Optimizing Quality of Service Parameter of Multimedia Traffic on IPV6 Enabled Linux Kernel Based Scheduling Parameters and Algorithms for HPN

Hitesh Nimbark[1], Paresh Kotak[2], Shobhen Gohel[3] and Rajkamal[3]

[1] B H Gardi College of Engineering and Technology, Rajkot, Gujarat, India
hitesh.nimbark@gmail.com
[2] A V Parekh Technical Institute, Rajkot
kotakp2003@yahoo.com
[3] Govt. Poly Technique, Jamnagar
{shobhen.ce,rajkamal}@gmail.com

Abstract. Current era is the era of Multimedia traffics. This paper investigates into the scheduling aspects of real time multimedia streams with a real time test-bed environment. This will include the Operating System Scheduling Issues, as well as Routing at the Network Layer. An Ipv6 enabled distributed system is designed for priority based scheduling of Multimedia stream generated by VideoLAN Software. The Multimedia Applications must adhere to stringent real-time constraints and Quality-of-Service (QoS) requirements. A DiffServ based, QoS management approach is developed using a new developed kernel level module to parse the network packets and identify the DSCP value, which is marked during multimedia streaming at the source. This paper also highlights several other approaches and comparison that can be implemented for QoS based scheduling of multimedia traffic.

Keywords: Quality-of-Service, Traffic Control, Integrated services, Differentiated services, XORP, ZEBRA, Forwarding Engine Abstraction.

1 Introduction

Real Time Linux operating system scheduling demands the setup of a distributed infrastructure. The proposed architecture is inspired from a Video On Demand (VOD) Server based application. The system includes one Real Time Linux server where files are stored. The client requests for a particular multimedia file from a remote host. The file is streamed on the server and sent to the client side. The server and the client lie on different networks. Hence routers come into picture and the multimedia traffic needs to be routed, with Quality of Service parameters. The routers used in the setup is XORP [3,4,5], a software based router. There are at least four routers each configured for different network is used. This forms the middleware of the implementation. Most of the results derived are for unicast transmission of data. The protocol stack on which the architecture is implemented is IPv6 [1], since Ipv6 has

V.V. Das and N. Thankachan (Eds.): CIIT 2011, CCIS 250, pp. 710–715, 2011.

got inherent support for real time applications and multimedia communications. Here the client side could be a Windows based system also. But the source server and the software routers are all implemented on XORP router running over RTLinux3.1 kernel, further configured on Red Hat Linux or Fedora Core 6 nodes. The main objective of the paper is to study and configuration of software based routers (XORP, ZEBRA)[3] and implements various methods for QoS [6] based routing of Multimedia IP Traffic in IPV6 and implementation of a kernel level module to read/set the DSCP value in the Traffic Class of IPv6 header and TOS field of Ipv4 field. The UDP traffic that is generated by streaming a multimedia file is to be scheduled as per the DiffServ[2] architecture frame work. Finally, test the performance of the network in routing of multimedia traffic, with the developed module on the experimental test-bed. With a heavy traffic is generated artificially using traffic Generation Tool.

2 Project Architecture

As shown in above Figure 1, The experimental set up created for the implementation of priority based scheduling of multimedia traffic through a real time Linux operating system, and routing the stream through software based routers; required tremendous effort.

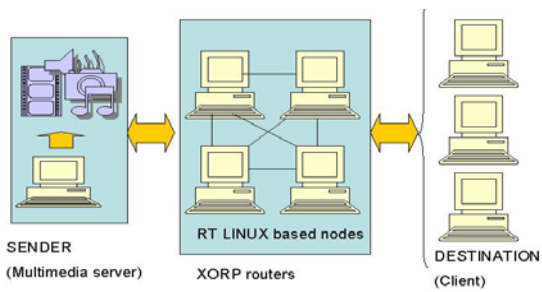

Fig. 1. Project Architecture (Higher Level)

Because of scarcity of space we are not including the following points of installation and configuration of this paper work eg. RTLinux 3.1 with Linux Kernel (2.6.26) [6] with IPv6 support, Ipv6 Layer Kernel Configuration as Router IPv6 To IPv4 Tunneling, kernel parameters Configuration at runtime, Streaming with Ipv6 support on RTLinux, XORP, ZEBRA configuration and IPV6 support, Forwarding Engine Abstraction of ZEBRA, Enable QoS support in Linux Kernel etc. The core part of the implementation was development of a module at the kernel level, which gets invoked, which is event driven and gets invoked when there is a packet at the network interface. The module parses the packets and reads the DSCP[22] value. A DSCP value of 16 is identified as a high priority multimedia traffic and if put in the High Priority queue. The other traffic, which is mostly TCP traffic is kept in the Low

Priority queue. The packets are released to the XORP router. On the routers, the DSCP value is checked propagated to the Routing Information Base (RIB)[3] at the Router. The RIB, in turn notifies this to the forwarding plane and release the traffic at a high priority.

A process on one end writes to a FIFO, which appears as a normal file, while another one reads from the other end. With RTLinux[6], the reader might be a real-time process, while the writer is a user-space program shuttling directives to the real-time code through the FIFO or vice versa. In either case, the FIFO devices are normal character devices (/dev/rtf*), and both ends can interact with the device through normal POSIX calls, such as open(), close(), read() and write(). In this implementation, the User Space code comprises of a program to read the configured network interface for outgoing packets. This is done using the pcap utility available in the pcap.h header file. The module is inserted at all the computers where XORP is running. The above implementation adds Priority based Scheduling or Class based Scheduling, which was otherwise lacking in XORP. The traffic is generated to create a real time scenario of loads of traffic in network, with different bandwidth and delay requirements; within a Computer Lab. The traffic was generated using the Multi-Generator (MGEN) tool. The Measurement tools have been chosen keeping in mind their open-source origins and flexibility of use. Ex OpenIMP, MGEN, Ethereal, Ping.

3 Test-Bed

This experiment was conducted using 2 streams, of which one was multimedia traffic while the other was normal periodic traffic. The bandwidth required is 120Mbps which is greater than 1OOMbps link. The multimedia traffic being Periodic with regular bursts, it was expected the delays would increase abruptly at burst instances when the traffic bandwidth required would be much greater than that available [14]. The queuing delays increase and average delay shoots up. Table 1 describes the traffic profile of the multimedia traffic's characteristics.

Table 1. Profile for Traffic Generation for both IPv6 and IPv4 network

Stream	Type	Rate (packet/Sec)	Size In bytes
Multimedia	Periodic Bust	4500	1024
UDP	Every 4.5 Sec	4500	1024
Normal TCP	Periodic	3000	1024

Table 2. Comparison of IPv4 and IPv6

	Packet Loss(%)	Average Jitter(msec)	Max jitter(msec)	Average receive time(msec)
IPV4	5	40	81	98
IPV6	1	21	65	20

The IPv6 network showed a better quality, as compared to the IPv4 network. Also the bandwidth consumption of the IPv6 network is much more stable as compared to the IPv4 network. This is shown in the Figure 2. Also, the UDP foreground traffic, Ex. multimedia traffic, suffers less packet loss with respect to the TCP background traffic (generated artificially through MGEN Traffic Generator Tool). The results derived after the experiment was used to display packet loss, delay and latency as shown in Figure 3.

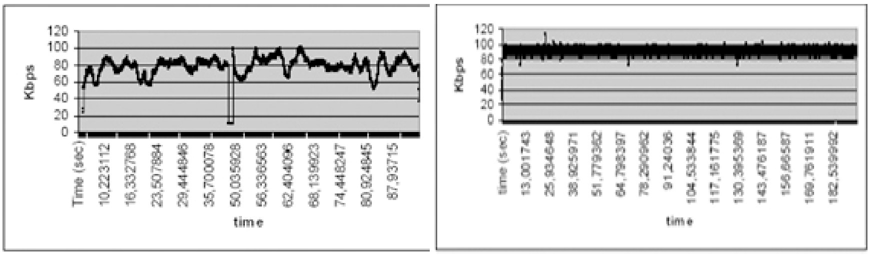

Fig. 2. (a) IPV4 (b) IPV6 bandwidth Consumption

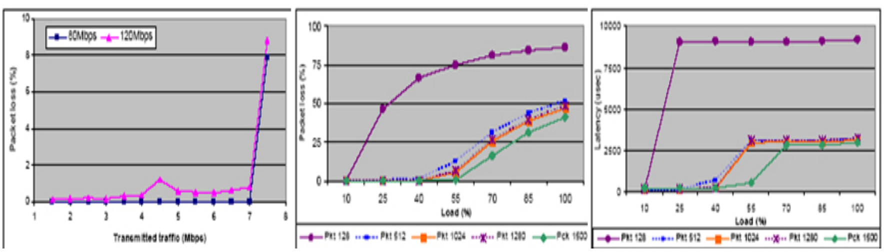

Fig. 3. Packet Loss for the (a) High Priority Multimedia Traffic (b)Different Packet Sizes (c) Latency for Different Packet Sizes

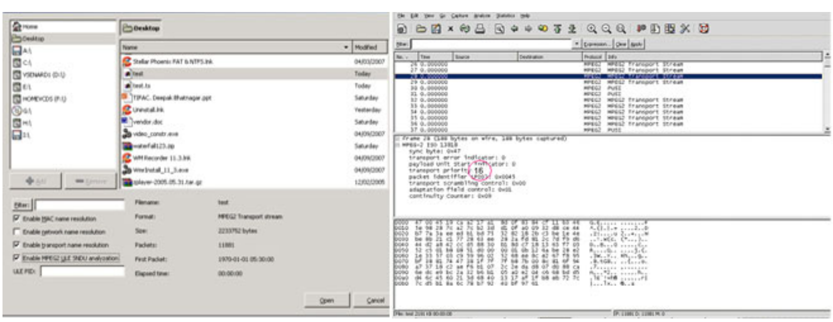

Fig. 4. (a) Ethereal's Open page for Transport stream file (b) showing a High Transport priority

The MPEG-2 TS (Transport Stream) dissector for Ethereal can display transport stream and section heading information, reassemble the sections and analyze the PSI (Program Specific Information) data. Open the .ts file (created through VideoLAN), from Ethereal (version 0.10.8). There are two possibilities to dissect MPEG 2 ULE (Unidirectional Lightweight Encapsulation), we will use etherial "Enable MPEG2 ULE SNDU analyzation" option as shown in Figure 4. If these settings are wrong the analysis will be incorrect. The Figure 4 shows the Transport priority marked with red circle. The experimental results indicated that priority based IPv6 traffic has less of packet loss, latency and jitter problems as compared to the normal IPV6/IPv4 network.

4 Conclusion

The complete architecture and configurations was considered more preferable due to its extensibility features, powerful API, event driven approach etc. The proposed module implemented to gives the priority to multimedia traffic gives significant improvement in multimedia streaming application. Table 2 conforms that if we can release router to priority of Multimedia Streaming data through kernel level IPV6 field modification then the user can get waste differentiated performance and QoS parameter optimization in the output. Also, the experimental results indicated that priority based IPv6 traffic has less of packet loss, latency and jitter problems as compared to the normal IPV6/IPv4 network.

References

1. Deering, S., Hinden, R.: RFC 2460 Internet Protocol, Version 6 (IPv6) Specification (December 1998)
2. Mahajan, M., Parashar, M.: Managing QoS for Multimedia Applications in the Differentiated Services Environment, The Applied Software Systems Laboratory (TASSL), 94 Brett Road, Piscataway, New Jersey (October 2004)
3. XORP, `http://www.xorp.org/`, `http://www.zebra.org`
4. Nimbark, H., et al.: Proxy Based TCP/IP Stack Modification to Support QoS. In: International Conference on Computer Engineering and Technology (2010), 10-981-4289-67-1(PBK)
5. Chimata, A.K.: A Route Control Platform using XORP Design Considerations (July 27, 2006)
6. Ferguson, P., Huston, G.: Quality of Service: Delivering QoS on the Internet and in Corporate Networks. Wiley Press, New York (1998); Almquist, P.: RFC1349: Type of Service in the Internet Protocol Suite (July 1998)
7. Nimbark, H., et al.: Sundry Path Routing Protocol for Multimedia Application in Wireless Networks (2011), ISBN Online: 2230-9039, Print: 2231-2153

8. Nimbark, H., et al.: Optimizing TCP Quality of Service Parameter Using Linux Kernel based Tuning Parameters for HPN. In: International Conference on Computer and Network Technology. World Scientific Press (2009) 13-978-981-4289-67-2(PBK)
9. Nimbark, H., Gohel, S., Doshi, N.: Width of a General Tree for Packet Routing. In: Das, V.V., Stephen, J., Chaba, Y. (eds.) CNC 2011. CCIS, vol. 142, pp. 322–324. Springer, Heidelberg (2011)
10. Nimbark, H., Gohel, S., Doshi, N.: A Novel Approach for Matrix Chain Multiplication Using Greedy Technique for Packet Processing. In: Das, V.V., Stephen, J., Chaba, Y. (eds.) CNC 2011. CCIS, vol. 142, pp. 318–321. Springer, Heidelberg (2011) ISSN: 978-3-642-19541-9