International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

QoS Provisioning Using Latency for IPV6

T. Vengatesh[1], Dr. S. Thabasu Kannan[2]

[1]Research Scholar, Research and Development Center, Bharathiar University, Coimbatore,
HOD, Dept. of .MCA, VPMM Arts & Science College for Women, Krishnankoil, Virudhunagar

[2]Principal, PCET, Sivagangai, India

Abstract: ***The main aim of the paper is to perform an unbiased empirical performance analysis between the two protocol stacks:IPv4 and IPv6, and how they are related to the performance on identical settings, and also focuses on QoS provisioning using latency. Here we investigate the latency while using TCP and UDP. Here two OSs (W2K and Linux Ubuntu) are configured with the two versions of IP and empirically evaluated for performance difference in terms of latency. This proposed QoS scheme is especially designed for transitional IP network containing both IPv4 and IPv6 network nodes, which is the reality in the process of internet transition from IPv4 to IPv6. Our simulation results show that the some application using IPV6 can be efficient manner. The findings reveal several significant factors which affect IPv6 implementation.***

Keywords: IPV4, IPV6, LATENCY

1. Why IPV6?

While increasing the pool of addresses is one of the most often-talked about benefit of IPv6, there are other important technological changes in IPv6 that will improve the IP protocol:

a) *No more NAT (Network Address Translation):*In IPv6 almost infinite number of addresses and the better renumbering makes NAT unnecessary so it will be possible to deploy new applications without tedious workarounds or random failures.

b) Auto-configuration:IPv6 supports the following types of auto-configuration:
Stateful auto-configuration requires a certain human intervention for the installation and administration of the nodes. It maintains state information so the server knows how long each address is in use, and when it might be available for reassignment. The server is listening on multicast addresses and memorizes client's state.
Stateless auto-configuration. Here each host determines its addresses from the contents of received router. When booting, the host asks for network parameters like IPv6 prefixes, default router addresses and hop limit. The routers and the servers have to be manually configured. Hosts can get automatically an IPv6 address, but it isn't automatically registered in the DNS.

c) No more private address collisions: The IPv6 header is completely re-designed. Required components are moved to the front of the header. Optional components are moved to an extension header; if there aren't any optional components, the extension headers are omitted and the packet size is reduced.

d) Better multicast routing: The effect of this is seen on all IPv6-enabled interfaces on the router which are then automatically enabled. When PIMv6 is enabled on an interface, the interface always operates in sparse mode. PIM-SM (Sparse-Mode) uses unicast routing to provide reverse-path information for multicast tree building, but it is not dependent on any particular unicast routing protocol.

e) Simpler header format:IPv6 headers have one Fixed Header and zero or more extension headers. All the necessary information that is essential for a router is kept in the Fixed Header. The extension header contains optional information that helps routers to understand how to handle a packet/flow.

f) Simplified and efficient routing: It provides forwarding capabilities between hosts that are located on separate segments. It performs sorting and delivery.IPv6 packets are exchanged and processed on each host by using IPv6 at the Internet layer. It creates the packets with source and destination address information. It passes packets down to the link layer. This process occurs in reverse order on the destination host. These are attached to two or more segments that are enabled to forward packets between them.

g) True quality of service: It specifies a guaranteed throughput level. One of the biggest advantages of ATM over competing technologies is that it supports QoS levels. It guarantees to their customers that end-to-end latency will not exceed a specified level. Flow label in IPv6 packet header provides an efficient way for packet marking, flow identification, and flow state lookup.

h) Built-in authentication and privacy support: In security systems, authentication is the process of giving individuals access to system objects based on their identity. Authentication merely ensures that the individual is who he or she claims to be, but says nothing about the access rights of the individual.

i) Easier administration:There are two separate address spaces for private addressing called *link-local* and *site-local*. A link-local address is like a single subnet and should not be routed. It is used for host auto configuration without DHCP, Neighbor discovery, Setting up ad-hoc LANs without a router. Site-local addresses are like a typical office containing several subnets. The subnet information is in the address so they can be routed within a site. They should not be forwarded outside the site.

2. Introduction

The IP is the principal communications protocol, comes under network-layer protocol, used for relaying datagram. IP

Licensed Under Creative Commons Attribution CC BY

is responsible for delivering datagram from the source host to the destination host on the basis of addresses. IP encapsulate the data to be delivered and addressing methods. IP was the connectionless datagram service. IP makes no guarantee that the packet will arrive without error. IP packet consists of a segment of data passed down from the transport or higher layer, plus a small *IP header* pretended to the data. *IP Address* is a unique identifier on a TCP/IP network to connect a private network to the Internet. IP address contains four segments of numbers (0 – 255) separated by periods.

Each node makes its forwarding decision based on the destination address within the IP packet header. The source address is examined when an error occurs. Routing decisions are based on the network-prefix of the IP destination address. The host portion of the IP address is used to differentiate individual hosts on the same link. The first major version of IP is IPv4, which is the dominant protocol of the internet. Its successor is IPV6. It contains two functions: *identifying hosts and providing a logical location service*.

Latency is the amount of time it takes one packet to travel from one host to another and back to the originating host. Latency is the delay from input to desired outcome. Latency greatly affects how usable and enjoyable electronic and mechanical devices as well as communications are. The latency is the wait time introduced by the signal travelling the geographical distance as well as over the various pieces of communications equipment. Having better latency could mean that the protocol would perform better for real time applications.

In communications, the lower limit of latency is determined by the medium being used for communications. In reliable two-way communication systems, latency limits the maximum rate that information can be transmitted.

Types of latency

a) Network latency is an expression of how much time it takes for a packet of data to get from one point to another. In some environments latency is measured by sending a packet that is returned to the sender; the round-trip time is considered the latency. Ideally latency is as close to zero as possible. It is measured either one-way (the time from the source sending a packet to the destination receiving it), or round-trip delay time (the one-way latency from source to destination plus the one-way latency from the destination back to the source). Round trip latency excludes the amount of time that a destination system spends processing the packet.
   Software used for latency arelft, paketto, hping, superping. NetPerf, IPerf. In a non-trivial network, the minimal latency is the sum of the minimum latency of each link, plus the transmission delay of each link except the final one, plus the forwarding latency of each gateway.
b) Audio latency is the delay between sound being created and heard. It is the cumulative delay from audio input to audio output. How long this delay is depends on the hardware and even software used .Potential contributors to latency in an audio system include A-D conversion, buffering, DSP, transmission time, D-A conversion and the speed of sound in air.
c) Operational latency: Any individual workflow within a system of workflows can be subject to some type of operational latency. It may even be the case that an individual system may have more than one type of latency, depending on the type of participant or goal-seeking behavior.
d) Mechanical latency:The behavior of disk drives provides an example of mechanical latency. It is the time needed for the data encoded on a platter to rotate from its current position to a position adjacent to the read-write head as well as the seek time required for the actuator arm for the read-write head to be positioned above the appropriate track.
e) Computer hardware and OS latencyis the delay between the process instruction commanding the transition and the hardware actually transitioning the voltage from high to low or low to high.

3. Experimental Setup

Two computers with similar hardware (CPU: Intel Pentium C2D, RAM 2GB, NIC PCI Intel Pro 100, HDD1TB) were connected using a cross-over cable and each of the OSs (W2K and Linux Ubuntu) to be tested were installed one at a time on P2P test-bed. IPv4 as the communication protocol was configured first and data was collected. Later this was replaced with IPv6 ensuring that all other parameters remained the same. D-ITG 2.6.1d was the primary tool used to evaluate performance of protocols on both the OSs. For accuracy all tests were executed 23 times, and to get the maximum latency for a given packet size, each run had duration of 30 seconds which netted about 50,000 packets to about 10,00,000 packets. The tests dealing with testing the latency of the TCP/UDP were limited to 1,472 byte datagrams because of a potential undocumented fragmentation bug in the IPv6 protocol stack.

4. Performance Evaluation

In this section, we present the results for IPv4 and IPv6 network protocols using both TCP and UDP transport protocols. Latency was empirically measured on P2P test-bed. The P2P Test-bed configuration had no routers between the end nodes. The PCs had a direct communication link via twisted pair Ethernet cable from one end to the other. For each experiment, we will be briefly reiterating the results depicted in the graph in case that it is not evident from the figures what the particular outcome may be.

As Fig1 and Fig2 indicate, both Windows and Linux Ubuntu offer comparable performance for the latency test, although Windows 2000 seems to perform slightly better than Linux Ubuntu in the larger packet sizes.

Licensed Under Creative Commons Attribution CC BY

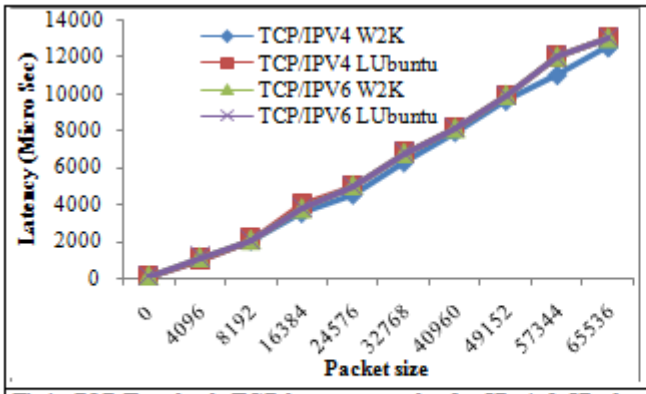

Fig1: P2P Test-bed: TCP latency results for IPv4 & IPv6 over Windows 2000 & Linux Ubuntu with packet size ranging from 64 bytes to 64 Kbytes

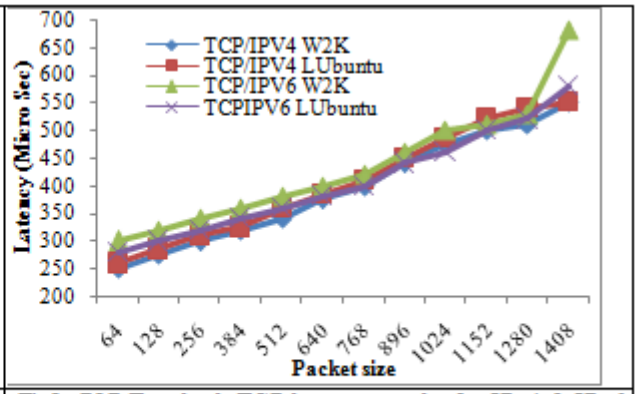

Fig2: P2P Test-bed: TCP latency results for IPv4 & IPv6 over Windows 2000 & Linux Ubuntu8.0 with packet size ranging from 64 bytes to 1408 bytes

- The notable difference between IPv4 and IPv6 under Windows was 15% higher latency for small packets and as little as 2% overhead for larger packets.
- Linux Ubuntu closed the gap to only 5% overhead for small packets while having as little as 1% overhead for larger packets.

In the Fig2, the odd spike in latency times for packet sizes of 1,344 and 1,408 byte packets in IPv6 under Windows 2000 is most likely due to a buffer allocation issue in which the contents of the packet plus the larger overhead of IPv6 cause the packet not to fit with the MTU of 1514 bytes. Therefore, the fragmentation mechanism probably caused the spike to occur. This kind of behavior is exactly the reason why we choose to display two different figures for each experiment; this way, we have enough detail at each respective level to see any odd behaviors.

For the UDP latency tests in Fig3 and Fig4,

- We have similar behavior as the TCP latency in terms of the IPv6 overhead.
- Specifically, we have a 7% to 2% overhead for Linux Ubuntu and an 18% to 4% overhead for Windows when comparing IPv6 with IPv4.
- However, unlike for the TCP latency tests, Linux Ubuntu marginally performs better in IPv4 than Windows.
- For IPv6, Linux Ubuntu performs quite better when compared to the Windows implementation of IPv6.

It is relatively important to notice the different scales for the latency used between Fig3 and Fig4; hence the same tests and data might look differently although they are only different because of the varying scales. This is of course valid latency the research work and therefore the scale should always be checked before conclusions are made between various figures. We tried to be as consistent as possible, but sometimes much detail would be left unnoticed if we were to keep scales identical throughout all the figures in entire research works

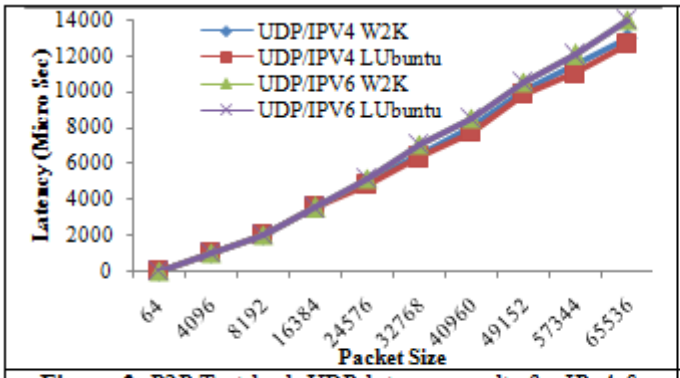

Figure 3: P2P Test-bed: UDP latency results for IPv4 & IPv6 over Windows 2000 & Linux Ubuntu with packet size ranging from 64 bytes to 64 Kbytes

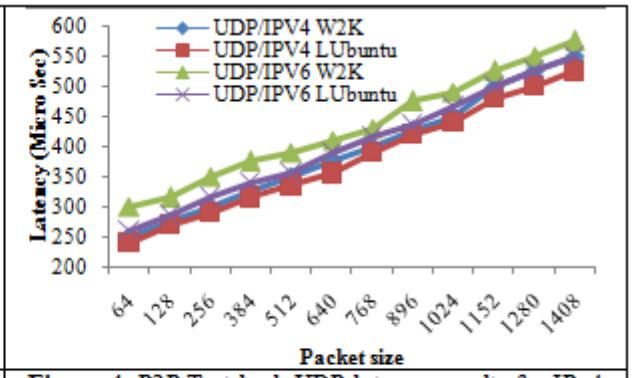

Figure 4: P2P Test-bed: UDP latency results for IPv4 & IPv6 over Windows2000 & Linux Ubuntu with packet size ranging from 64 bytes to 1408 bytes

For the CPU utilization for the latency tests in Fig5, it is obvious that TCP has a higher CPU utilization overhead over UDP, and IPv6 has a higher overhead above each IPv4 protocol.

Licensed Under Creative Commons Attribution CC BY

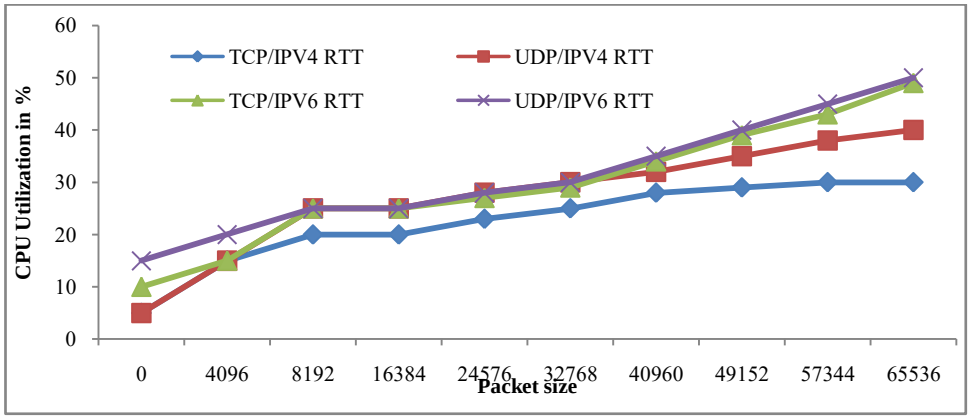

Figure 5: P2P Test-bed: CPU Utilization results for the latency experiments in IPv4 & IPv6 running TCP and UDP over Windows2000 with packet size ranging from 64 bytes to 64 Kbytes

5. Conclusion

The existing IPv6 has to improve its viability to reduce the overhead. IPv6 is still in the process of maturing it would be a matter of time to finally reflect its theoretical counterpart. Here the toughest part was in configuring the routers. It is very cumbersome and has many bugs with poor documentation and user feedback. In the next paper, we plan to use IBM and Ericson as test bed instead of P2P test bed. And also we have proposed to review on the above research work based on various transition mechanisms. IPv6 also supports prioritizing packets, which might be an easy way to offer a lighter version of QoS without specifying any requirements. According to our evaluation, IPv6 has a lack while using traditional data streams. In near future, this paper can be extended to incorporate more OSs including server environments. This paper proposes the end-to-end QoS provisioning by using latency. The results show the performance of the proposed end-to-end QoS provisioning by latency is maintained during network congestion.

References

[1] Wen-Shyang Hwang and Pei-Chen Tseng, AQoS-aware Residential Gateway with BandwidthManagement, IEEE Transactions on ConsumerElectronics, pp. 840-848, Aug. 2012.
[2] Shu-Fen Tseng, His-Chieh Lee, Te-Ching Kung,Shou-Lien Chou, and Jing-Yi Chen, TheDevelopment of Global IPv6 Products, Proc. ofthe 28th International Conference on AdvancedInformation Networking and Applications, pp.845-850, 2014.
[3] R. Banerjee, S. P. Malhotra, and M. Mahaveer,"A Modified Specification for use of the IPv6latency for providing an efficient Quality ofService using ahybrid approach", IETF IPv6Working Group Internet Draft, 2013.
[4] Guozhen Tan, Hengwei Yao, Yi Liu, andNingning Han, "QoS Provision for IPv6 TrafficUsing Dynamic Packet State", Proceeding of the jointinternational conference on Autonomic andAutonomous Systems and InternationalConference on Networking and Services, pp. 23-28, Oct. 2014.
[5] P. Srisuresh, M. Holdrege, IP Network Address Translator (NAT) Terminology and Considerations, RFC 2663, IETF, August 2013.
[6] Seaman, Mick and Klessig, Bob, 3 Com Corp. Going the Distance with QoS, Data Communication, February2013. Pages 120 to 129.
[7] Marcus A. Goncalves, Kitty Niles. IPv6 Networks, McGraw-Hill, 1998.
[8] S. Deering, R. Hinden, "IP Version 6 Addressing Architecture," Request for Comments 1884, Internet Engineering Task Force, December 2011.
[9] S. Thomson, T. Narten, "IPv6 Stateless Address Autoconfiguration," Request for Comments 1971, Internet Engineering Task Force, August 2012.
[10] T. Narten, E. Nordmark, W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)," Request for Comments 1970, Internet Engineering Task Force, August 2013.
[11] G. Tsirtsis, P. Srisuresh, "Network Address Translation – Protocol Translation (NAT-PT)," Request for Comments 2766, Internet Engineering Task Force, February 2013.
[12] B. Carpenter, C. Jung, Transmission of IPv6 over IPv4 Domains without Explicit Tunnels, Request for Comments 2529, Internet Engineering Task Force, March 2013.
[13] Fiuczynski, Marc E et. Al. "The Design and Implementation of an IPv6/IPv4 Network Adress and Protocol Translator". 2012.
[14] Draves, Richard P., et al. "Implementing IPv6 for Windows NT", Proceedings of the 2nd USENIX Windows NT Symposium, Seattle, WA, August 3-4, 2012.
[15] Seiji Ariga, Kengo Nagahashi, Asaki Minami, Hiroshi Esaki, Jun Murai. "Performance Evaluation of Data Transmission Using IPSec over IPv6 Networks", INET 2000 Proceedings, Japan, July 18th, 2012.
[16] Peter Ping Xie. "Network Protocol Performance Evaluation of IPv6 for Windows NT", Master Thesis, California Polytechnic State University, San Luis Obispo, June 2012.
[17] EttikanKandasamyKaruppiah. "IPv6 Dual Stack Transition Technique Performance Analysis: KAME on FreeBSD as the Case", Faculty of Information Technology, Multimedia University (MMU), Jalan Multimedia, October 2013.

Licensed Under Creative Commons Attribution CC BY