of the things it addressed is voice delay. The G.114 recommendation of the International Telecommunication Union (ITU), published in the year 1994, stated that < 150 milliseconds delay in voice is acceptable, > 400 milliseconds is unacceptable and in between people should be made aware of quality issue. (Habibullah Jamal, Kiran Sultan, 2008) suggested VoIPv6 improvement in the protocol level, as concurrently in this research work, specifying the contribution of the sendto and recevefrom socket calls is a step before any improvement can be made.

## I. INTRODUCTION

In this paper the time needed to obtain the communication socket in VoIPv6 was quantified and studied. The G.711 was the codec technique in the VoIPv6 software that was used in both the VoIPv6 client and server couples in both the source IPv6 and destination IPv6 as (W Richard Stevens *1998*) discussed voice transmission under IPv6 in detail. This research work added codes that were developed using socket programming FreeBSD to run the real time tests. The aim here is to run performance tests for VoIPv6 network related functions that are frequently used in both the TCP and UDP client and server programs or codes, functions like socket applications (Client/Server) with combination of functions such as Sendto and Receivefrom, Client/Server codes were developed as in (*Asaad A. Abusin, M. D. Jahangir Alam, Junaidi Abdullah, 2012).*

## II. RESEARCH MOTIVATION

This research work motivation from literature review, can be clearly described and summarized as, there is a need to quantify, and analyze the delay in VoIPv6 caused by different factors contributing to the overall delay as highlighted also in

(*Mark Allman of the* Internet Measurement Research Group (IMRG) Australian National University, *06 August 2012*), one of these factors is the time needed to obtain the communication socket.

## III. Literature Review

In (W Richard Stevens *1998*) UDP was defined as a transmission protocol in IPv6 applications, since it simply prepares its 8-bytes header and passes the Datagrams to IP, IP prepares its header, determines the outgoing interface by performing the routing functions, and then either adds the datagram to the Datalink output queue or fragment the datagram and adds each fragment to the Datalink output queue. UDP, as part of IPv6, was chosen to be the test environment focusing on testing and performance evaluation of the network related functions in VoIPv6 traffic. (*Asaad A. Abusin, M. D. Jahangir Alam, Junaidi Abdullah, 2012)* focused on testing and quality measurement and analysis of VoIPv6 performance. A client, server codes were developed using Free BSD, as a step before analyzing the Architectures of VoIPv6 in the current internet in order for it to cope with IPv6 traffic transmission requirements in general and specifically voice traffic, this is being attracting the efforts of research bodes currently.

According to (Kaushik Das, 2008), the quality of a voice-call can degrade significantly if VoIP voice packets are lost or delayed at any point in the network between VoIP users, and users can also notice this quality degradation more in highly congested networks or over long distances. In order to address this quality issue a set of service requirements to deliver performance guarantee while transporting voice traffic over the network are needed. A fundamental fact is that the delay in VoIPv6 is an accumulation of many factors one of which is the time needed to obtain the communication socket, and this is exactly the focus in this paper. *(Krishna P. Gummadi, Stefan Sarotu, W.A. Steven D. Gribble, Seattle, and W.A, King, 2002)* investigated the ability to estimate network latencies between arbitrary internet end hosts, such ability would enable new measurement studies and applications, such as investigating routing path inefficiencies on a wide-scale or constructing topologically sensitive overlay networks.

(*Mark Allman of the* Internet Measurement Research Group (IMRG) Australia, *06 August 2012*) highlighted a considerable network measurement research work being conducted within the Internet community, both in standards bodies (such as, IETF IPPM WG) and in various research labs. *(Geoff Huston of Telstra, GEOFF HUSTON Holds March 2003*) razed a big question that is, how *good* is your network? Or how can the quality of the service that specific network is offering can be measured and quantified? And in this context this research work is an attempt contributing to answering this question which has been lurking behind many IP vendors for many years. With the increasing levels of deployment of various forms of high-speed (or broadband) services within today's Internet there is new impetus to find answers that allow both providers and users to place some objective benchmarks against the current state of the current service offerings, putting in mind the need for a functional description of network performance addressing the distortion of transactions that are carried across the network, one of which is this research work focus. (`Request for Comments (RFC 5181), 2008`) provided a detailed description of IPv6 deployment and integration methods and scenarios in wireless broadband access networks in coexistence with deployed IPv4 services in which the main components of IPv6 IEEE 802.16 access networks and their differences from IPv4 IEEE 802.16 networks, were discussed and also how IPv6 was deployed and integrated in each of the IEEE 802.16 technologies.

With some new features introduced by IPv6 network, such as a wider address space, a different scope based on addressing scheme, a modified protocol field, according to (*M-K. Shin Ed., Y-H. Han, S-E., Kim, and D. Premec, 30 July 2007*) the IPv6 topology discovery faces some challenging problems, one of which is the ability to transmit real time applications including voice, and many of topology discovery algorithms developed to collect IPv4 network topology information cannot be readily applied to IPv6 network. Problems generated when adopting IPv6 network, and some methods and algorithms for IPv6 network topology discovery need to be covered in detail when considering network performance improvements discussing the advantages and disadvantages highlighting also some challenging problems faces IPv6 network infrastructure which affect the overall protocol performance focusing on the network level. *(Cisco Support Community, Document ID: 5125, Feb 2, 2006)* discussed the voice performance behavior in between transmission rates of 150-to-400 Milliseconds of delay limited by the 150 milliseconds hit value in VoIPv6 as stated by the ITU provided that network administrators are aware of the Echo cancellers are required when one-way delay exceeds a value of 25 milliseconds as in recommendation G.131 of the ITU. For private VoIPv6 networks a conditional value 200 milliseconds of delay is a reasonable goal and 250 milliseconds is a value of maximum conditional limit, provided priory that the maximum expected voice connection delay is known as 150Ms, Table 1 illustrate this.

TABLE I. Itu Delay Specifications

| Range in Milliseconds | Description |
|---|---|
| 0-150 | Acceptable for most user applications. |
| 150-400 | Acceptable provided that administrators are aware of the transmission time and the impact it has on the transmission quality of user applications. |
| Above 400 | Unacceptable for general network planning purposes. However, it is recognized that in some exceptional cases this limit is exceeded. |

ITU G.711, and (*Cisco Support Community, Document ID: 5125, Feb 2, 2006)* specified limits for the acceptable amount of delay under which the voice quality will be acceptable, and this paper discuss the impact and the percentage that the network related functions contribute to this amount. (*Request for Comments:* RFC 2460, *R. Hinden,* Published by the IETF, *Category: Standards Track Nokia December 1998*) specified version 6 of the Internet Protocol (IPv6), also sometimes referred to as IP Next Generation or IPNG. RFC 2460 specified an Internet standards track protocol for the Internet community and requests discussion and suggestions for farther improvements. Reviewers, according to RFC 2460, and by referring to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol, IPv6 solves the problem imposed by the limited 32-bit address field in IPv4, and the definition of the new protocol (IPv6) results in longer header size, which affects services that used to be transmitted over IP especially real-time services such as voice. (*El-Hennawy, Hadia M S, Ain Shams University, Cairo, Egypt, Gomaa, Haitham H.; Amin, Amany S.* 07 April 2010) demonstrated the effectiveness of transmitting voice traffic over IPv6 convergence using simulation results obtained using NS2.

(*W. Richard Stevens*, 1998) offered guidance tips on making the most of sockets, which are the de facto standard for UNIX network programming as in this research work the socket programing was the performance testing tool which is FreeBSD. (*W. Richard Stevens*, 1998) begin his book by introducing virtually every basic capability of TCP and UDP sockets, including socket functions and options, I/O multiplexing, also the name and address conversions presenting detailed coverage of the Posix, for example standard for the sockets and the Posix threads and also introduced advanced techniques for establishing both the IPv4/IPv6 interoperability, implementing non-blocking Input /Output routing sockets broadcasting, in addition to the multicasting as IP options, multithreading, advanced name and address conversions, UNIX based and domain protocols, and also raw sockets, such techniques teaches researchers how to choose among today's leading client/server design approaches. This research work tackled this idea using FreeBSD, including both the TCP iterative, concurrent, and Pre-forked and in addition to Pre-threaded servers, in this context the Internet/intranet revolution has dramatically increased in the demand for researchers graduating with a sophisticated understanding of network programming APIs especially sockets helping those researchers and students to achieve that goal.

(Habibullah Jamal, Kiran Sultan, 2008) discussed a possible performance improvement in the protocol level of the dominant transport protocol of today, TCP, which does not meet the demand for high quality voice because it favours reliability Over timeliness and fails to fully utilize the IP network capacity, most certainly this is due to limitations of its conservative congestion control algorithm. This paper *rather focus on testing how well UDP (User Datagram Protocol) and TCP (Transmission Control Protocol) performs in sending and receiving VoIPv6 traffic and bulk data transfer, and* the slow response of TCP in fast long distance networks leaves sizeable unused bandwidth in such networks. A large variety of TCP variants have been proposed to improve the connection’s throughput by adopting more aggressive congestion control algorithms. Some of the flavours of TCP congestion control are loss-based, high-speed TCP congestion control algorithms that uses packet losses as an indication of congestion; delay-based TCP congestion control that emphasizes on packet delay rather than packet loss as a signal to determine the rate at which to send packets. Some other efforts combined the features of loss-based and delay-based algorithms to achieve fair bandwidth allocation and fairness among flows.

(W. Lei, W. Zhang, and S. Liu (North-eastern University), July 25, 2014) stated that in the year 2014 most multimedia applications utilized a combination of real-time transport protocol (RTP) and user datagram protocol (UDP). Application programs at the source end format payload data into RTP packets using RTP specifications and dispatch them using unreliable UDP along a single path multipath transport as an important way to improve the efficiency of data delivery. In order to apply the framework of multipath transport system based on application-level relay (MPTS-AR) to RTP-based multimedia applications defining multipath real-time transport protocol based on application-level relay (MPRTP-AR), which is a concrete application-specific multipath transport protocol (MPTP), and the packet formats and packet types of MPRTP-AR follow the common rules specified in MPTP profile.

## IV. The Aim of this Research Work

The aim of this research is to test and evaluate the VoIPv6 (Voice over Internet Protocol v6) network related functions that are frequently used in both the UDP, and TCP client and the server programs, functions like socket applications (Client/Server) with combination of functions such as Sendto and Receivefrom, this is as part of the accumulated nature of the VoIPv6 overall delay in the same track as in (Kaushik Das, 2008), and *(Krishna P. Gummadi, Stefan Sarotu, W.A. Steven D. Gribble,of Seattle, and W.A. King, 2002), and* according to (*Mark Allman of the* Internet Measurement Research Group (IMRG) Australia, *06 August 2012*), the Internet is subjective to continuous measurements, testing, and evaluations, and accordingly improvements for its ability to convey and cope with real time traffic.

## V. Experiments

There is a continuous need to test how good VoIPv6 network is as *(Geoff Huston of Telstra, GEOFF HUSTON Holds March 2003*) also investigated this issue, his approach was the broad spectrum of the network compared to this research work focus on a specific part of the network which is the network related functions. The first test in this research work was run using developed Free BSD codes for VoIPv6 network; see also (*Asaad A. Abusin, M. D. Jahangir Alam, Junaidi Abdullah, 2012).* These tests did not measure the whole application performance but only the selected function calls. The time was captured just before the function was called and immediately after the function return, as in the developed FreeBSD code below:

```
Struct timeval timestart, timeend;

gettimeofday(&timestart, 0);

int s = Sockent(AF_INET, SOCK_STREAM, 0);

gettimeofday(&timeend, 0);
```

Free BSD developed code to capture the time taken to process the calls.

All tests and experiments were conducted in Network and PCs with the following platform specification and parameters:

The test network platform was MMU Multimedia University Malaysia intranet.

Single Pentium II 450 MHz processor

64 Mbytes 100 MHz Random Access Memory

Free BSD 4.5-Released Operating system.

KAME version 20010528/Free BSD

### A. *Developing UDP IPv6 Client and Server:*

IPv6 UDP source (Client) and sink (Server) source codes were developed, part of the codes are shown below in Code 1 and 2 respectively.

## VI. CODES

### A. *Code1:*

IPv6 UDP Client:

Part of the developed code is shown in code 1 below:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdp.h>
#define MYPORT 4950    // the port users will be connecting to
int main(intargc, char *argv[])
{
intsockfd;
struct sockaddr_in6 their_addr; // connector's address information
structhostent *he.
intnumbytes;
char buf6(INET6_ADDRSTRLEN)
#define MYPORT 4950    // the port users will be connecting to
int main(intargc, char *argv[])
{
intsockfd;
struct sockaddr_in6 their_addr; // connector's address information
structhostent *he.
intnumbytes;
char buf6(INET6_ADDRSTRLEN)
if (argc != 3) {
fprintf(stderr,"usage: talker hostname message\n" );
exit(1);
}
if ((he=gethostbyname2 argv[1], AF_INET6)) == NULL) {  // get the host info
perror("gethostbyname");
exit(1); ]
```

### B. *Code 2:*

IPv6 UDP Server:

Part of the developed code is shown in code 2 below:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#define MYPORT 4950    // the port users will be connecting to
#define MAXBUFLEN 100
int main (void)
{
intsockfd;
struct sockaddr_in6 my_addr;   // my address information
struct sockaddr_in6 their_addr // connector's address information
intaddr_len, numbytes;
charbuf[MAXBUFLEN],buf6addr [INET6_ADDRSTRLEN];
perror("bind");
exit(1);
}
addr_len = sizeof(structsockaddr) ;
```

```
if ((numbytes=recvfrom(sockfd,buf, MAXBUFLEN-1, 0,
 (structsockeddr*)&their_addr,&addr_len)) == -1) {
perror("recvfrom");
exit(1);
}
close(sockfd);
return 0;
}
```

In the UDP client code, communication socket pair (Server IP address and port number) must be obtained first then only VoIPv6 stream could be sent to a specific UDP server.

Obtaining the communication socket is crucial to obtain first the pair (IP address and the port number) for the communication to take place between the client and the server. In the client code side obtaining the communication socket using FreeBSD developed code can be:

```
IntsockFd;
If ((sockfd=socket(AF_INET, SOCK_DGRAM, 0))==.1) (
Perror("socket");
Exit(1);
```

## VII. Test Results

A test to obtain the communication socket in IPv6 network:

A test was conducted to quantify the duration of time needed to obtain the socket of communication in the TCP and UDP client within Multimedia University (MMU) intranet: Table 2 Shows Different Trails and Tests to Obtain the Required Socket and the Average Time to obtain them.

TABLE II. Time Needed to Obtain the Communication Socket in IPv6 UDP Client in Microseconds

| Test Nomber | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Average |
|---|---|---|---|---|---|---|---|---|
| Time in<br>Mmilliseconds | 19 | 19 | 20 | 18 | 19 | 20 | 18 | 19 |

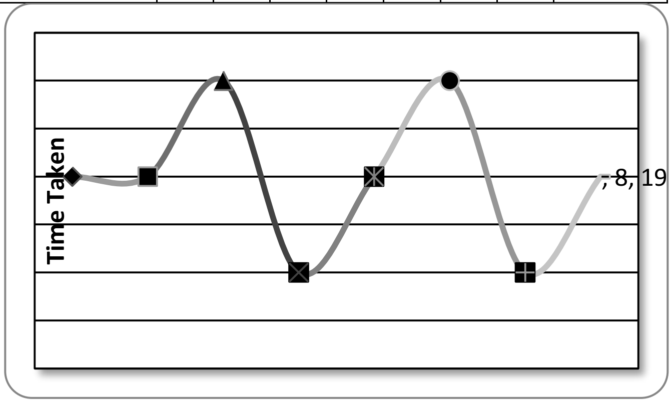

Fig. 1. Time taken to obtain the communication socket in IPv6 UDP client

### A. Comparing the VoIPv6 maximum delay limit with the time needed to obtain the communication socket:

When comparing the time needed to obtain the communication socket with the VoIPv6 voice less limit which is 150 Milliseconds according to (Request for Comments (RFC 5181), 2008), the time needed to obtain the communication socket in IPv6 UDP client according to this research work above tests is around 19 Microseconds with contribution of 6% to the less possible amount of delay stated by the Request for Comments (RFC 5181), 2008) above which the voice quality will not be acceptable.

This is the main contribution in this research work which is evaluating and quantifying the time needed to obtain the communication socket and comparing it with the maximum possible delay limit stated by (Request for Comments (RFC 5181), 2008).

The Sendto socket call is needed to send packets and information to the UDP server; this is the case in any UDP client/server application. The Sendto call in the IPv6 UDP client program is shown in Code 1 above. The performance was tested to know the time duration needed to transmit different sizes of data in IPv6 UDP client; Table 3 shows this:

TABLE III. Time Needed to Send Different Packet Sizes in IPv6 UDP Clients

| Bytes Sent | 100 bytes | | | Av. |
|---|---|---|---|---|
| | TS1 | TS2 | TS3 | |
| Time in microseconds Ipv6 (Native) | 49 | 48 | 49 | 49.7 |

| Bytes Sent | 200 bytes | | | Av. |
|---|---|---|---|---|
| | TS1 | TS2 | TS3 | |
| Time in microseconds Ipv6 (Native) | 51 | 50 | 51 | 50.7 |

| Bytes Sent | 300 bytes | | | Av. |
|---|---|---|---|---|
| | TS1 | TS2 | TS3 | |
| Time in microseconds Ipv6 (Native) | 53 | 53 | 52 | 52.70 |

| Bytes Sent | 400 bytes | | | Av. |
|---|---|---|---|---|
| | TS1 | TS2 | TS3 | |
| Time in microseconds Ipv6 (Native) | 53 | 54 | 53 | 53.3 |

| Bytes Sent | 500 bytes | | | Av. |
|---|---|---|---|---|
| | TS1 | TS2 | TS3 | |
| Time in microseconds Ipv6 (Native) | 55 | 55 | 55 | 55 |

### B. *Comparing the VoIPv6 maximum delay limit with the time needed to obtain the communication socket in IPv6 UDP Client (Via Native IPv6):*

The Sendto socket call of the UDP client was tested under IPv6 native environment with one network switch separating the IPv6 TCP and UDP client and server, the IPv6 TCP and UDP servers were given a test IPv6 addresses, and the client was given a different test address. The tested IPv6 addresses of the client and server were configured first using Inet6 function call as below:

Ipconfig x10 Inet6 1::1, (Where x10 is the network driver card from 3Com)

TABLE IV. The Obtained Results. Time Needed to Send Different Packet Sizes Using Ipv6 UDP, Voipv6 Clients Through Native Ipv6 Test Bed Networks

| Bytes Sent | 100 bytes | | | Av. |
|---|---|---|---|---|
| | TS1 | TS2 | TS3 | |
| Time in microseconds Ipv6 (Native) | 70 | 70 | 69 | 69.67 |

| Bytes Sent | 200 bytes | | | Av. |
|---|---|---|---|---|
| | TS1 | TS2 | TS3 | |
| Time in microseconds Ipv6 (Native) | 73 | 75 | 88 | 78.67 |

| Bytes Sent | 300 bytes | | | Av |
|---|---|---|---|---|
| | TS1 | TS2 | TS3 | |
| Time in microseconds Ipv6 (Native) | 76 | 76 | 76 | 76 |

| Bytes Sent | 400 bytes | | | Av |
|---|---|---|---|---|
| | TS1 | TS2 | TS3 | |
| Time in microseconds Ipv6 (Native) | 78 | 89 | 77 | 81.33 |

| Bytes Sent | 500 bytes | | | Av |
|---|---|---|---|---|
| | TS1 | TS2 | TS3 | |
| Time in microseconds Ipv6 (Native) | 76 | 78 | 75 | 76.33 |

The Performance of the Sendto UDP Socket call in Native IPv6 was tested and the results are shown in Figure 2.

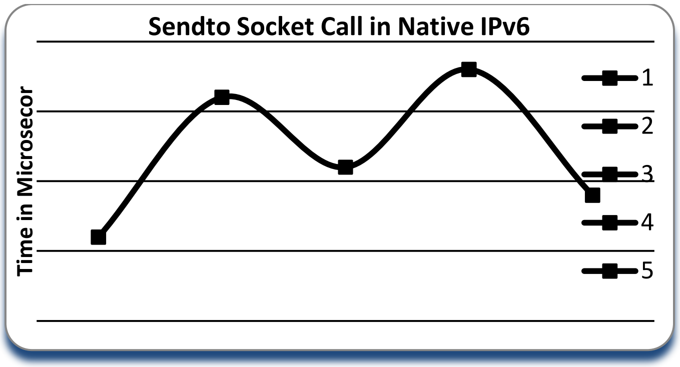

Fig. 2. Performance of the Sendto UDP Socket call in Native IPv6 Network

### C. *IPv6 UDP Clients (Via Tunneled IPv6 over IPv4):*

In order to tunnel IPv6 over IPv4, it is needed first to configure an IPv6 address for the UDP client and the server using the following FreeBSD (UNIX based) line codes:

Ifconfig giff) create

Gifconfig giff) source-Ipaddres(IPv4) destination IP address(IPv4)

Ifconfig giff) Inet6 source-Ip-address(IPv6) destination IP-address (IPv6) prifixlen/x

The command line Ifconfig here configures the physical address for the generic IP tunnel interface, such as giff which is a pseudo device for IPv4-IPv6 tunneling (see FreeBSD Hypertext Man Pages, 2003). The obtained results were shown in Table 5.

TABLE V. Time Needed to Send Different Packet Sizes Using Developed Ipv6 UDP Client Through: Native Ipv6 Test Bed Network

| Bytes Sent | 100 bytes | | | Av. |
|---|---|---|---|---|
| | TS1 | TS2 | TS3 | |
| Time in microseconds Ipv6 (tunneled over IPv4) | 86 | 87 | 86 | 86.3 |

## VIII. Conclusions and Future Works

The time taken to obtain the communication socket in IPv6 (TCP, UDP) clients was tested and quantified, and the contribution of this in the less possible amount of delay for VoIPv6 was specified. The VoIPv6 performance was tested in IPv6, in the TCP and UDP transmission protocols in a test bed IPv6 network confirming (M-K. Shin, Ed., Y-H Han, S-E Kim, D. Premec, 30 July 2007) expectation that the VoIP performance might not be the same in a tunneled IPv6 over IPv4 networking system, this is due to the simplicity of the IPv6 header which consists of seven fields compared to the IPv4 header which consist of 13 fields in the header affecting the voice test results in IPv6.

From all the tests and analysis done, one conclusion can be derived: That is the performance of the applications running over IPv6 are affected by the time needed to obtain the communication socket, this is due to the protocol nature, and accordingly any improvement in the protocol level or in the network level will contribute to improving the overall VoIPv6 performance as the long address space of IPv6 adds some overhead to the packet processing time, and the removal of checksum operation at the IP level in IPv6 is not enough to recover the long address processing time. Simpler IPv6 header does not help in this case because experiments in this research work were conducted in local loop environment however, from the theoretical point of view, it will enhance the end-to-end performance if there are many intermediate hosts in between since there will be less processing in the intermediate hosts and nodes. IPv6 uses smaller send and receive buffer compared to IPv4, this may affect the performance since small buffer become full faster. According to the IETF as in (Request for Comments (RFC 5181, 2008) IPv6 is still in its infancy stage experiencing continues renovation. The IPv6 KAME stack is still under development and frequently being changed and updated. The current updated release of IPv6 is not the optimum all the time.

Future work as a continuation to this research work can be carried out in investigating the performance of the implementation of the IPv6 stack in different platforms. According to (Request for Comments (RFC 5181), 2008), KAME IPv6 stack implementation needed continually to be improved to the most optimal level to give better performance results supporting different applications. These performance tests can be repeated in different network conditions highlighting the differences and similarities. The IPv6 features supporting real-time traffic can be clearly noticed as discussed in (Request for Comments: RFC 2460, R. Hinden, Published by the IETF, Category: Standards Track Nokia December 1998), and (El-Hennawy, Hadia M S, Ain Shams University, Cairo, Egypt. Gomaa, Haitham H.; Amin, Amany S. 07 April 2010). The evolving enhancement in IPv6 network with the introduction of Broadband internet service can contribute to better performance, and performance improvements in the real-time applications under IPv6 supported also by (Request for Comments (RFC 5181), 2008).

## References

[1] IETF, (May 2008), Rfc 5181 Title: IPv6 Deployment Scenarios in 802.16 Networks, and their differences from IPv4 IEEE 802.16 networks and how IPv6 is deployed and integrated in each of the IEEE 802.16 technologies, Internet Engineering Task forth.

[2] Asaad A. Abusin, M. D. Jahangir Alam, Junaidi Abdullah, (May 2012), Testing and Analysis of VoIPv6 (Voice over Internet Protocol V6) Performance Using FreeBSD, IJCNS, Volume 5, No. 5.

[3] Kaushik Das, (2008), VoIP – Next Generation of Voice & IPv6, IPv6.com Technical spotlight.

[4] Krishna P. Gummadi University of Washington, Seattle, Stefan Sarotu University of Washington, Seattle, W.A, Steven D. Gribble University of Washington, Seattle, and WA

[5] King (University of Washington Seattle Department of Computer Science & Engineering), (2002), Estimating Latency Between Arbitrary Internet End Hosts, Proceeding IMW '02 Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement, Pages 5-18 ACM New York, NY, USA

[6] Internet Measurement Research Group (IMRG), (06/08/2012), Internet Measurement Research, the IMRG Chair (Mark Allman.

[7] GEOFF HUSTON (Chief Scientist in the Internet area for Telecom Australia (Telstra) E-mail: gih@telstra.net), (March 2003), How can the quality of the service that specific network is offering can be measured and quantified?, Volume 6, Number 1, GEOFF HUSTON Holds and the Australian National University.

[8] Network Working Group Request for Comments (RFC 5181), (May 2008), IPv6 Deployment Scenarios in 802.16 Networks, Internet Engineering Task Forth (IETF)

[9] Luo Jun Hai, Chengdu Fan Mingyu, and Ye Danxia (University of Electronics, Since and Technology of China), (30 July 2007), Research on Topology Discovery For IPv6 Networks, ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel Distributed Computing.

[10] Cisco Support Community, (Feb 2 2006), Understanding Delay in Packet Voice Networks [Voice Quality]. Cisco Support Community

[11] , Document ID: 5125.

[12] R. Hinden. Category: Standards Track Nokia, ( December 1998), Request for Comments (RFC 2460), The International Telecommunication Union (ITU).

[13] El-Hennawy, Hadia M S . Electron. &Electr. Commun. Dept., Ain Shams University, Cairo, Egypt. Gomaa, Haitham H.; Amin, S. Amany, (07 April 2010), The Effectiveness of Transmitting Voice Traffic over IPv6 Convergence Sub-layer of WiMAX Mesh Networks, IEEE, the 17th International Conference on Telecommunications (ICT). Product Type: Conference Publications.

[14] Pages: 293-298

[15] W. Richard Stevens, (1998), UNIX Network Programming, Volume 1, Second Edition: Networking APIs: Sockets and XTI, Prentice Hall, 1998, ISBN 0-13-490012-X. Prentice Hall, 1998, ISBN 0-13-490012-X

[16] Habibullah Jamal, Kiran Sultan, (2008), Performance Analysis of TCP Congestion

[17] Control Algorithms, International Journal of Computers and Communications, Issue 1, Volume 2.

[18] W. Lei, W. Zhang, and S. Liu (North-eastern University), (July 25, 2014), Multipath Real-Time Transport Protocol Based on Application-Level Relay (MPRTP-AR), Internet-Drafts working documents of the Internet Engineering

[19] Task Force (IETF).