Cite it as :
Bayan Khannah and Ahmad Alsadeh, “Impact of IPv4/IPv6 Transition Techniques on Applications Performance”, International Conference on Electronics, Information and .Communication (ICEIC2017), 11-14 January 2017 in Phuket, Thailand. pp. 374—381, 2017

Impact of IPv4/IPv6 Transition Techniques on Applications Performance

Bayan Khannah
Birzeit University, Palestine
Email: khbayaan@gmail.com

Ahmad Alsa’deh
Birzeit University, Palestine
Email: asadeh@birzeit.edu

***Abstract*—During the coexistence period between IPv4 and IPv6 networks, it is important to test the effect of using IPv6 transition techniques on applications’ performance. We evaluate the performance of specific user’s applications over three transition techniques: dual-stack, automatic 6to4 and manual tunneling. A set of experiments have been carried out using OPNET network simulator to evaluate the performance of five applications: web browsing, file transfer, voice, email and database access over these transition techniques and comparing with applications’ performance over pure IPv4 and IPv6 networks. The finding results show varying in the applications’ performance between dual-stack, automatic 6to4 tunneling, and manual 6to4 tunneling. For most applications, dual-stack performed better than tunneling regarding the response time. In some cases, tunneling performed better than dual-stack regarding other performance parameters, such as throughput and Jitter.**

***Index Terms*—IPv6 Transition; Dual-stack; Automatic Tunneling; Manual Tunneling**

I. Introduction

Internet Engineering Task Force (IETF) developed Internet Protocol, Version 6 (IPv6) to solve the problem of address exhaustion with the IPv4 [1]. Thus the main advantage of IPv6 over IPv4 is the address size. IPv4 has 32-bit address length which gives 2^{32} addresses while IPv6 has 128-bit address length which gives 2^{128} addresses [2]. Compared to IPv4, IPv6 has unlimited address space. However, IPv6 has many new features over IPv4. With these enhanced features, IPv6 is not backward compatible with IPv4. This means that IPv6 hosts cannot communicate with IPv4 hosts directly and vice verse.

Nowadays with the numerous growth of Internet usage devices, the migration from IPv4 to IPv6 protocol become a necessity. However this migration could not be happen all over the world at the same time. Such operation is complicated and needs a lot of time due the compatibility problem between IPv4 and IPv6 networks. Therefore, specific techniques were designed to enable the communication between IPv4 and IPv6 hosts during the transition period.

Many transition techniques have been implemented to ensure a smooth, stepwise and independent transition to IPv6 with existing IPv4. Each technique has its own behavior and strategy. These techniques can be classified into three major categories; Dual-stack, Tunneling and Translation.

Many difficulties face IPv6 transition techniques; one of these difficulties is its impact on the web applications performance. The concern is to face poor application performance during the transition period. Testing and monitoring the performance during this period will help in determining the weak points and resolve it before full migrating to the IPv6.

This paper studies and analyzes the performance of various web applications (VOIP, Web browsing, File transfer, Email and Database) over three transition techniques (dual-stack, manual and automatic 6to4 tunneling) and compare these results with results obtained from pure IPv4, pure IPv6 networks. First, we configured the three IPv6 transition mechanisms using GNS3 simulator and then we measured the impact of these mechanisms on different web applications using OPNET. We use OPNET network simulator because it provides an extensive library models for different applications traffic. We present a comparative analysis between them regarding to various parameters that affect the overall applications performance in the network, such as the response time and the amount of traffic received by the applications.

The results that we obtained by the manual ping test show that dual-stack has the best performance regards the network delay. The finding results show that there is a difference in these applications performance between dual-stack, manual tunneling, and automatic 6to4, and between IPv4 and IPv6 networks. Dual-stack is the best choice for most of the applications that have been studied regarding to the response time. This is due the delay which is caused by the encapsulation and decapsulation processes in tunneling while in dual-stack the two protocols are working concurrently. In some cases tunneling had better performance than dual-stack regarding the amount of traffic received in web browsing applications and the jitter value in the voice applications. In some cases manual tunneling performed better than automatic and in other the automatic 6to4 performed better.

II. IPv6 Transitions

IPv6 Transition refers to the changing of the Internet from its current IPv4 infrastructure to the successor addressing and routing system of IPv6 [3]. Since IPv4 and IPv6 networks are not compatible, specific techniques were designed to permit the communication between IPv4 and IPv6 hosts during the transition period. These techniques are classified into three main categories [4]: Dual-stack, Tunneling, and Translation.

A. Dual-stack

Dual-stack includes both of IPv4 and IPv6 protocols working in parallel. It can be implemented in both end system and network node. In end systems, they enable both IPv4 and IPv6 applications to operate at the same time. IPv4 applications use the IPv4 stack, while IPv6 applications use the IPv6 stack [4]. A dual-stack node has support both protocol versions and is referred to as an IPv6/IPv4 node. Flow decisions are based on the version field of IP header for receiving, and on the destination address type for sending.

B. Tunneling

Tunneling refers to minimize any dependencies between the two protocols during the transition, so that all the routers in the path between two IPv6 nodes do not need to support IPv6. Basically, IPv6 packets are placed inside IPv4 packets, which are routed through the IPv4 routers. The entry node of the tunnel (the encapsulating node) creates an encapsulating IPv4 header and transmits the encapsulated packet. The exit node of the tunnel (the de-encapsulating node) receives the encapsulated packet, removes the IPv4 header, updates the IPv6 header, and processes the received IPv6 packet. The IPv6 tunnels can be classified according to their configuration into manual and automatic tunnels:

1) *Manual Tunneling*. Manual tunnels must be configured manually. It allows IPv6 traffic to be carried across an IPv4 network. Tunnel destination address is specified in the tunnel source configuration creating a P2P (peer to peer) topology [4]. The tunnel acts as 1 hop for a IPv6 packet whereas an IPv4 encapsulation packet may take many hops. These tunnels are used when using IPv6 addresses that do not have any embedded IPv4 information. The IPv6 and IPv4 addresses of the endpoints of the tunnel must be specified.
2) *Automatic Tunneling*. Automatic tunnels are configured by using IPv4 address information embedded in an IPv6 address, the IPv6 address of the destination host includes information about which IPv4 address. The packet should be tunneled to [4]. Automatic Tunneling has many types:
   - IPv4 compatible IPv6 - RFC 4213
   - IPv6 over IPv4 - RFC 2529
   - IPv6 to IPv4 RFC 3056
   - ISATAP (Intra-Site Automatic Tunnel Addressing Protocol)-RFC 5214
   - Tunnel Broker RFC 3053
   - Teredo RFC 4380

C. Translation

The concept of address translation is not a new concept; because Network Address Translation (NAT) is implemented between different IPv4 networks in almost every home networks. The IPv6 translation has been proposed in RFC 2766 [5] and obsoleted by RFC 4966 [6]. Here, translation means that a device on the network converts IPv4 packets into IPv6 packets and vice versa. The device must be capable of performing this translation both ways so that bidirectional communication between end hosts is possible. Two main types of IPv6 translation [5]: Network Address Translation - Protocol Translation (NAT-PT), and NAT64 and DNS64.

III. Related Work

The transition topic of IPv6 has been in discussion for years. Three transition mechanisms; dual-stack, tunneling and translation were discussed in [7]. Dual-stack was also discussed in [4]. The dual-stack, tunneling, and translation mechanisms were presented as well as technical issues related to IPv6 deployment in [3]. The transition of un-managed networks from IPv4 to IPv6 has been discussed in [8]. Un-managed networks typically correspond to home networks or small office networks.

Many researchers evaluated the performance of IPv6 transition methods. 6to4 transition mechanism and tunneling were empirically compared on a testbed setup with Windows 2000 operating system in [9]. The performance of configured and 6to4 tunnels on Linux and Windows Server has been evaluated using throughput, latency, and CPU usage in [10]. The performance of IPv6 transition mechanisms over the Multi-protocol Label Switching (MPLS) backbone has been analyzed in [11], it proved that dual-stack has the best overall performance metrics compared to other transition techniques. A comparison of the performance of automatic and manual transition strategies in order to show how the transition strategy affects network behavior has been illustrated in [12]. Other research has proved the ease of TCP data transmission using 6to4 tunneling compared to native IPv4 and IPv6 networks [13]. A comparison of recent IPv6 tunneling techniques: Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) and Teredo using UDP audio streams, video streams, and ICMP-ping traffic [14] has been performed through certain parameters, namely: tunneling overhead, tunnel setup delay, query delay, and auxiliary devices required.

Basic problems and difficulties that faces the IPv4-IPv6 transition were discussed in [15], it also introduced the principles of tunneling and translation techniques. Other research area focused on the performance evaluation of video and voice over IP using various IPv6 transition techniques [16]. The performance of FTP over the dual-stack tunneling has been studied by analyzing the size of files transfer performance and measure the Quality of Service (QoS) delivered by IPv6 using best effort approach in comparison to IPv4 [17].

IV. Simulation of IPv6 Transition Techniques

This section includes the experimental network setup design and network testbed. It explains the structure of the network for each transition mechanism. It also includes network and IP transition mechanisms configurations. Three transition mechanisms (Dual-stack, Manual and automatic 6to4 tunneling) have been implemented using GNS3 network simulator. GNS3 is a good network simulator, it supports most of the IPv6 transition techniques and can easily be configured to connect virtual machines to simulate the real ones.

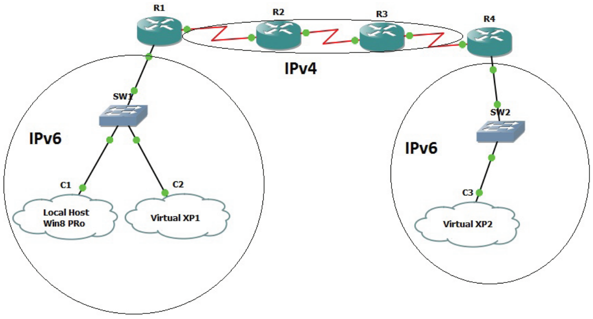

Fig. 1: GNS3 Network Simulation

A. Experiment Requirements

Hardware: a personal computer of Processor speed 1.87 GHz, 32-bit operating system, x64-based processor and 4.00 GB RAM is used.
Software: GNS3 network simulator with 4 Cisco routers of 3660 IOS and 2 Ethernet switches, Windows 8.1 Professional, and VMware software to implement virtual hosts which used in GNS3 network.

A network testbed was implemented with 4 Cisco routers of 3660 series as an IPv4 network, and at each end an IPv6 host is connected by Ethernet switch. All the routers and network hosts were configured, routing protocol RIP for the two protocol versions was configured and the network was tested for each transition strategy so that IPv6 hosts in the test network can communicate via IPv4 network.

B. Network Configuration

This section describes the basic configuration of the two networks ends (R1 and R4 in Figure 1) for the three transition techniques. In all cases the routing protocol RIP for version 4 and version 6 was configured.

1) Dual-stack: In dual-stack, all interfaces and hosts were configured with both IPv6 and IPv4 protocols.

R1:

```
ipv6 unicast-routing
interface Serial1/0
ip address 172.168.0.1 255.255.255.0
ipv6 enable
ipv6 address 2001:2:2:2::1/64
```

R4:

```
ipv6 unicast-routing
interface Serial1/1
ip address 172.168.7.2255.255.255.0
ipv6 enable
ipv6 address 2001:7:7:7::2/64
```

2) Manual Tunneling: An IPv6 address is configured manually on a tunnel interface, IPv4 addresses are configured manually to the tunnel source and the tunnel destination. The router at each end of a configured tunnel must support both IPv4 and IPv6 protocols. The connection for the manual is a point to point (P2P) mode which is assigned the source and the destination address of the tunnel by the operator.

R1:

```
ipv6 unicast-routing
interface Serial1/0 ip address 172.168.0.1
255.255.255.0
!
interface Tunnel 0
ipv6 address 2001:DB8:1122:12::1/64
tunnel source 172.168.0.1
tunneldestination172.168.2.2
tunnel mode ipv6ip
!
ipv6 route 2001::/16 Tunnel0
```

R4:

```
ipv6 unicast-routing
interface Serial1/1
ip address 172.168.2.2 255.255.255.0
!
interface Tunnel 0
ipv6 address 2001:DB8:1122:12::2/64
tunnel source 172.168.2.2
tunnel destination 172.168.0.1
tunnel mode ipv6ip
!
ipv6 route 2001::/16 Tunnel0
```

3) Automatic 6to4 Tunneling: It is a stateless automatic tunneling, uses the encapsulation of IPv6 addresses automatically into IPv4 address. The automatic tunnel is a point to multipoint (P2MP) connection where the source address is assigned by the operator and the destination address is found automatically.

Automatic 6to4 has a special address defined by IANA (2002::/16) [18], RFC 3068 is obsoleted by RFC 7526 in May 2015 [19]. The tunnel address on each tunnel end is a translation of the source and destination IPv4 address to IPv6 address [7]; for example when converting the 138.14.85.210 IPv4 address to hexadecimal address of base 16, the result equals to 8a0e:55d2, and Resulting 6to4 prefix is: 2002:8a0e:55d2::/48. In automatic 6to4 tunnels, the IPv4 address embedded in the IPv6 address is used to find the other end of the automatic tunnel. The tunnel destination is determined by the IPv4 address of the border router extracted from the IPv6 address that starts with the prefix 2002::/16. In this experiment the following addresses were configured:

Tunnel source: 172.168.0.1 = ACA8:1;
IPv6 tunnel source address=2002:ACA8:1:1::1
Tunnel destination 172.168.2.2= ACA8:202;
IPv6 tunnel destination address =
2002:ACA8:202:1::1

R1:

```
interface Serial1/0
ip address172.168.0.1 255.255.255.0
ipv6 unicast-routing
!
interface Tunnel0
ipv6 address 2002:ACA8:1:1::1/64
tunnel source Serial1/0
tunnel mode ipv6ip 6to4
!
ipv6 route 2002::/16 Tunnel 0
```

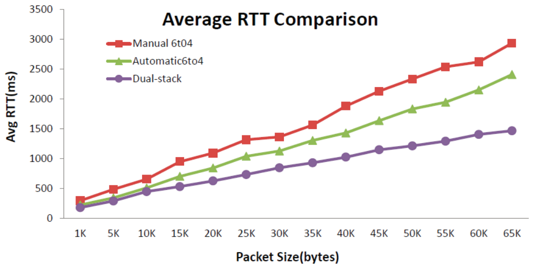

Fig. 2: RTT Comparison

R4:

```
interface Serial1/1
ip address 172.168.2.2 255.255.255.0
ipv6 unicast-routing
!
interface Tunnel0
ipv6 address 2002:ACA8:202:1::1/64
tunnel source Serial1/1
tunnel mode ipv6ip 6to4
!
ipv6 route 2002::/16 Tunnel 0
```

C. Testing the Connectivity

After successful configuration of all the networks, connectivity tests were performed between IPv6 hosts and it was observed that all the IPv4 and IPv6 devices successfully communicated with each other.

For testing the connectivity between the IPv6 hosts in each method; IPv6 packets of various sizes were sent using PING command and tested one by one. During the test no packet loss or any other network faults were noticed. All packets were delivered successfully.

The PING test was done between the two virtual XP hosts (clouds C2 and C3 as shown in Figure 1), from XP1 to XP2, observations were taken from XP1 command prompt using PING command for different packet sizes. For each packet size PING repeated and observed for ten times then the average values were computed.

Round-trip time (RTT) is the time required for a packet to travel from a specific source to a specific destination and back again. RTT is a good indicator of the network latency and the overall network performance, RTT can be found directly by the PING command. The RTT values obtained from the ping command for different packet sizes were observed for the three transition mechanisms one by one, the average values were computed and summarized in Figure 2. The lowest RTT means less delay and thus better network performance. The finding results indicates that dual-stack has the lowest values of RTT for all packet sizes, while automatic 6to4 had the second lowest RTT values. The highest RTT values were measured for manual tunneling.

V. Impact on Applications Performance

During the IPv6 transition period, we can expect web applications performance to be affected. With no compatibility between IPv6 and IPv4 the two networks will have to either live side by side in dual-stack environments or in some cases IPv6 traffic will have to be tunneled through IPv4, encapsulated and then de-encapsulated by routers (i.e. automatic and manual 6to4 tunneling). When something like this happens along the application delivery; this will result in a performance hit, producing performance and availability problems.

The challenge is to avoid the IPv6 transition techniques performance problems, by monitoring the application delivery and manage performance over these techniques. It is important to monitor the performance of web applications during the transition period and to compare the performance of IPv4 and IPv6 applications because no users wants to turn into the new IPv6 technology and find that the performance of their applications does not meet their expectations, this will be disappointing. By monitoring the applications performance over different IPv6 transition techniques and comparing that with IPv4 and IPv6 application performance, we can see where performance falls short and determine what impacted areas need modifications.

A. Experiment Setup

The network was implemented using the Optimized Network Engineering Tool (OPNET). OPNET is an efficient way to provide a complete study for the network analysis. The graphical user interface (GUI) is simple to use and the result is shown as graphical and static. The OPNET analyzes the network as a real life network which gives a complete view before building the network in real life. Therefore, we use OPNET to monitor and capture the performance of the file transfer, web browsing, email and database applications regarding response time and the size of traffic received. For voice applications OPNET was able to monitor the jitter, end to end delay and MOS (mean opinion score) values.

The OPNET modeler provides an extensive model library, including application traffic models (e.g. HTTP, FTP, VOIP, Database Access and Email. It also provides a discrete-event simulation (DES).The DES provides a simulation in the same way as a real network; it is very useful in analyzing the performance and behavior of the protocols and packets. In this experiment the OPNET Riverbed Modeler Academic Edition version 17.5 was used to run simulations on different IPv6 transition techniques. A network testbed including the following components as shown in Figure 3:

- 4 Router nodes (7200 cisco router).
- 2 Ethernet16_switch nodes: these nodes are used to represent switches supporting up to 16 Ethernet interfaces.
- 5 Ethernet_wkstn nodes: to implement 5 workstations (clients) which are running the 5 different applications, connected at the end of the network.
- 5 Ethernet_server nodes: to support 5 applications (Voice, HTTP, FTP, Email, Database), connected at the other end of the network.
- Application_Config node: the application module defines which applications are supported by the network (i.e., web browser HTTP heavy and FTP heavy applications.

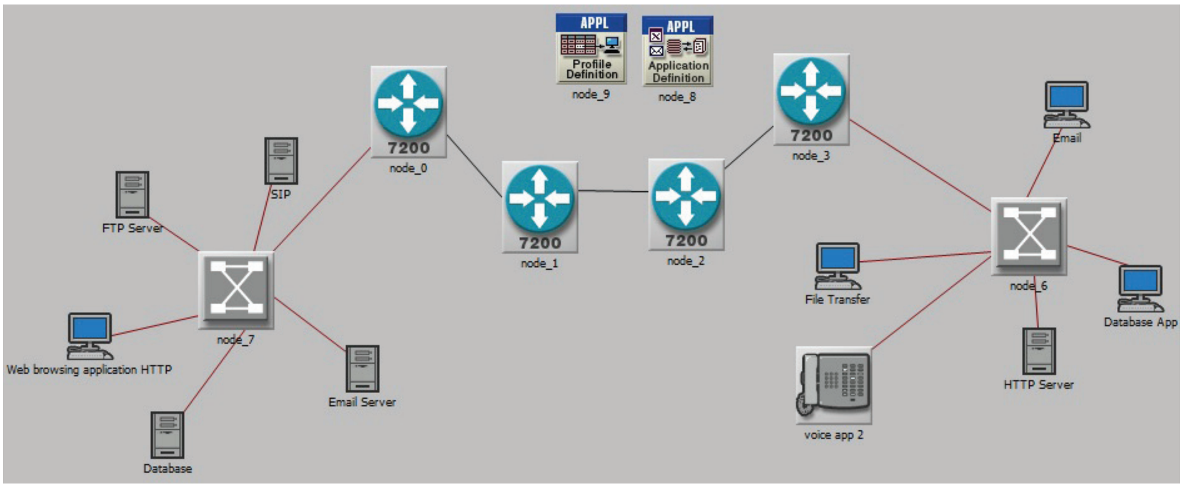

Fig. 3: OPNET Network Test-Bed

The specified application name is used while creating user profiles on the ”Profile_Config” object.

- Profile_Config node: the profile module describes how the users employs the applications defined in the application configuration module. This is used to create user profiles. These user profiles can be specified on different nodes in the network to generate application layer traffic. The applications defined in the Application_Config are used by this object to configure profiles.

The same network testbed with the same applications was configured in 5 different scenarios: Pure IPv4 network, dual-stack network, Manual 6to4 Tunneling, Automatic 6to4 Tunneling and Pure IPv6 Network.

B. Experiment Results

In all cases the simulation process lasted for one hour using Discrete-event Simulation (DES). OPNET computes the desired value at each moment during the simulation time, in this experiment the average values were monitored from the OPNET results viewer and exported to Excel in order to draw bar charts for comparison purposes.

1) Web Browsing: Hyper Text Transfer Protocol (HTTP) [20] is a protocol used for transferring web pages. HTTP takes care of the communication between a web server and a web browser. HTTP is used for sending requests from a web client (a browser) to a web server, returning web content (web pages) from the server back to the client. Web browser is a software application used to locate, retrieve and display content (web pages) via the HTTP, the time it takes for each web page to load is known as response time.

The respose time was monitored for the network in all scenarios and the results were depicted in the Figure 4. Comparisons between IP transition mechanisms regarding the page response time for web browsing application indicate that dual-stack performed better than the other two IPv6 transition mechanisms. This due the delay that caused by the encapsulation and de-capsulation processes in tunneling mechanisms while in dual-stack the two protocols are working concurrently. IPv6 performed slightly better than IPv4. Compared to pure IPv4 and IPv6 networks dual-stack performed as well as the IPv4 network. The next parameter that was monitored is the amount of traffic received by the HTTP client, it affects the performance of web browsing applications. Figure 5 depicts a comparison between the averages bytes per second forwarded to the HTTP Application by the transport layer in the HTTP client. The higher amount of traffic means better performance. Comparing between IP transition mechanisms, the manual tunneling performed slightly better than automatic 6to4 but much better than dual-stack. Comparison between the two protocols, IPv4 performed better than IPv6. Manual tunneling performed better than automatic 6to4 in this case because manual tunnel is point to point connection, which means that the data travel immediately from source to destination while in automatic 6to4 tunnel a point to multi-point connection.

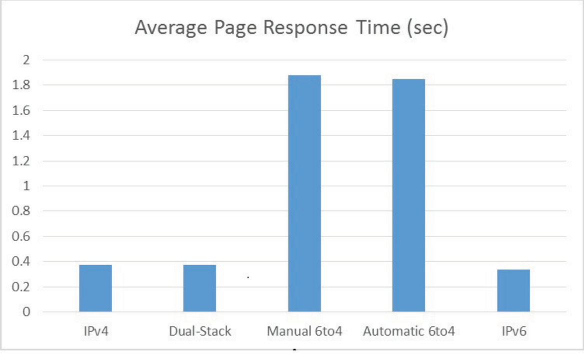

Fig. 4: Page Response Time

2) File Transfer Applications: The File Transfer Protocol (FTP) [21] is a standard network protocol used to transfer computer files from one host to another host. FTP is built on a client-server architecture and uses separate control and data connections between the client and the server. The download

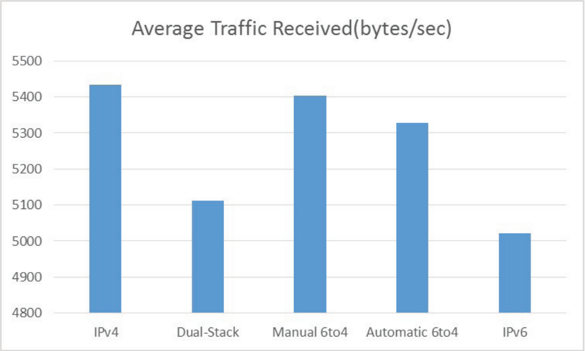

Fig. 5: Received HTTP Traffic

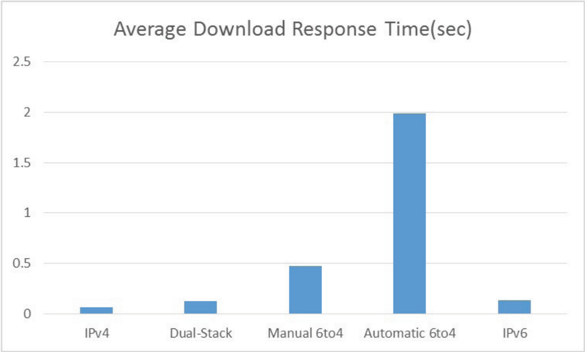

Fig. 6: FTP download Response Time

response time is the time elapsed between sending a request and receiving the response packet for the FTP application. This time is measured from the time a client application sends a request to the server to the time it receives a response packet. It is an important factor in measuring the FTP applications performance, short download response time means better performance. Figure 6 shows that dual-stack has the shortest response time and thus it performed better than the other two tunneling mechanisms, this due the encapsulation and de-encapsulation process in the tunneling mechanisms which will cause more delay in such networks. The difference of delay between manual and automatic 6to4 tunneling comes from the difference of the encapsulation and de-encapsulation strategy between the two tunneling mechanisms. Compared to pure IPv4 and IPv6 networks cases, dual-stack had longer response time; this is due to the both IP stacks processing simultaneously and packets have to flow in two different IP stacks, which caused slight more delay. Other important parameter that affect the performance of FTP traffic is the amount of traffic received by the FTP client. Figure 7 shows a comparison between the averages of bytes per second forwarded to the FTP application by the transport layer in the FTP client. The results indicate that dual-stack had the best performance over all cases. Manual tunneling performed better than automatic and a slight bit difference between the performance of IPv4 over IPv6.

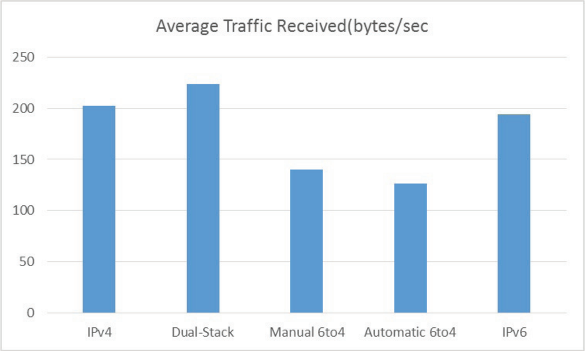

Fig. 7: Received FTP Traffic

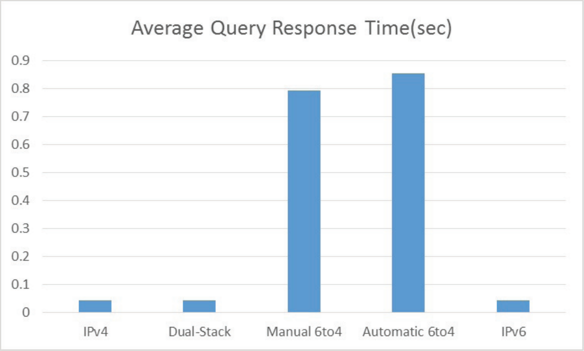

Fig. 8: Query Response Time

3) Database Access: A database is a collection of information that is organized so that it can easily be accessed, A Web database is a database application designed to be managed and accessed through the Internet.

The database query response time is the total amount of time it takes to respond to a given query and it plays an important role in the quality of the web database application, Figure 8 shows a comparison between the effects of the different IPv6 transition mechanisms on this time.

More response time means more delay which will decrease the overall network performance. Dual-stack performed much better than the othe two tunneling methods and again this due the delay cased by the encapsulation and de-encapsulation processes in tunneling. Comparison between the two protocols indicates that IPv4 performed as well as IPv6.

The amount of traffic received is a good indicator about the quality of the database traffic, it illustrates the the average bytes per second forwarded to the database query applications by the transport layer in the database client. This value was monitored for all cases and the results were depicted in the Figure 9. The results of the query traffic received shows that dual-stack performed slightly better than the two tunneling mechanisms. Comparison between the two tunneling mechanisms indicates that automatic tunneling performed better then manual.

4) Email Applications: An email refers to the transmission of messages over communications networks. The response

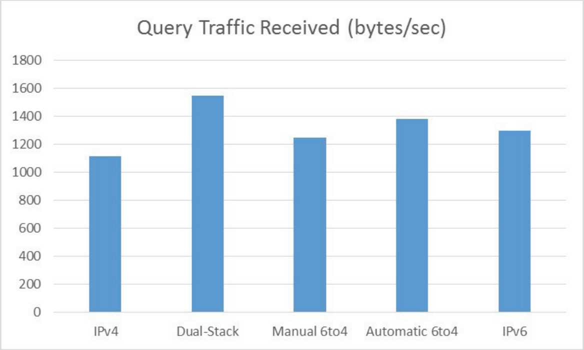

Fig. 9: Received Query Traffic

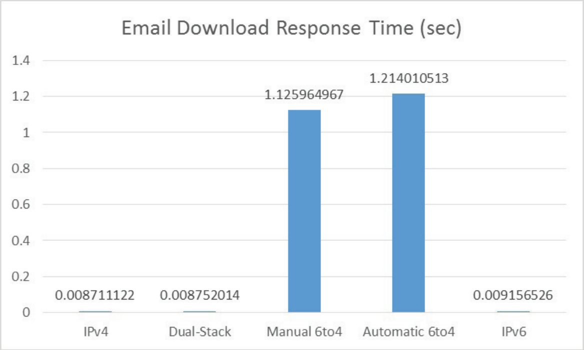

Fig. 10: Email Download Response Time

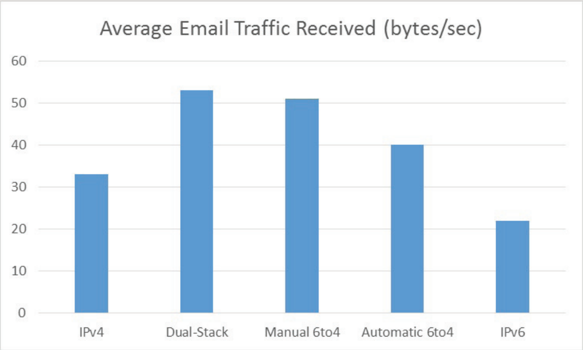

Fig. 11: Received Email Traffic

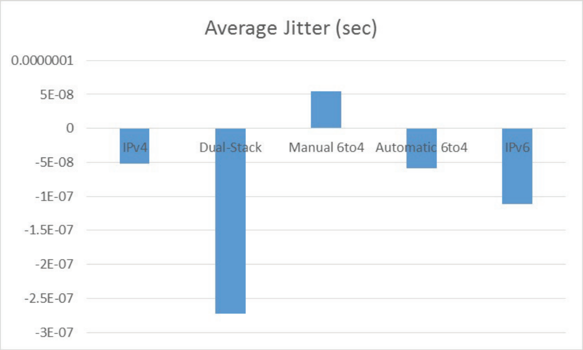

Fig. 12: VOIP Jitter

time for downloading an email is the time elapsed between sending request for emails and receiving emails from email server in the client email. This parameter plays an important role in the quality of an email application. The values of the email response time was monitored during the simulation of the three transition methods plus the IPv4 and IPv6 network and the results were summarized in Figure 10. Dual-stack performs much better than the manual and automatic 6to4 tunneling, manual tunneling performed slightly better than automatic 6to4. Figure 11 shows that dual-stack performed better than manual and automatic 6to4 and manual tunneling performed better than automatic 6to regarding the amount of traffic received. The traffic received here means the average bytes per second forwarded to the Email application by the transport layer in the client Email.

5) Voice Applications: Voice over Internet Protocol (VoIP) [22] is a service that allows users to communicate with each other by voice. A very important indicator about the performance of any voice application traffic is jitter. Jitter is defined as the variation in the delay of received packets [22]. It is important in voice because if transmission delay varies during a VoIP conversation, the quality of voice will be degraded.

From the simulation network in Figure 3. The voice packets transmits from voice app 2 node to ”SIP” server node. The jitter results were monitored on this node during the simulation. The results were depicted in Figure 12. The better jitter value is the closer to zero. The measurements show that manual and automatic 6to4 tunneling perform much better than dual-stack. The comparison between the two protocols indicates that IPv4 performed better than IPv6.

Another parameter that affect the performance of a voice application is the end-to-end delay, it refers to the time taken for a voice packet to be transmitted across a network from source to destination. The average values of the VOIP end to end delay was monitored and the results were shown in the Figure 13. The measurements indicate that dual-stack performed better than manual and automatic 6to4. The comparison between the two protocols shows that IPv4 performs better than IPv6 in this case.

Another indicator about the quality of a voice application is the Mean Opinion Score (MOS). MOS is a test that has been used in telephony networks to obtain the human user’s view of the quality of the network. MOS is a subjective measurement. It is computed by asking a group of listeners to sit in a quiet room and score call quality as they perceived by giving it one of these values (1,2,3,4,5), where 5 indicates an excellent quality and 1 indicates a bad quality. After that the average of the scored values was computed.

OPNET has the ability to monitor MOS value and compute its average. In this experiment, the average MOS values were monitored and the results were depicted in Figure 14. The higher MOS value indicates a better performance. The results indicates that dual-stack performed better than the other two

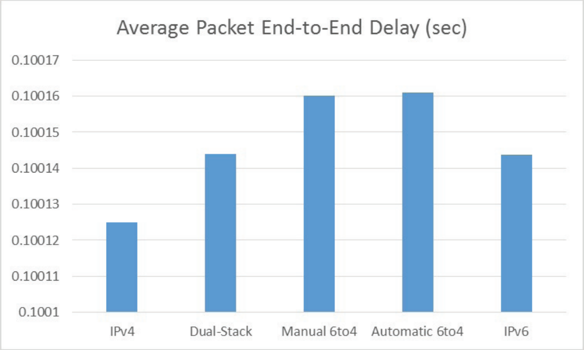

Fig. 13: Voice Packet End-to-End Delay

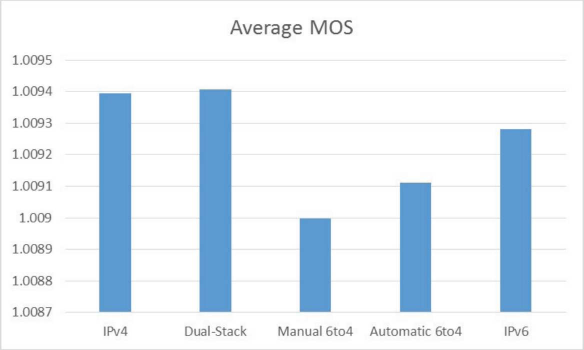

Fig. 14: VOIP MOS

mechanisms, automatic 6to4 performs better than manual 6to4 and IPv4 had better voice quality than IPv6.

VI. Conclusion

Dual-stack is easy to implement but network devices must support both IPv4 and IPv6 protocols and it can be used to send packets between IPv4 networks or IPv6 networks. On the other hand, the tunneling transition mechanism is a good choice for networks that have devices that do not yet support IPv6.

Although dual-stack performed better than tunneling in most cases. However for very large network with a large number of routers dual-stack needs a lot of work to configure since it require to configure each router with both IPv4 and IPv6 protocols, while tunneling just needs to configure the two ends of the network with both IPv4 and IPv6 protocols, thus we have to choose between manual and automatic tunneling according to the finding results in each application.

Compared to pure IPv4 and IPv6 networks in most cases dual-stack had more response time; this is due to the both IP stacks processing simultaneously and packets have to flow in two different IP stacks, which caused slight more delay.

The future work could focus on adding other transition mechanisms (such as ISATAP, Teredo tunneling) to the analysis and study their impact on various applications performance. Other future research could be on adding more analysis about the reasons that causes the differences between the transition mechanisms regarding different applications performance in order to suggest modifications on these techniques.

References

[1] S. E. Deering and R. M. Hinden, “Internet protocol, version 6 (ipv6) specification,” RFC2460, December 1998.

[2] IEEEUSA, “Next generation internet: Ipv4 address exhaustion, mitigation strategies and implications for the u.s.” 2009, an IEEE-USA White Paper.

[3] D. Punithavathani and K.Sankaranarayana, “Ipv4/ipv6 transition mechanisms,” *European Journal of Scientific Research*, vol. 34, no. 1, pp. 110–124, 2009.

[4] E. Nordmark and R. E. Gilligan, “Basic transition mechanisms for ipv6 hosts and routers,” RFC4213, October 2005.

[5] G. Tsirtsis and P. Srisuresh, “Network address translation - protocol translation (nat-pt),” RFC2766, February 2000.

[6] C. Aoun and E. B. Davies, “Reasons to move the network address translator - protocol translator (nat-pt) to historic status,” RFC4966, July 2007.

[7] D. G. Waddington and F. Chang, “Realizing the transition to ipv6,” *IEEE Communications Magazine*, vol. 40, no. 6, pp. 138 – 147, June 2002.

[8] C. Huitema, R. Austein, S. Satapati, and R. van der Pol, “Evaluation of ipv6 transition mechanisms for unmanaged networks,” RFC3904, September 2004.

[9] I. Raicu and S. Zeadally, “Evaluating ipv4 to ipv6 transition mechanisms,” in *10th International Conference on Telecommunications, ICT 2003*, vol. 2. IEEE, 23 Feb.–1 March 2003, pp. 1091–1098.

[10] S. Narayan and S. Tauch, “Ipv4-v6 transition mechanisms network performance evaluation on operating systems,” in *3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, vol. 5. Chengdu: IEEE, 9-11 July 2010, pp. 664 – 668.

[11] P. Grayeli, S. Sarkani, and T. Mazzuchi, “Performance analysis of ipv6 transition mechanisms over mpls,” *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 4, no. 2, pp. 91–103, August 2012. [Online]. Available: http://www.ijcnis.org/index.php/ijcnis/article/view/155

[12] A. Albkerat and B. Issac, “Analysis of ipv6 transition technologies,” *International Journal of Computer Networks & Communications (IJCNC)*, vol. 6, no. 5, pp. 19–38, September 2014. [Online]. Available: http://dblp.uni-trier.de/rec/bib/journals/corr/Albkerat014

[13] N. Bahaman, E. Hamid, and A. Prabuwono, “Network performance evaluation of 6to4 tunneling,” in *International Conference on Innovation Management and Technology Research ((ICIMTR,)*. Malacca: IEEE, May 2012, pp. 263–268.

[14] M. Aazam, I. Khan, M. Alam, and A. Qayyum, “Comparison of ipv6 tunneled traffic of teredo and isatap over test-bed setup,” in *International Conference on Information and Emerging Technologies (ICIET)*. Karachi: IEEE, 14-16 June 2010, pp. 1–4.

[15] P. Wu, Y. Cui, J. Wu, J. Liu, and C. Metz, “Transition from ipv4 to ipv6: A state-of-the-art survey,” *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1407–1424, December 2013.

[16] H. Sathu and M. A. Shah, *Performance Analysis of Video Protocols over IP Transition Mechanisms*. Springer Berlin Heidelberg, 2011, vol. 193, ch. Advances in Computing and Communications, pp. 491–500.

[17] M. N. Ismail, M. A. M. Shukran, K. Maskat, , and M. H. Mohamed, “File transfer over dual-stack ipv6 tunnelling in real network environment: Router to router performance analysis using best effort approach,” *International Journal of Multidisciplinary Sciences and Engineering*, vol. 4, no. 8, pp. 18–24, September 2013.

[18] C. Huitema, “An anycast prefix for 6to4 relay routers,” RFC3068, June 2001.

[19] O. Troan and B. C. (editor), “Deprecating the anycast prefix for 6to4 relay routers,” RFC7526, May 2015.

[20] T. Berners-Lee, R. T. Fielding, and H. F. Nielsen, “Hypertext transfer protocol – http/1.0,” RFC1945, May 1996.

[21] J. Postel, “File transfer protocol,” RFC959, October 1985.

[22] B. Goode, “Voice over internet protocol (voip),” *IEEE*, vol. 90, no. 9, pp. 1495 – 1517, September 2002.