Modern Applied Science; Vol. 8, No. 3; 2014
ISSN 1913-1844 E-ISSN 1913-1852
Published by Canadian Center of Science and Education

Study on Migration from IPv4 to IPv6 of a Large Scale Network

Muhammad Yeasir Arafat[1], M Abdus Sobhan[1] & Feroz Ahmed[1]

[1] Department of Electrical and Electronic Engineering, School of Engineering and Computer Science, Independent University, Bangladesh

Correspondence: Muhammad Yeasir Arafat, School of Engineering and Computer Science, Independent University, Bangladesh. Tel: 88-017-1930-1525. E-mail: yeasir08@yahoo.com

Received: February 11, 2014 Accepted: April 1, 2014 Online Published: April 8, 2014

doi:10.5539/mas.v8n3p67 URL: http://dx.doi.org/10.5539/mas.v8n3p67

Abstract

This work mainly addresses the design a large scale network using dual stack mechanisms. We concentrated on the most imperative theoretical notions of the IPv6 protocol, such as IP addressing, address allocation, routing with the OSPF and BGP protocols and routing protocols performance in dual stack network using GNS3 simulations and Wireshark Network protocol analyzer. It is evaluate a real large-scale network environment utilizing accessible end-to-end estimation methods that focuses on a large-scale IPv4 and IPv6 backbone and made performance the IPv4 and IPv6 network. In this paper, we compiled IPv6 address planning in a large scale network, performance metrics of each network in terms of time sequence graph, round trip time, TCP throughput, TCP connection time and the number of TCP connections per second that a client can establish with a remote server. It is found that, a minor degradation in the throughput of the TCP, TCP response time and a lower packet loss rate are arise in a real large scale dual stack network. We also showed a concise case study on relationship between RTT and network topology, which is cooperative to develop the performance of IPv6 networks. The result shows the proposed scheme for network migration from IPv4 to IPv6 is more reliable than other existing schemes.

Keywords: IPv6, IPv4, double stack, simulators, performance measurement, BGPv4, OSPFv3, migration, ISP

1. Introduction

Currently, the Internet consists of native IPv4, native IPv6, and IPv4/IPv6 dual stack networks. Unfortunately, IPv4 and IPv6 are unsuited protocols. When both IP versions are available and the users of Internet desire to connect without any limitations, a transition mechanism is mandatory. During the occasion of migration from IPv4 to IPv6 networks, a number of transition mechanisms have been suggested by IETF to ensure smooth, stepwise and independent changeover. The conception of transiting from IPv4 mesh to IPv6 mesh is being processed strongly. The transition between IPv4 internet and IPv6 will be a long procedure as they are two completely separate protocols. IPv6 is not backward well-suited with IPv4, and IPv4 hosts and routers will not be adept to deal directly with IPv6 traffic and vice-versa. As IPv4 and IPv6 will co-exist for a long time, this wants the transition and inter-operation mechanism. Due to this cause several transitions mechanisms have been developed that can be used to build the transition to IPv6 efficiently. Most of the applications today support IPv4 and therefore there is a need to run these applications on IPv6 access network, especially to persons who are generally on mobile and they want to securely connect to their home network so as to reach IPv4 services. IPv6 is developed as a network layer protocol, overcoming the problems in IPv4. Its 128-bit address format considerably enlarges the address space and will gratify the address demands for a fairly long time. Although, IPv6 has no built-in backwards compatibility with IPv4, which means IPv6 networks cannot correspond with IPv4 in environment. Competently IPv6 has considered a parallel, independent network that coexists with its support IPv4. IPv6-supported applications and IPv6-accessible contents are still the marginal; the majority of network resources, services and applications still remain in IPv4. A number of transition techniques are existing to maintain the connectivity of both IPv4 and IPv6, to accomplish inter connection between IPv4 and IPv6, and also support the adoption process of IPv6. Vendors expect to invest on implementing well-developed transition techniques, so that their products can have good capability and bring high profits. As for internet service providers (ISP), they require to find a way to provide the existing services for both IPv4 and IPv6 users, and organize their services with an expected deployment of transition techniques on the Internet.

In this paper first, present an IP address planning and network design for dual stack network which encompass

IPv4 and IPv6 address simultaneously. A large scale network implemented with help of advance dynamic routing protocols. Second, investigate the different behaviors of IPv6 performance on a path-by-path basis over time performance. In this paper a virtual study of the performance of IPv4-only network with that of dual stack transition mechanism (DSTM) under various types of traffic patterns is accepted out. In the proposed DSTM enabled network architecture, the hosts in IPv4 network commences connection with hosts in the IPv4 network over an integrated IPv4/IPv6 network. The analysis makes use of Wireshark graphing capabilities to showed round trip time for ACKs overt time known as a round trip time graph, transmission throughput using TCP sequence numbers called throughput graph, sequence number versus time graphs that help to see if traffic is moving along without interruption, packet loss, or long delays called as time sequence graph Stevens and tcptrace.

2. Transition Method

Since there is such a large difference between IPv4 and IPv6, they cannot communicate directly with each other. A system that is capable of handling IPv6 traffic can be made backward compatible, but an already deployed system that handles only IPv4 is not able to handle IPv6 datagram. This means that a major upgrade process would need to take place, involving hundreds of millions of machines, in order to make a complete transition to IPv6. This way is too expensive and time consuming and in any case will not happen overnight. The network world will most likely see a gradual transition to IPv6, where IPv6 will be integrated into the IPv4 world that exists today. There are different kinds of technologies which can be applied such as dual stack, tunnelling, and translation. At present, three transition mechanisms, dual stack, tunneling, and translation, are proposed to solve the problems due to the co-existence of IPv6 and IPv4. Over several transition techniques have been used and tested for the communications between different networks to ensure IPv4 and IPv6 interoperability. Therefore, to make decision on the best suited transition methods, it is really important to have an overview of the current IPv4 networks. In addition, enterprises must analyze needed functionalities, scalability, and securities in the corporation.

2.1 Dual Stack

The Dual Stack technique is entitled as native dual stack or dual IP layer. Both protocols IPv4 and IPv6 run parallel on the similar network infrastructure which does not necessitate encapsulation of IPv6 interior IPv4 and vice versa. A common dual-stack migration approach as shown in Figure 1 makes a transition from the core to the edge. This includes enabling two TCP/IP protocol stacks on the WAN core routers. In a common dual stack migration firstly the edge routers, and firewalls, then the server-farm switches and finally the desktop access routers. Once the network supports IPv6 and IPv4 protocols, the process will enable dual protocol stacks on the servers and then the edge entities. The dual stack doubles the communication requirements, which in turn causes performance degradation. The dual stack method is literally to use two IPv4 and IPv6 stacks for operating simultaneously, which enables apparatus to run on either protocol, according to accessible services, network availability, and administrative policies. This can be accomplished in both end systems and network devices. As a result, IPv4 enabled programs use IPv4 stack and this goes the identical for IPv6.

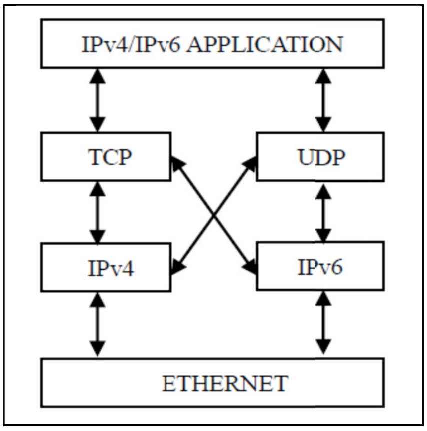


Figure 1. Dual stack mechanism

The IP header version field would play an important role in receiving and sending packets. In other words, this kind of IPv6 transition is the encapsulation of IPv6 within IPv4. The complete transition can be managed by domain name system (DNS).

2.2 Tunnelling

Tunnelling, from the insight of transitioning, enables unsuited networks to be bridged and it is usually applied in a point-to-point or sequential manner of a network. Three mechanisms of tunnelling are offered: IPv6 over IPv4, IPv6 to IPv4 automatic tunnelling and tunnel broker. Tunnelling IPv6 traffic over an IPv4 network is another possibility. This approach allows the IPv6 traffic to be encapsulated in an IPv4 packet and forwarded, creating an IPv6 tunnel over the IPv4 infrastructure. A tunnel can be created as a solution for transporting IPv6 traffic, from IPv6 node to the destination IPv6 node, over the IPv4-only network. A “virtual link” is created and, from the perspective of the two establishing IPv6 nodes, this appears as a point-to-point link. The different types of tunnelling techniques can be categorized into two types: manually configured and automatic tunnelling. A point-to-point link has to be manually configured, as the name suggests. For automatic tunnelling, an IPv6 node can dynamically tunnel packets by using a 6 to 4 address. This is used to transfer data between compatible networking nodes over incompatible networks. There are two ordinary scenarios to apply tunnelling: the allowance of end systems to apply off link transition devices in a distributed network and the act of enabling edge devices in networks to inter-connect over incompatible networks.

2.3 Translation

The meaning of translation is to convert directly protocols from IPv4 to IPv6 or vice versa, which might result in transforming those two protocol headers and payload. This mechanism can be established at layers in protocol stack, consisting of network, transport, and application layers. The translation method has many mechanisms, which can be either stateless or stateful. While stateless means that the translator can perform every conversion separately with no reference to previous packets, stateful is the vice versa, which maintains some form of state in regard to previous packets. The translation process can be conducted in either end systems or network devices. The fundamental part of translation mechanism in transition process is the conversion of IP and ICMP packets. All translation methods, which are used to establish communication between IPv6-only and IPv4-only hosts, for instance, NAT-PT or BIS, apply an algorithm known as stateless IP/ICMP translator (SIIT).

3. Design a Large Scale Network in Dual Stack

In this paper, a large scale network is design based on dual stack network. This dual stack network is designed for a nationwide ISP. Design considerations are given below:

3.1 Topology Design

In this paper our designed ISP has 4 main operating area or region. Each region has 2 small POP. Each region network has one data centre to host content. Regional network are inter-connected with multiple link.

(i) *Regional Network*

Each regional network has three routers. One core and two edge routers, point of presence (POP) for every region. POP will use a router to terminate customer network, i.e. edge router. Each POP is an aggregation point of ISP customer.

(ii) *Design Consideration*

Each regional network should have address summarization capability for customer block and CS link WAN. Prefix planning should have scalability option for next couple of years for both customer block and infrastructure. No Summarization require for infrastructure WAN and loopback address. All WAN links should be ICMP reachable for link monitoring purpose. Conservation will get high preference for IPv4 address planning and aggregation will get high preference for IPv6 address planning. OSPF is running in ISP network to carry infrastructure IP prefix. Each region is a separate open shortest path first (OSPF) area. Transport core is in OSPF area 0. Customer will connect on either static or external border gateway protocol (eBGP). Internal border gateway protocol (iBGP) will carry external prefix within ISP core IP network.

(iii) *IPv6 Address Plan Consideration*

Big IPv6 address space can cause very large routing table size. Most transit service provider apply IPv6 aggregation prefix filter (i.e. anything other than /48 &<=/32 prefix size. Prefix announcement need to send to internet should be either /32 or /48 bit boundary IPv6 address plan consideration (RFC3177). WAN link can be used on /64 bit boundary. End site/customer sub allocation can be made between /48~/64 bit boundary. APNIC utilization/HD ratio will be calculated based on /56 end site assignment/sub-allocation.

(iv) *Addressing Plans – ISP Infrastructure*

• *Point-to-Point links:* Protocol design expectation is that /64 is used (/127 now recommended/standardized (rfc6164)). (Reserve /64 for the link, but address it as a /127).

• *Other options:* A /126s are being used (mirrors IPv4 /30). A /112s are being used. For node ID we have leaves free final 16 bits. There are some discussion about /80s, /96s and /120s.

• *ISPs should receive /32 from their RIR:* Normally address block for router loop-back interfaces a number all loopbacks out of one /48 address. We have used /128 IP address for loopback. Address block for infrastructure /48 network block allow 65k subnets. A /48 network address per region, this is for the largest international networks. A /48 network address for whole backbone. Customers get one /48 block network address. Unless they have more than 65k subnets inwhich case they get a second /48 (and so on). In typical deployments today Several ISPs give small customers a /56 or single LAN end-sites a /64, e.g.: /64 if end-site will only ever be a LAN. A /56 block network for medium end-sites (e.g. small business) and a /48 network block for large end-sites. Registries will regularly assign the next block to be contiguous with the first allocation, minimum allocation is /32. Very expected that subsequent allocation block will make this up to a /31.

• IPv6 Allocation Form registry is: 2406:6400::/32.

• IPv4 Allocation From registry is: 172.16.0.0/19.

3.2 Address Plan for IPv6

For the address planning we followed RFC 3849, which is IPv6 address prefix reserved for documentation. Details IP address plan for IPv6 is given below (Tables 1 to 8):

Table 1. Top level distribution infrastructure and customers

Block	Prefix	Description
1	2406:6400::/32	Parent Block
2	2406:6400:0000:0000::/36	Infrastructure
	2406:6400:1000:0000::/36	
	2406:6400:2000:0000::/36	
	2406:6400:3000:0000::/36	
	2406:6400:4000:0000::/36	
	2406:6400:5000:0000::/36	
	2406:6400:6000:0000::/36	
	2406:6400:7000:0000::/36	
3	2406:6400:8000:0000::/36	Customer network Region 1
	2406:6400:9000:0000::/36	
4	2406:6400:a000:0000::/36	Customer network Region 2
	2406:6400:b000:0000::/36	
5	2406:6400:c000:0000::/36	Customer network Region 3
	2406:6400:d000:0000::/36	
6	2406:6400:e000:0000::/36	Customer network Region 4
	2406:6400:f000:0000::/36	

Table 2. Summarization option infrastructure and customers

Block	Prefix	Description
7	2406:6400:8000:0000::/35	CS net summery region1 [R2]
8	2406:6400:a000:0000::/35	CS net summery region2 [R5]
9	2406:6400:c000:0000::/35	CS net summery region3 [R8]
10	2406:6400:e000:0000::/35	CS net summery region4 [R11]

Table 3. Details distribution infrastructure

Block	Prefix	Description
2	2406:6400:0000:0000::/36	Infrastructure
11	2406:6400:0000:0000::/40	Loopback, Transport & WAN
	2406:6400:0100:0000::/40	
	2406:6400:0200:0000::/40	
	2406:6400:0300:0000::/40	
	2406:6400:0400:0000::/40	
	2406:6400:0500:0000::/40	
	2406:6400:0600:0000::/40	
	2406:6400:0700:0000::/40	
16	2406:6400:0800:0000::/40	R2 DC
	2406:6400:0900:0000::/40	
17	2406:6400:0a00:0000::/40	R5 DC
	2406:6400:0b00:0000::/40	
18	2406:6400:0c00:0000::/40	R8 DC
	2406:6400:0d00:0000::/40	
19	2406:6400:0e00:0000::/40	R11 DC
	2406:6400:0f00:0000::/40	

Table 4. Details loopback address

Block	Prefix	Description
20	2406:6400:0000:0000::/48	Loopback
43	2406:6400:0000:0000::1/128	R1 loopback 0
44	2406:6400:0000:0000::2/128	R2 loopback 0
45	2406:6400:0000:0000::3/128	R3 loopback 0
46	2406:6400:0000:0000::4/128	R4 loopback 0
47	2406:6400:0000:0000::5/128	R5 loopback 0
48	2406:6400:0000:0000::6/128	R6 loopback 0
49	2406:6400:0000:0000::7/128	R7 loopback 0
50	2406:6400:0000:0000::8/128	R8 loopback 0
51	2406:6400:0000:0000::9/128	R9 loopback 0
52	2406:6400:0000:0000::10/128	R10 loopback 0
53	2406:6400:0000:0000::11/128	R11 loopback 0
54	2406:6400:0000:0000::12/128	R12 loopback 0

Table 5. Summarization customers block region 1

Block	Prefix	Description
	2406:6400:8000:0000::/35	Customer block region 1 [R2]
	2406:6400:8000:0000::/37	Customer block POP1 [R1]
	2406:6400:8800:0000::/37	Customer block future use/POP
	2406:6400:9000:0000::/37	Customer block future use/POP
	2406:6400:9800:0000::/37	Customer block POP2 [R3]

Table 6. Summarization customers block region 2

Block	Prefix	Description
	2406:6400:A000:0000::/35	Customer block region 2 [R5]
	2406:6400:A000:0000::/37	Customer block POP1 [R4]
	2406:6400:A800:0000::/37	Customer block future use/POP
	2406:6400:B000:0000::/37	Customer block future use/POP
	2406:6400:B800:0000::/37	Customer block POP2 [R6]

Table 7. Summarization customers block region 3

Block	Prefix	Description
	2406:6400:c000:0000::/35	Customer block region 3 [R8]
	2406:6400:C000:0000::/37	Customer block POP1 [R7]
	2406:6400:C800:0000::/37	Customer block future use/POP
	2406:6400:C000:0000::/37	Customer block future use/POP
	2406:6400:C800:0000::/37	Customer block POP2 [R9]

Table 8. Summarization customers block region 4

Block	Prefix	Description
	2406:6400:e000:0000::/35	Customer block region 4 [R11]
	2406:6400:E000:0000::/37	Customer block POP1 [R10]
	2406:6400:E800:0000::/37	Customer block future use/POP
	2406:6400:E000:0000::/37	Customer block future use/POP
	2406:6400:E800:0000::/37	Customer block POP2 [R12]

3.3 Address Plan for IPv4

IP address Plan for IPv4 is given below (Table 9 to 12):

Table 9. Parent block IPv4

Block	Prefix	Size	Description
1	172.16.0.0	/19	Parent Block
2	172.16.0.0	/20	Infrastructure
3	172.16.16.0	/20	Customer Network

Table 10. Details infrastructure WAN block IPv4

Block	Prefix	Size	Description
12	172.16.10.0	/24	WAN Prefix
13	172.16.10.0	/30	Router 2-1 WAN
14	172.16.10.4	/30	Router 2-3 WAN
15	172.16.10.8	/30	Router 1-3 WAN
16	172.16.10.24	/30	Router 5-4 WAN
17	172.16.10.28	/30	Router 5-6 WAN
18	172.16.10.32	/30	Router 4-6 WAN
19	172.16.10.48	/30	Router 8-7 WAN
20	172.16.10.52	/30	Router 8-9 WAN
21	172.16.10.56	/30	Router 7-9 WAN
22	172.16.10.72	/30	Router 11-10 WAN
23	172.16.10.76	/30	Router 11-12 WAN
24	172.16.10.80	/30	Router 10-12 WAN

Table 11. Details infrastructure block transport & loopback IPv4

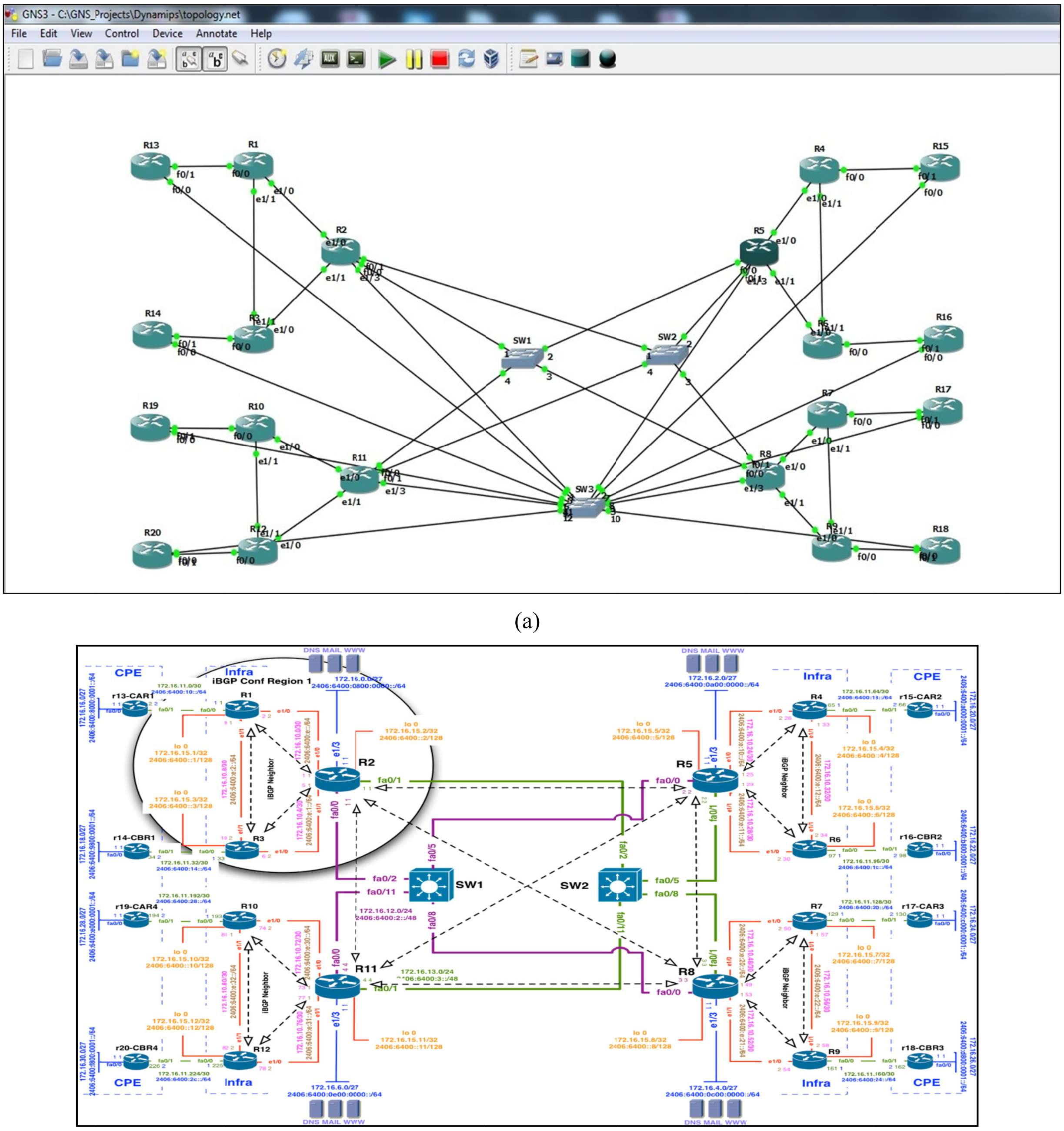
Block	Prefix	Size	Description
25	172.16.12.0	/24	Transport link
26	172.16.13.0	/24	Transport link
27	172.16.15.0	/24	Loopback

Table 12. Details customer IPv4 block

Block	Prefix	Size	Description
28	172.16.6.0	/20	Customer network
29	172.16.16.0	/22	Router 2 summary net
30	172.16.16.0	/23	Router 1 CS network
31	172.16.18.0	/23	Router 3 CS network
32	172.16.20.0	/22	Router 5 summary net
33	172.16.20.0	/23	Router 4 CS network
34	172.16.22.0	/23	Router 6 CS network
35	172.16.24.0	/22	Router 8 summary net
36	172.16.24.0	/23	Router 7 CS network
37	172.16.26.0	/23	Router 9 CS network
38	172.16.28.0	/22	Router 11 summary net
39	172.16.28.0	/23	Router 10 CS network
40	172.16.30.0	/23	Router 12 CS network

4. Simulation and Analysis

Simulations are performed using GNS3. For the simulation we used Cisco 7200 series router. A server contain core i7 processor with 8GB RAM used for run the simulation.



(a)

(b)

Figure 2. (a) Simulation network in GNS 3 (b) iBGP peering For Region 1 simulation Network in GNS 3

In the test bed dual stack network, an ISP has 4 main operating area or region. Each region has 2 small POP. POP will use a router to terminate customer network. GNS 3 topology is given in Figure 2(a).

4.1 Network Connection Pattern

Before enabling OSPF3 on an Interface, the following steps must be done on a Router. Enable IPv6 unicast routing and enable IPv6 CEF. In region 1 router R1, R2, R3 have iBGP peering with other networks showed in Figure 2(b).

In the same way R5, R4, R6 have iBGP peering with others in region 2. R8, R7, R9 have iBGP peering with others in region 3. R11, R10, R12 have iBGP peering with others in region 4.

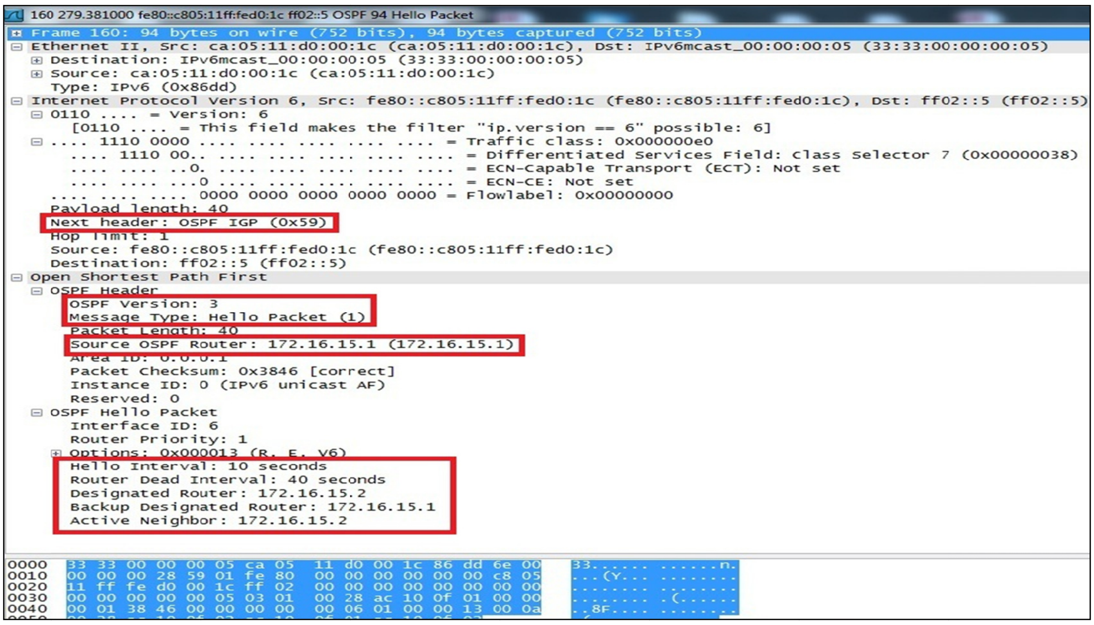
4.2 Output and Analysis

For analysis routing protocols we used Wireshark. This paper contains details analysis of wireshark traces for investigating the behaviors of TCP congestions control mechanism. Packets are captured from the routers interface. It is discussed some measurement from the topology with necessary figures. The papers will analysis the following facets:

- Basic slow start and avoidance mechanisms in IP network.
- Variation of the TCP slow starts mechanism that uses fast retransmit followed by congestion avoidance.
- Receiver-advertised flow control mechanisms
- Round trip time and Throughput of the connections.

4.2.1 OSPFv3 Packet Analysis

There are two types of changes which may occur in the topology: link status changes and link weight changes. The OSPF protocol detects link status changes via the HELLO protocol. The Hello packet encloses no address information at all. Rather, it contains an Interface ID that the originating router has allottedto uniquely identify (among its own interfaces) its interface to the link. This Interface ID will be used as the network-LSA's link state ID if the router becomes the designated router on the link. The OSPF packet header now embraces an "Instance ID" that permits multiple OSPF protocol illustrations to be run on a single link. The neighbour arrangement performs the same function in both IPv6 and IPv4. Specifically, it collects all information mandatory to form an adjacency between two routers when such an adjacency becomes essential. When the OSPF protocol detects a change in the topology, it creates new LSAs appropriate for the cause and floods them throughout the OSPF domain. As the new LSAs are flooded they are accounted for in the "OSPF caused OSPF updates" statistic.



```
160 279.381000 fe80::c805:11ff:fed0:1c ff02::5 OSPF 94 Hello Packet
⊞ Frame 160: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)
⊟ Ethernet II, Src: ca:05:11:d0:00:1c (ca:05:11:d0:00:1c), Dst: IPv6mcast_00:00:00:05 (33:33:00:00:00:05)
  ⊞ Destination: IPv6mcast_00:00:00:05 (33:33:00:00:00:05)
  ⊞ Source: ca:05:11:d0:00:1c (ca:05:11:d0:00:1c)
    Type: IPv6 (0x86dd)
⊟ Internet Protocol Version 6, Src: fe80::c805:11ff:fed0:1c (fe80::c805:11ff:fed0:1c), Dst: ff02::5 (ff02::5)
  ⊟ 0110 .... = Version: 6
      [0110 .... = This field makes the filter "ip.version == 6" possible: 6]
  ⊟ .... 1110 0000 .... .... .... .... .... = Traffic class: 0x000000e0
      .... 1110 00.. .... .... .... .... .... = Differentiated Services Field: Class Selector 7 (0x00000038)
      .... .... ..0. .... .... .... .... .... = ECN-Capable Transport (ECT): Not set
      .... .... ...0 .... .... .... .... .... = ECN-CE: Not set
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 40
    Next header: OSPF IGP (0x59)
    Hop limit: 1
    Source: fe80::c805:11ff:fed0:1c (fe80::c805:11ff:fed0:1c)
    Destination: ff02::5 (ff02::5)
⊟ Open Shortest Path First
  ⊟ OSPF Header
      OSPF Version: 3
      Message Type: Hello Packet (1)
      Packet Length: 40
      Source OSPF Router: 172.16.15.1 (172.16.15.1)
      Area ID: 0.0.0.1
      Packet Checksum: 0x3846 [correct]
      Instance ID: 0 (IPv6 unicast AF)
      Reserved: 0
  ⊟ OSPF Hello Packet
      Interface ID: 6
      Router Priority: 1
    ⊞ Options: 0x000013 (R, E, V6)
      Hello Interval: 10 seconds
      Router Dead Interval: 40 seconds
      Designated Router: 172.16.15.2
      Backup Designated Router: 172.16.15.1
      Active Neighbor: 172.16.15.2

0000  33 33 00 00 00 05 ca 05  11 d0 00 1c 86 dd 6e 00   33...... ......n.
0010  00 00 00 28 59 01 fe 80  00 00 00 00 00 00 c8 05   ...(Y... ........
0020  11 ff fe d0 00 1c ff 02  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 05 03 01  00 28 ac 10 0f 01 00 00   ........ .(......
0040  00 01 38 46 00 00 00 00  00 06 01 00 00 13 00 0a   ..8F.... ........
```

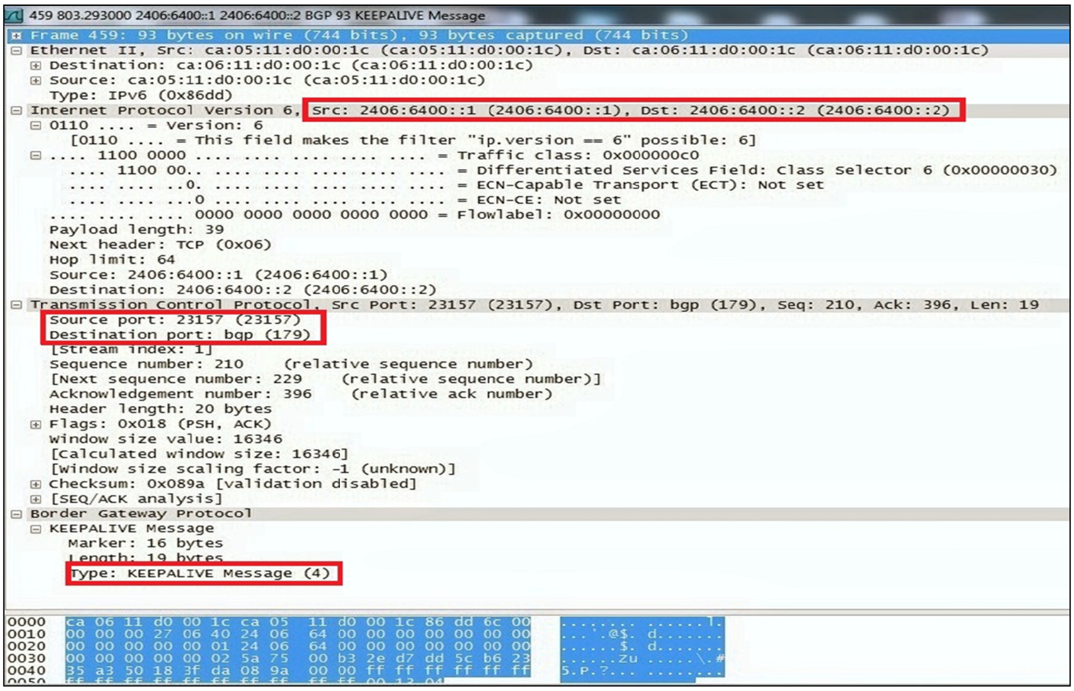
Figure 3. OSPF Hello packet

Each neighbor arrangement is hop to a single OSPF interface. The Interface ID that the neighbor advertises in its Hello packets should be traced in the neighbor structure. The router will include the neighbor's Interface ID in the router's router-LSA when either a) advertising a point-to-point or point-to-multipoint link to the neighbor or b) advertising a link to a network where the neighbor has become the designated router. In IPv6 OSPF sprints directly over IPv6's network layer. When, it is encapsulated in one or more IPv6 headers with the Next Header field of the immediately encapsulating IPv6 header set to the value 89. In Figure 3 we have shown OSPF v3 hello packet which source OSPF route from 172.16.15.1. This designated route from 172.16.15.2 and backup designated route from 172.16.15. This is OSPF hello packet, which next header is OSPF IGP. Hello packet interval time duration is 10s. OSPF caused BGP updates are measured when the connection between two iBGPpeers changes. This signals a change in the underlying OSPF network between the peers, and so the cause of the subsequent updates is attributed to the OSPF protocol. All routing protocols create overhead when performing routing; often this is routing traffic overhead when exchanging information with other routers, this routing traffic is necessary for protocol operation. However in certain situations such as a rapidly changing network this traffic can come to consume large amounts of available bandwidth and be detrimental to the network throughput. Link state protocols such as OSPF are more complicated than distance vector protocols and create extra overhead in the form of bandwidth, memory and CPU usage in order to calculate and store the routing tables, in smaller networks this leads to EIGRP being more efficient. When used in larger networks. OSPFs hierarchical nature gives an advantage over EIGRP when used with properly configured areas in order to limit routing overhead. Simulations performed by found that IGP has better bandwidth utilization and lower protocol traffic than OSPF.

OSPFv3 packets are transmitted in IPv4 datagram's with a protocol identifier equals to 89. That is, OSPF does not use TCP or UDP in the transport layer. OSPF packets have a common header plus a specific part, resulting in five different types of packets: Hello (Type-1), Database Description (Type-2), Link-State Request (Type-3), Link-State Update (Type-4), and Link-State Acknowledgment (Type-5).

4.2.2 BGP Packet Analysis

BGP is the path-vector protocol that is work on exchanging external AS routing information and operates level of address blocks or AS prefixes. BGP routers exchange routing information using open, update, notification and keepalive message.A snippet of captured date showing update and keepalive message is shown in Figure 4 captured BGP traffic is from the TCP port 179. Figure 4 illustrates frame number 138 and the number of bytes captured on this frame. As messages originate from multiple protocols, the frame shows Ethernet protocol source and destination address, the source and destination addresses of IP, source and destinations port numbers for TCP and details of BGP. In Figure 4 we have shown BGP keep alive message from 2406:6400::1 to 2406:6400::2.



```
459 803.293000 2406:6400::1 2406:6400::2 BGP 93 KEEPALIVE Message
⊞ Frame 459: 93 bytes on wire (744 bits), 93 bytes captured (744 bits)
⊟ Ethernet II, Src: ca:05:11:d0:00:1c (ca:05:11:d0:00:1c), Dst: ca:06:11:d0:00:1c (ca:06:11:d0:00:1c)
  ⊞ Destination: ca:06:11:d0:00:1c (ca:06:11:d0:00:1c)
  ⊞ Source: ca:05:11:d0:00:1c (ca:05:11:d0:00:1c)
    Type: IPv6 (0x86dd)
⊟ Internet Protocol Version 6, Src: 2406:6400::1 (2406:6400::1), Dst: 2406:6400::2 (2406:6400::2)
  ⊟ 0110 .... = Version: 6
      [0110 .... = This field makes the filter "ip.version == 6" possible: 6]
  ⊟ .... 1100 0000 .... .... .... .... .... = Traffic class: 0x000000c0
      .... 1100 00.. .... .... .... .... .... = Differentiated Services Field: Class Selector 6 (0x00000030)
      .... .... ..0. .... .... .... .... .... = ECN-Capable Transport (ECT): Not set
      .... .... ...0 .... .... .... .... .... = ECN-CE: Not set
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 39
    Next header: TCP (0x06)
    Hop limit: 64
    Source: 2406:6400::1 (2406:6400::1)
    Destination: 2406:6400::2 (2406:6400::2)
⊟ Transmission Control Protocol, Src Port: 23157 (23157), Dst Port: bgp (179), Seq: 210, Ack: 396, Len: 19
    Source port: 23157 (23157)
    Destination port: bgp (179)
    [Stream index: 1]
    Sequence number: 210    (relative sequence number)
    [Next sequence number: 229    (relative sequence number)]
    Acknowledgement number: 396    (relative ack number)
    Header length: 20 bytes
  ⊞ Flags: 0x018 (PSH, ACK)
    Window size value: 16346
    [Calculated window size: 16346]
    [Window size scaling factor: -1 (unknown)]
  ⊞ Checksum: 0x089a [validation disabled]
  ⊞ [SEQ/ACK analysis]
⊟ Border Gateway Protocol
  ⊟ KEEPALIVE Message
      Marker: 16 bytes
      Length: 19 bytes
      Type: KEEPALIVE Message (4)

0000  ca 06 11 d0 00 1c ca 05  11 d0 00 1c 86 dd 6c 00   ........ ......l.
0010  00 00 00 27 06 40 24 06  64 00 00 00 00 00 00 00   ...'.@$. d.......
0020  00 00 00 00 00 01 24 06  64 00 00 00 00 00 00 00   ......$. d.......
0030  00 00 00 00 00 02 5a 75  00 b3 2e d7 dd 5c b6 23   ......Zu .....\.#
0040  35 a3 50 18 3f da 08 9a  00 00 ff ff ff ff ff ff   5.P.?... ........
0050  ff ff ff ff ff ff ff ff  ff ff 00 13 04
```

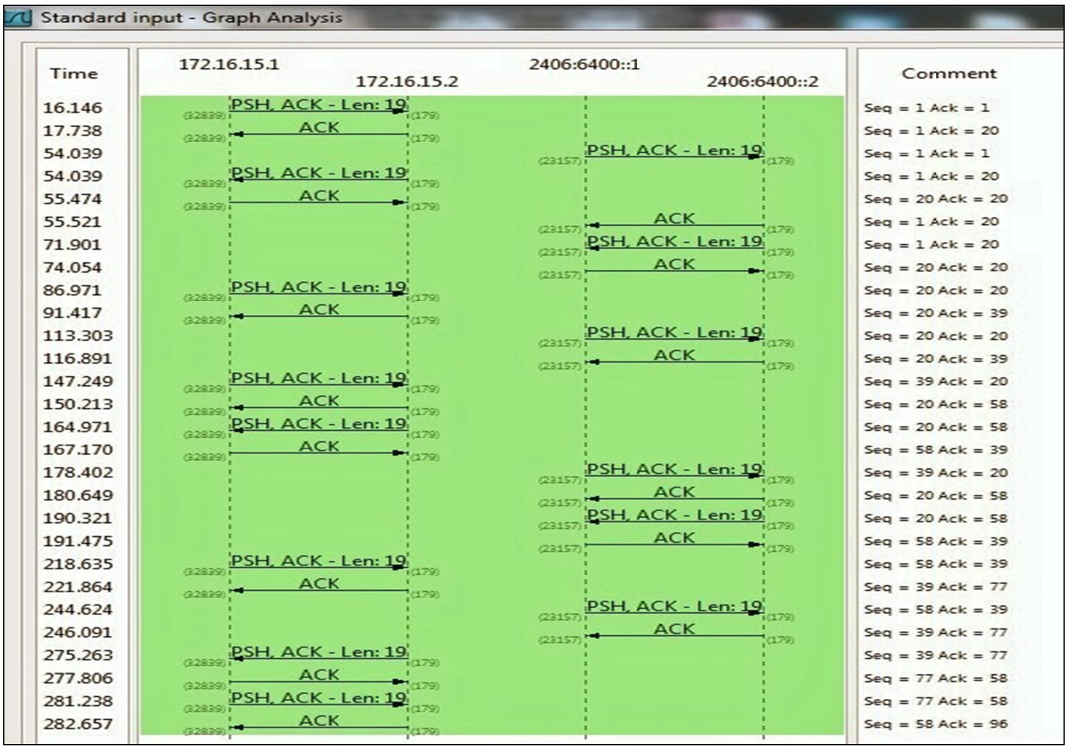
Figure 4. BGP Packet analysis

BGP operates over a Transmission Control Protocol (TCP) as a transport protocol (port number 179). TCP has an advantage over User Datagram Protocol (UDP) connections: BGP does not need to implement fragmentation, retransmission, acknowledgment, and sequencing. BGP has the capability to support Classless Inter-Domain Routing (CIDR) in order to reduce the size of the Internet routing tables. CIDR allows routers to group routes together in order to minimize the number of routing information carried by the core routers, which makes it a dominant Internet routing protocol and allows the aggregation of routers. Internet Protocol version 6 (IPv6) uses CIDR routing technology and CIDR notation in the same way as Internet Protocol version 4 (IPv4). IPv6 was designed for fully classless addressing. In CIDR, all Internet blocks can be of random size and classless addressing uses a variable number of bits for the network and host portions of the address. A view of the traffic collected using Wireshark is shown in Figure 4. It illustrates the protocol structure for a randomly selected BGP update message, which contains path attributes for the advertised Network Layer reach ability Information (NLRI). It opens and saves captured packet data, imports and exports packet data from and to other capture programs, filters and searches packets based on various criteria, colorizes packet display based on filters, and creates various statistics.As messages originate from multiple protocols, the frame shows Ethernet protocol source and destination address, the source and destination addresses of IP, source and destinations port numbers for TCP, and details of BGP. The update message has a marker of 16 bytes and length of 19 bytes. There are four types of message like type 1 indicates open message, Type 2 indicates that this message is an update message, type 3 indicates notification message, and type 4 indicates keepalive message. IGP is assigned to the origin attribute, AS path attribute has a length of 19 bytes, maximum hop limit 64, and payload length 39 bytes.

4.2.3 TCP Operations

The TCP operation is defined in RFC1323 are no operation (for padding), maximum window size (SYN), window scale (SYN), SACK permitted (SYN), SACK option (Acknowledges), time stamp (SYN & Acknowledges). The usage of the TCP SACK option is negotiated during the 3-Way hand shake. The Selective acknowledgement (SACK) option can be activated from one or both sides. Without SACK option, only the last received segment of a contiguous series can be acknowledged. The SACK Option allows acknowledging non-contiguous segments of a series and can request for specific segments. The SACK Option can improve the throughput of LFN’s significantly. Acknowledges (ACK) is used to point whether the acknowledgment field is valid. PSH is place when the sender requests the remote application to push this data to the remote application. RST is used to reset the connection. SYN (for synchronize) is used inside the connection start up phase, and FIN (for finish) is used to close the connection in an arranged mode. Information gathered during the handshake consists of the sender and receiver advertised window Sizes (rwnd), maximum segment size (MSS), whether a window scale option (WS) is being used, and if the sender and receiver support selective acknowledgement (SACK) options. The TCP checksum is applied to a synthesized header that contains the source and destination addresses from the external IP datagram. The first stage of a TCP session is establishment of the connection. This requires a three-way handshake, ensuring that both sides of the connection have an explicit understanding of the sequence number space of the remote side for this session.

The performance insinuation of this protocol exchange is that it takes one and a half round-trip times (RTTs) for the two schemes to synchronize status before any data can be sent. Once the connection is established, TCP starts slowly to determine the bandwidth of the connection and to avoid overflowing the receiving host and other devices or links in the path. After the connection has been established, the TCP protocol manages the consistent exchange of data between the two systems. The traffic service reply time is explicit as the time between a request and the corresponding response. A single packet of length 19 is sent with the PSH flag set. The PSH flag indicates to the receiver that the contents of the receive buffer should be immediately passed to the application layer. Another data packet of size 19 is sent. At this point there are 20 bytes of in flight or unacknowledged data on the wire.



Standard input - Graph Analysis

Time	172.16.15.1 172.16.15.2	2406:6400::1 2406:6400::2	Comment
16.146	(32839) PSH, ACK - Len: 19 → (179)		Seq = 1 Ack = 1
17.738	(32839) ← ACK (179)		Seq = 1 Ack = 20
54.039		(23157) PSH, ACK - Len: 19 → (179)	Seq = 1 Ack = 1
54.039	(32839) PSH, ACK - Len: 19 → (179)		Seq = 1 Ack = 20
55.474	(32839) ACK → (179)		Seq = 20 Ack = 20
55.521		(23157) ← ACK (179)	Seq = 1 Ack = 20
71.901		(23157) PSH, ACK - Len: 19 → (179)	Seq = 1 Ack = 20
74.054		(23157) ACK → (179)	Seq = 20 Ack = 20
86.971	(32839) PSH, ACK - Len: 19 → (179)		Seq = 20 Ack = 20
91.417	(32839) ← ACK (179)		Seq = 20 Ack = 39
113.303		(23157) PSH, ACK - Len: 19 → (179)	Seq = 20 Ack = 20
116.891		(23157) ← ACK (179)	Seq = 20 Ack = 39
147.249	(32839) PSH, ACK - Len: 19 → (179)		Seq = 39 Ack = 20
150.213	(32839) ← ACK (179)		Seq = 20 Ack = 58
164.971	(32839) PSH, ACK - Len: 19 → (179)		Seq = 20 Ack = 58
167.170	(32839) ACK → (179)		Seq = 58 Ack = 39
178.402		(23157) PSH, ACK - Len: 19 → (179)	Seq = 39 Ack = 20
180.649		(23157) ← ACK (179)	Seq = 20 Ack = 58
190.321		(23157) PSH, ACK - Len: 19 → (179)	Seq = 20 Ack = 58
191.475		(23157) ACK → (179)	Seq = 58 Ack = 39
218.635	(32839) PSH, ACK - Len: 19 → (179)		Seq = 58 Ack = 39
221.864	(32839) ← ACK (179)		Seq = 39 Ack = 77
244.624		(23157) PSH, ACK - Len: 19 → (179)	Seq = 58 Ack = 39
246.091		(23157) ← ACK (179)	Seq = 39 Ack = 77
275.263	(32839) PSH, ACK - Len: 19 → (179)		Seq = 39 Ack = 77
277.806	(32839) ACK → (179)		Seq = 77 Ack = 58
281.238	(32839) PSH, ACK - Len: 19 → (179)		Seq = 77 Ack = 58
282.657	(32839) ← ACK (179)		Seq = 58 Ack = 96

Figure 5. TCP connection handshake

The sequence number of the TCP SYN segment that is used to initiate the TCP connection between the router to router. The flow graph shows the BGP peers participating in the traffic exchange. At time 16.146 s, an update message is sent from source address 172.16.15.1 to the destination address 172.16.15.2, port 179 and the destination sends an acknowledgement back to the source implying that it is ready for traffic exchange. An exchange of data happens between only two BGP peers at a time.When the addresses 172.16.15.1 and 172.16.15.2 exchange data among themselves, there is no data exchanged between 2406:6400::1 and 2406:6400::2.

The receiver acknowledges (ACK) data sent by the sender. A window size (rwnd) of 16346 is also advertised by the receiver with this ACK. When the connection was established, a congestion window (cwnd) of size 1 MSS (size 1460 bytes) is initialized. Each time an ACK is received this congestion window is increased by 1 MSS. With ACK sent with, the congestion window size is increased by 1, like, cwnd = cwnd + 1 MSS, or 2 MSS = 2920 bytes. This pattern of sending data and receiving ACKs between the sender and receiver continues. When congestion is encountered, as indicated in the trace by a fast retransmission, a congestion avoidance algorithm is used to reduce the sender's window size, and to grow it back towards the receiver's advertised window size. Congestion avoidance requires that another variable be maintained called the slow start threshold or sshtresh. The flow graph of the captured IPv4 and IPv6 traffic is shown in Figure 5. It includes the source address, destination address, TCP port number, TCP message (ACK).The flow graph has shown the IPv4 and IPv6 BGP peers participating in the traffic exchange. At time 17.738 s, an ACK message is sent from source address 172.16.15.2 to the destination address 172.16.15.1 and the destination sends an acknowledgement back to the source implying that it is ready for traffic exchange. An exchange of data happens between only two BGP peers at a time. At time 55.521 s, an ACK message is sent from source address 2406.6400::2 to the destination address 2406.6400::1 and the destination sends an acknowledgement back to the source implying that it is ready for traffic exchange. An exchange of data happens between only two BGP peers at a time. When the destination address receives the acknowledgement, it knows that the link is active and it resumes the data exchange again. The Wireshark System recognizes many abnormalities or errors and creates a list sorted by severities likes: segment lost, duplicate ACK, retransmissions, fast Retransmissions, zero window and window full.

4.2.4 Time Sequence Graph

Time sequences show the general activity and events that occur during the lifetime connection. The X-axis represents time, and Y-axis represents sequence number space. TCP Stream Graph allows recognizing all the following abnormalities:

- *Lost Frames*
- Duplicate Frames
- Out of order Frames
- TCP Sequence number and Segment Sizes
- Acknowledges, Delayed Acknowledges
- Duplicate and Selective Acknowledges
- Retransmissions and Fast Retransmissions
- Windows Sizes, sliding Window, exceeded und frozen Windows Size
- Window Scaling, Zero Window and Window Full Situation
- Slow Start, full Flow rate and Flow throttling

A graph of TCP sequence numbers versus time is given in Figure 6. When an ACK is obtained, it contains the sequence number relating to the following byte to be received. By default, Wireshark convert all sequence numbers into relative numbers to facilitate comprehension and tracking of the loads engaged in a TCP session. This means that the sequence number corresponding to the first packet in a TCP connection always begins with 0 and not from a random value 0 - (2^32)-1) generated by the TCP/IP stack of the operating system. By under perfect situation, the representation of connection will show a line growing over time showing the effective performance of TCP connection. A time sequence graph is shown in Figure 6. Using the Time-Sequence-Graph (Stevens) of this trace, all sequence numbers from the source (2406:6400::5) to the destination (2406:6400::4) are increasing monotonically with respect to time. If there is a retransmitted segment, the sequence number of this retransmitted segment should be smaller than those of its neighboring segments. When ACKs are delayed, such as when an ACK is produced for every other packet, the growth is still approximately linear but somewhat slower. Due to occasion's gaps and jumps that interrupts the continuity of the line. This is normally due to a resend of data as a consequence of lost segments, ACK duplications, expired timeouts, etc.

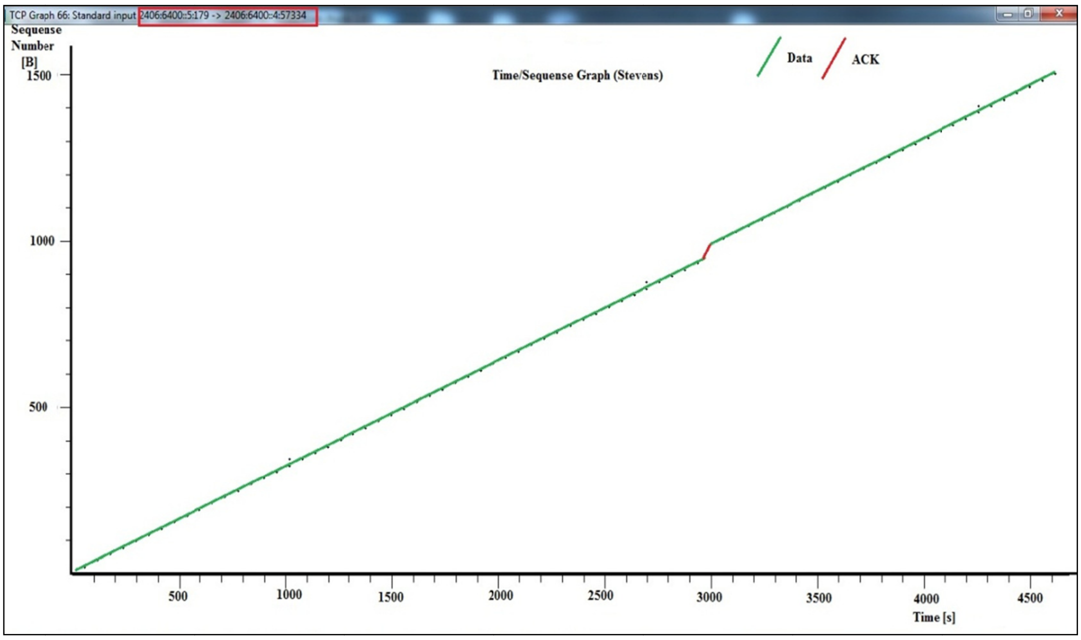


Figure 6. Time sequence graph

The time-sequence graph for the same period mentions there are gaps (showed in red) between sequence numbers, this indicating congestion in the network. These gaps also concur with the fast retransmission and retransmission events in the trace. The round trip time graph for ACK indicates some dots that are gather closer towards the x-axis indicating a consistent response time but there are a quit a few dots that are sharply climbing towards the top. The dots in red color are during the time the fast retransmits arise and shown the maximum length in RTT. This graph gives a very valuable source of information to notice anomalies in the behavior of certain connections. The given time-sequence graph for the connections shows a reasonable slope, equal to the maximum bandwidth from end-to-end.

In Figure 7 we have shown time sequence graph in TCP trace from 2406:6400::9:179 to 2406:6400::8:50475. The *y*-axis of Figure 7 represents the relative TCP sequence number. Each small tick mark represents 5,000 sequence numbers. The *x*-axis is time, in seconds. The solid line comprises many smaller *I*-shaped line segments, each of which represents the range of sequence numbers contained in a TCP segment. The height of the *I* indicates the user-data payload size, in bytes. The slope of the “line” formed by these I-shaped characters is the data rate achieved by the connection. Any movement to the lower right indicates a retransmission.

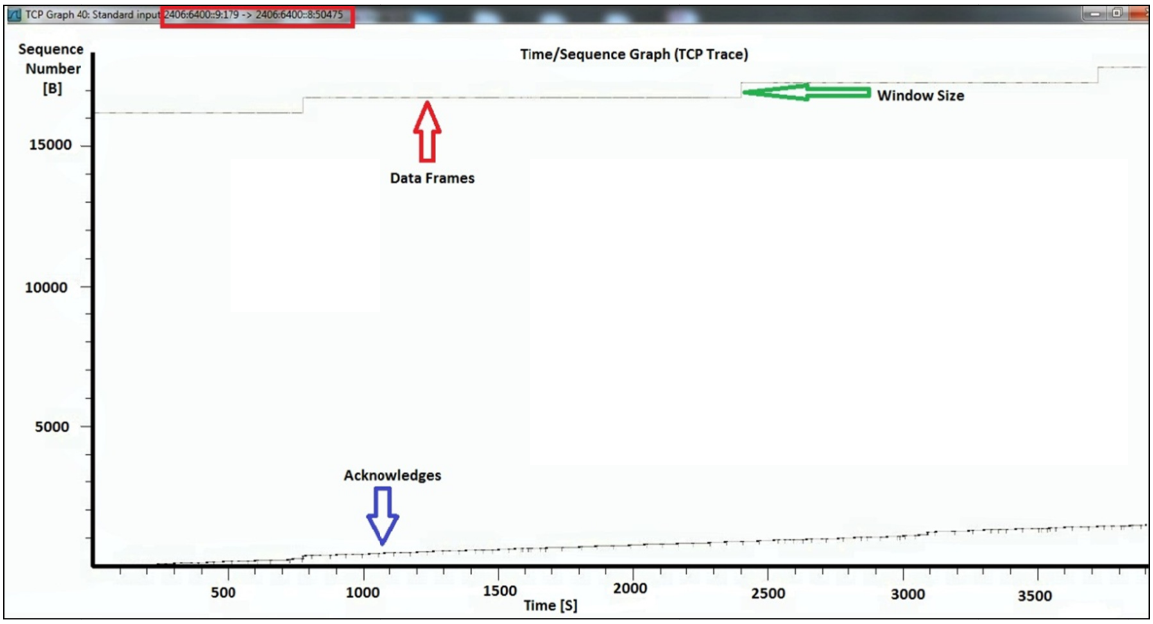


Figure 7. Time sequence graph (TCP trace)

The slope of the line for any given time range provides the average throughput over that time. As we can see, the highest sequence number sent was about 17000 at time 3500, which provides for a rough average good put rate of 4.85 bytes/s.

The bottom line represents the highest ACK number received at the sender so far. As discussed previously, TCP searches for additional bandwidth while it operates, by increasing its congestion window. It does not violate the receiver’s advertised window. We see this in operation in this graph as the upper line moves from the lower line toward the upper line over time. If the upper line is never reached, either the sender or the usable network capacity is the limiting factor for the throughput of the connection. If the upper line is always reached, the receiver’s window is the likely limiting factor.From the figure it shown increase of time sequence of Byte are increase accordingly. Here data frame are beginning from 16000 Bytes.

4.2.5 RTT Measurements

Round-trip time (RTT) is the duration of time it takes for a signal to be sent plus the length of durations it takes for an acknowledgment of that signal to be received. This time delay consequently comprisesof the transmission time spanbetween the two points of a signal. In the context of computer networks, the signal is generally a data packet, and the RTT is also known as the ping time. The RTT was originally estimated in TCP which can be found reference number: RTT = ($\alpha \cdot$ Old RTT) + ((1 − α) $\cdot$ New round trip sample), Where α is constant weighting factor ($0 \leq \alpha < 1$). Choosing a value α close to 1 makes the weighted average resistant to changes that last a short time. Choosing a value for α close to 0 makes the weighted average respond to changes in delay very quickly. In a network, particularly a wide-area network or the Internet, RTT is one of several factors affecting latency, which is the time between a request for data and the complete return or display of that data. RTT evaluation is one of the most important performance parameters in a TCP exchange, particularly in the case of a large file transfer. All TCP implementations eventually drop packets and retransmit them, no matter how good the quality of the link. If the RTT estimate is too low, packets are retransmitted unnecessarily; if it is too high, the connection may sit idle while the host waits for a timeout.

One simple way to find the RTT for such a flow is to find the time between the syn-ack and the data packet. The response flows take a little different information. When a host transmits a TCP packet to its peer, it ought to wait a certain time for an acknowledgment. If the reply does not reach within the expected period, the packet is implicit to have been lost and the data are retransmitted again. If the traffic flows over the wide-area Internet, a second or two seconds are reasonable during peak operation times. Network traffic for transmission control protocol RTT is shown in Figures (8, 9). If the RTT guess is too low, packets are retransmitted gratuitously; if it is too high, the connection may sit inactive while the host waits for a timeout.

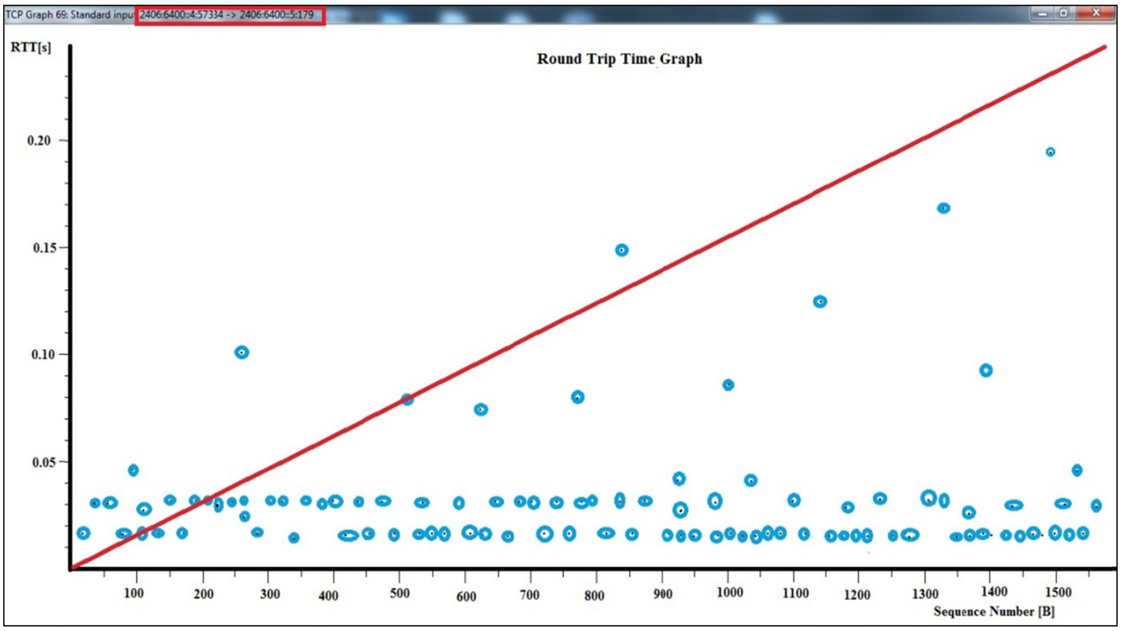


Figure 8. RTT graph for IPv6 connection

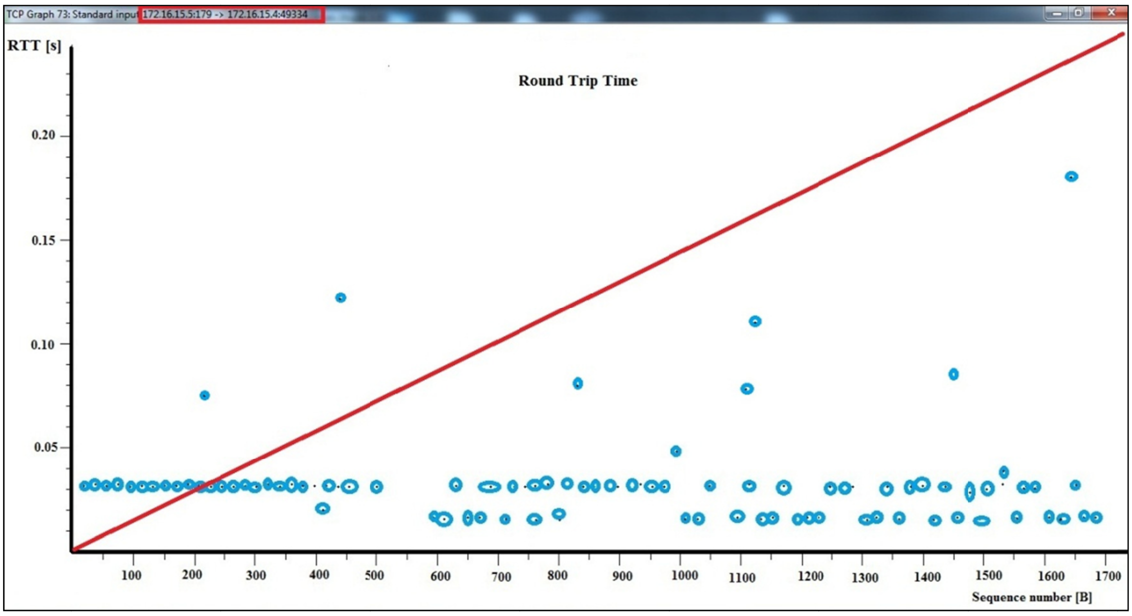


Figure 9. RTT graph for IPv4 connection

From the Figure 8 average RTT for IPv6 address 2406:6400::4 to 2406:6400::5 are below 0.05 s. Sometimes we got higher RTT when router working process becomes high. In the Figure 9 shows RTT for IPv4 address 172.16.15.5 to 172.16.15.4. In Figures 8 and 9 shown for IPv4 and IPv6 average RTT are below 0.05 s. The differences in round trip time on IPv4 and IPv6connection do not show significant difference.

4.2.6 TCP CUBIC

TCP CUBIC is an achievement of TCP that has an optimized congestion control algorithm for networks with large bandwidth delay product (BDP). The key aspect of CUBIC is that its window enlargement function is defined in real time so that its increase will be independent of RTT. Instead, window development depends only on the time between two successive congestion events. This property of CUBIC makes it friendly and fair to other flows in heterogeneous network. Congestion window of CUBIC is determined by the following function: $Wcubic = C(t\text{-}K)^3 + Wmax,$ where C is a scaling factor, t is the elapsed time from the lastwindow reduction,*Wmax*is the window dimensions just before the last window reduction and $K=\sqrt[3]{Wmax\beta/C}$, where*K* is theestimated time period that would take to reach *Wmax*. Disregarding further packet loss, K is computed as follows: *Wmin* is the reduced window size just after the last congestion event. Congestion window after congestion event is in steady state where it grows concavely up to *Wmax*, after which it enters probing state and grows convexly until next congestion event. ß is a constant multiplicative decrease factor applied to window reduction at the time of loss event. As per the Figure 8 exposed, the minimum RTT was around 0.03 sec and maximum RTT was around 0.18. The congestion C=0.5, t=0.18, K=3, β=0.8

$$K=\sqrt[3]{65535*0.8/0.5}= 47.1553$$

$Wcubic = 0.5(0.18\text{-}47.1553)^3 + 65535 = 13705.3004$ or 13705 approx. In this Graph we study that, CUBIC set up snooping for bandwidth in which the window grows step-by-step at the start, accelerating its development as it proceeds away from *Wmax*. This assessed development *Wmax* improves the constancy of the protocol and increase the consumption of the network while the fast growth left from *Wmax* ensures the scalability of the protocol.

4.2.7 Throughput

Throughput is imperative to understanding end-to-end performance. In Figure 10 shows the TCP throughput outcomes of a perfect form and a real IPv6 backbone for different packet sizes. From the TCP throughput outcomes we discerned the very close throughputs for both IPv4 and IPv6 networks in terms of small message sizes. From the TCP throughput results, we furthermore discoveredthat throughputs for both IPv4 and IPv6 networks in any message size are very similar. In a real large dual stack network, the throughputs of the IPv6 augment rapidly in small message sizes of 256 bytes. Then the throughput becomes level until the 768-byte message size range. Conversely, the throughputs decreased a little bit up to the 1408-byte message size range. In a real large-scale network, we attained a minor degradation for IPv6 compared to IPv4 networks because the overhead of the IPv6 address size is more significant. TCP reflect on two most important factors: TCP window size and the round trip latency to transfer data. If the TCP window size and the round trip latency are known, the maximum possible throughput of a data transfer between two hosts may be calculated in spite of of the bandwidth using the expression: TCP-Window-Size-in-bits / Latency-in-seconds = Bits-per-second throughput. Instantaneous throughput is the rate (bps) at which a host receives the packets.

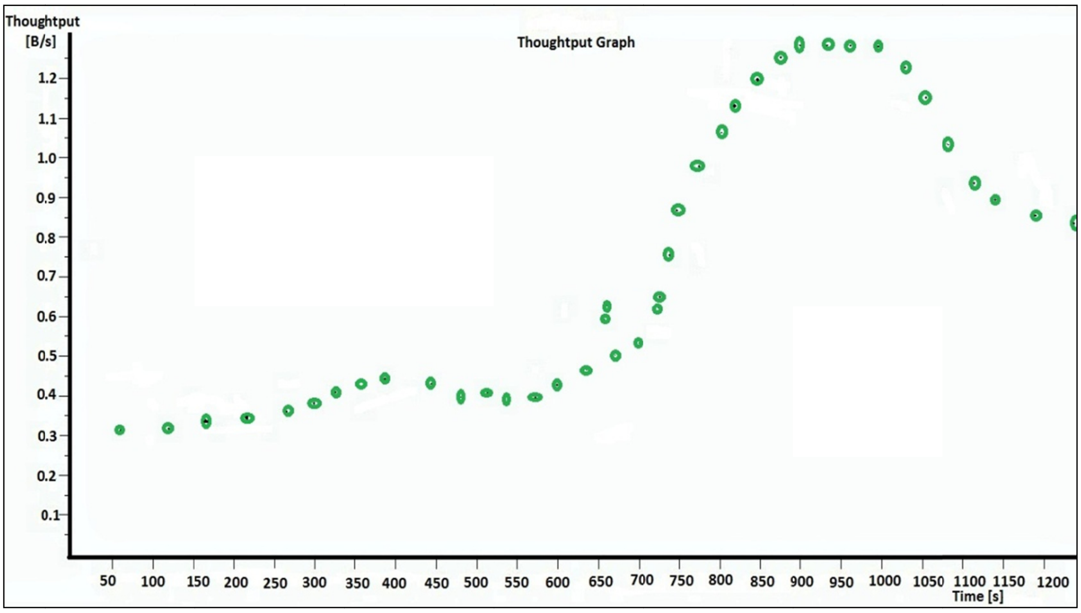


Figure 10. Throughput graph

If the packets consist of *F* bits and the transfer takes *T* seconds for the host to receive all F bits, then the average throughput of the packets transfer is *F/T* bps. In the figure we have shown throughput [Byte/second] increasing over increase of time. At 950 s throughput becomes maximum 1.2 B/s. After a certain times throughput becomes stable. We can calculate throughput two ways. Throughput without window scaling depends on TCP window size and round trip time.Maximum throughput = TCP Window Size / Round-trip time.Throughput with widow scaling depends on bandwidth, round trip time and TCP window size.Throughput with window scaling calculate by given formula Scaling factor calculation = Bandwidth x Round trip Time /TCP window Size. For 10Mbit/second scaling factor is = 10Mbit/s x 200ms/ 500000bits = Factor 4. We analysis the throughput, when error rate is 0, we reached maximum throughput. When router process increases rapidly congestion occurred due to RTT loss in router interface. We determined whether window is put reasonable is the most precise. If the window is put unreasonable, the throughout produced by congestion vary significantly. Suppose that each link is 10 Mbits / s, RTT is around 2-8 ms from the router interface, the length of each TCP data segment is 64 to 1000 bytes, the value of the receiving window of the receiver is rwnd = 32 to 64 bytes, window size is 16346, maximum hop limit 64 andthe simulation step size is 50 ms.

5. Conclusion

In this paper we have focused network migration from IPv4 to IPv6 in a large scale network. Dual stack network permits hosts to simultaneously reach IPv4 and IPv6 content making it a flexible coexistence approach.In the tunnelling concept we cannot move fully in IPv6. There are also other problems like TCP migration from IPv4 to IPv6 increase protocol overhead that increase latency and packet delay, each and every router need to configure tunnel for IPv6, which gives more overhead. It is also difficult to maintain the network as a Service Provider.In this paper, we conducted IPv6 address planning, an end-to-end performance assessment on a real large-scale network backbone. In order to build a dual stack network, it is necessary devices and network infrastructures that have supported IPv6 connections. The whole system of dual stack must be IPv6 compatible from source address to destination address. The following shows our investigative findings are explored on end-to-end user application performance using metrics such as: round trip time, time sequence, TCP throughputs, the IPv6 network does as well as the IPv4 network in terms of end-to-end performance. In dual stack network round trip time on IPv6 and IPv4 connection do not show significant difference. In a real large dual stack network situation, the throughput of the IPv6 expanded rapidly in small message sizes of 256 bytes, and, it leveled out before 768-byte message size range. In dual stack network we found same TCP throughput for IPv6 and IPv4 network. Our estimation outcomes show that the dual stack network is adept to provide stable network connectivity for IPv4 and IPv6 end-hosts.

Acknowledgements

The authors would like to thanks Department of Electrical and Electronic Engineering, School of Engineering and Computer Science, Independent University, Bangladesh.

References

Arafat, M. Y., Ahmed, F., & Sobhan, M. A. (2014). On the Migration of a Large Scale Network from IPv4 to IPv6 Environment. *International Journal of Computer Networks & Communications (IJCNC), 6*(2), 111-126. http://dx.doi.org/10.5121/ijcnc.2014.6210

Aziz, M. T., Islam, M. S., & Khan, M. N. I. (2012). Throughput Performance Evaluation of Video/Voice Traffic in IPv4/IPv6 Networks. *International Journal of Computer Applications, 35*(2), 6-12.

Bates, T., Chandra, R., Katz, D., & Rekhter, Y. (2007). Multiprotocol Extensions for BGP-4. *Internet Request for Comments, RFC 4760, Jan. 2007.*

Ciflikli, C., Gezer, A., & Ozsahin, A. T. (2012). Packet traffic features of IPv6 and IPv4 protocol traffic. *Turk J. Elec. Eng. & Comp. Science, 20*(5), 727-749.

Fall, K. R., & Stevens, W. R. (2012). TCP/IP Illustrated. *Addison-wisely professional computer series, Pearson Education,* 2012, vol. 1.

Fatah, F. N., Suhendra, A., Marwan, M. A., & Firdaus, H. (2013). Performance Measurements Analysis of Dual Stack IPv4-IPv6. *Second Intl. Conference on Advances in Information Technology — AIT, 2013.*

Hinds, A., Atojoko, A., & Zhu, S. Y. (2013). Evaluation of OSPF and EIGRP Routing Protocols for IPv6. *International Journal of Future Computer and Communication, 2*(4), 287-291. http://dx.doi.org/10.7763/IJFCC.2013.V2.169

Internet Engineering Task Force (IETF) RFC 6052, 3513. (n.d.). Retrieved from http://tools.ietf.org/html/rfc6052,3513

Internet Engineering Task Force (IETF) RFC. (n.d.). Retrieved from http://tools.ietf.org/html/rfc3849, 4291, 6104

Lefty, V. R., Lizzie, N. D., Cinhtia, G. S., & Victor, C. P. (2012). Design and Simulation of an IPv6 Network Using Two Transition Mechanisms. *IJCSI International Journal of Computer Science Issues, 9*(6), 60-65.

Marrone, I. L., Barbieri, L. A., & Robles, M. M. (2013). TCP Performance - CUBIC, Vegas & Reno. *Journal of Computer Science & Technology, 13*(1), 1-8.

Rekhter, Y., Li, T., & Hares, S. (2006). A Border Gateway Protocol 4 (BGP 4). *Internet Request for Comments, vol. RFC 4271*, Jan. 2006.

Rhee, I., & Xu, L. (2008). CUBIC: A New TCP-Friendly High-Speed TCP Variant (2008). *ACM SIGOPS Operating Systems Review, 42*(5), 64-74. http://dx.doi.org/10.1145/1400097.1400105

Tahir, A., Shahbaz, N., & Afzaal, H. (2013). Network Migration and Performance Analysis of IPv4 and IPv6. *European Scientific Journal, 8*(5), 72-84.

Wang, Y., Ye, S., & Li, X. (2005), Understanding Current IPv6 Performance: A Measurement Study. *10th IEEE Symposium on Computer Communications,* June 2005. http://dx.doi.org/10.1109/ISCC.2005.151

Copyrights