International Journal of Current Engineering and Technology
ISSN 2277 - 4106

Available at http://inpressco.com/category/ijcet

Research Article

Design and implementation of MPLS for Ethernet Packets on FPGA

Mirza Raheber Raza[a*], Praveen Kumar Y G[a], M. Z. Kurian[a], and K.V. Narayanswamy[b]

[a]Sri Siddhartha Institute of Technology, Tumkur.
[b]Dept of EEE, M.S.Ramaiah School of Advanced Studies, Bangalore.

Accepted 25 June 2013, Available online 01 August 2013, **Vol.3, No.3 (August 2013)**

Abstract

This paper presents hardware architecture of Multi-Protocol Label Switching (MPLS). MPLS is a protocol used primarily to prioritize internet traffic and improve bandwidth utilization. MPLS solutions are meant to be used with Layer 2 or Layer 3 protocols. Ethernet although initially conceived as a Local Area Network technology has been steadily making roads into access and core networks. This paper presents hardware architecture to implement MPLS for Ethernet on FPGA.

Keywords: *MPLS, Ethernet, FPGA.*

1. Introduction

Today's Internet applications like Voice over Internet Protocol (VoIP), real-time video streaming, video calling requires very high data-rates. If the resources are limited then the network gets congested and high speed applications perform poorly. Increasing bandwidth is one of the options for these networks. However increasing bandwidth alone is not sufficient and there is a need for efficient prioritization of traffic and network resources.

Traffic Engineering (TE) and Quality of Service (QoS) is used to prioritize internet traffic and use network resources efficiently to provide services for Internet applications. TE and QoS is used to create Virtual Private Networks (VPN) services which are dynamically defined, created and activated to satisfy QoS parameters.

MPLS is a protocol framework which is defined and created for efficient and faster packet processing. MPLS implementations exist today are mostly software based due to its complexity. But software implementations tend to reduce overall performance of the network since packet processing can take the order of milliseconds and that kind of delay can reduce the performance of the network and also cause loss of data in high speed networks. Hence hardware implementations have been researched.

This paper is organized into the following sections. The second section gives the brief description of related work done in this area. The third section provides an overview of MPLS and explains its basic characteristics including the techniques used and the protocols necessary. In fourth section hardware architecture is proposed and last section gives the conclusion and future works of this project.

2. Related Work

MPLS is comprised of different protocols, each performing a different task in the MPLS work flow. Each protocol has been researched differently with respect to hardware implementation. The work in (M. Abou-Gabal et al , 2004) describes hardware implementation of IS-IS protocol. In (M. Abou-Gabal et al , 2004) research on hardware implementation of OSPF protocol is discussed. Research in (H. Wang et al , 2004) and (Li, Z. Taoet al , 2005) discuss the hardware implementation of subset of RSVP-TE and CR-LDP protocols respectively. The work in (S. Li et al, 2003) describes the hardware implementation of reconfigurable MPLS router.

The work in (Raymond Peterkin et al , 2007) has introduced a hardware processor for the implementation of MPLS using RSVP-TE as its signalling protocol. In (Peterkin.R et al , 2005), an embedded architecture for the MPLS protocol was proposed. The design uses both hardware and software to implement different aspects of MPLS. The architecture proposed implementing routing functionality in software, label switching functionality in hardware.

3. MPLS Overview

In MPLS, data transmission is performed on two types of routers. Label Edge Router (LER) is a device that operates at the edge of the access network and MPLS network. An LER typically contain interfaces to dissimilar networks (such as ATM, Frame Relay or Ethernet) and perform the

*Corresponding author: **Mirza Raheber Raza** is a PG Student**; Praveen Kumar Y G** is working as Lecturer, **Dr. M. Z. Kurian** as HOD, and **Dr. K.V. Narayanswamy** as Prof.

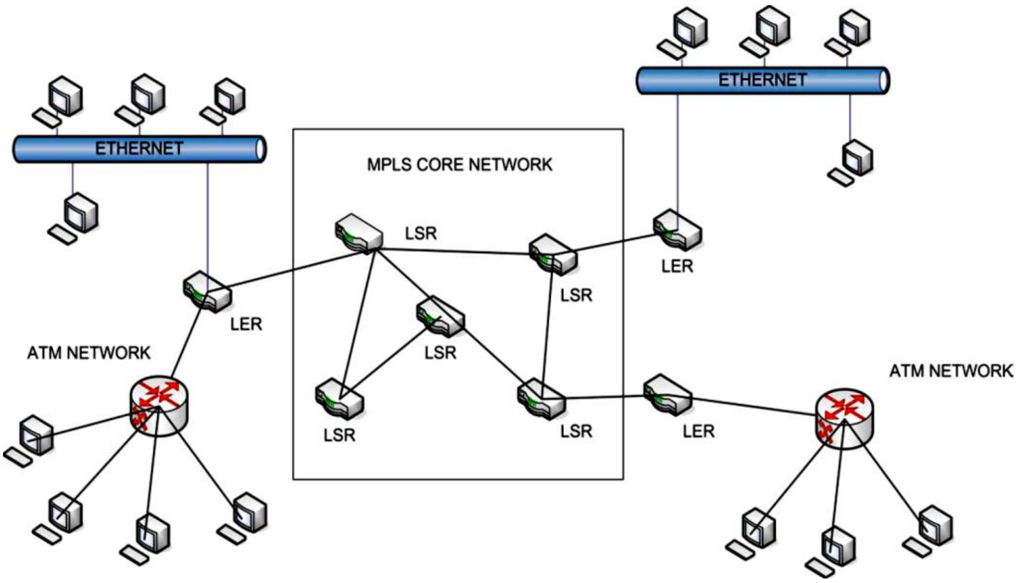


Fig 1: MPLS Network

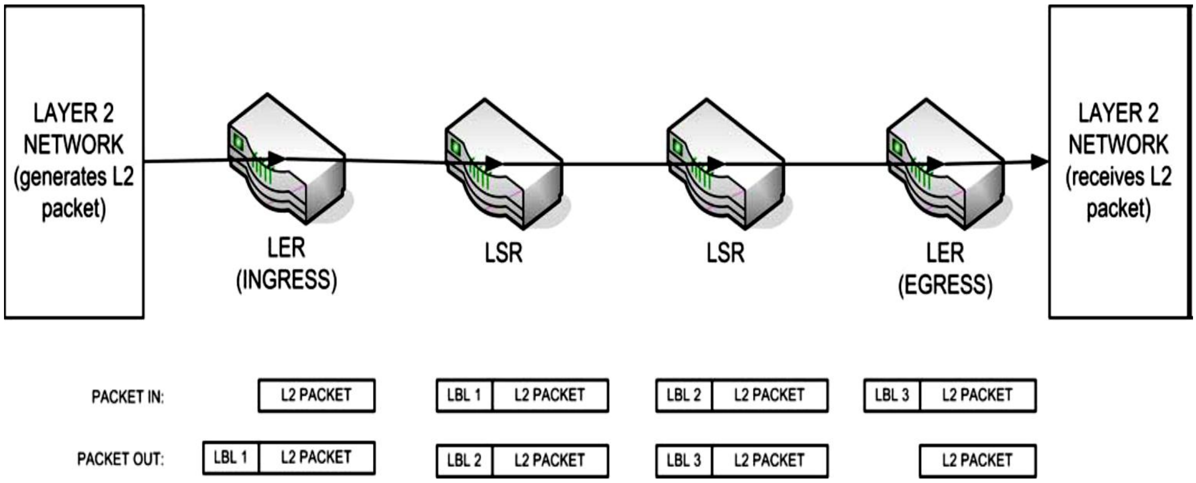


Fig 2 MPLS Packet Exchange

task of assigning or removing the label from the packet as the traffic enters or exits an MPLS network.

An LER that forwards traffic onto the MPLS network from the access network is called Ingress LER and an LER that distributes traffic back to access network from the MPLS network is called Egress LER. When Ingress LER receives a packet, a label is attached to that packet and forward onto MPLS network and when Egress LER receives packet, label is removed from the packet and traffic is distributed back to the access network.

A Label Switch Router (LSR) receives packet from the LSR or an LER. When LSR receives packets, it analyses the label and forward the packet to an LER or LSR depending on the label contents. LSRs form the core of the MPLS network.

A typical MPLS router is shown in fig 1(Peterkin.R et al , 2005). Each LER is connected to different access network and MPLS network. LSR is connected to another LSR or an LER. Packets are received through an access network and forwarded to and LER, this LER forward the packets to next LSR depending upon the Label Switched Patch (LSP) established. The packets travels through different LSRs before reaching the Egress LER which forward the packet back onto the access network.

Fig 2 shows a set of packet exchanges when a packet travels through the MPLS network (Peterkin.R et al , 2005). When the ingress LER receives packet from the access network, it is analyzed and a label is added to the packet. The new packet is then forwarded to the next LSR. Subsequent LSRs analyze the label, remove it and attach a new label so the next MPLS router can correctly interpret the label information. When the packet reaches the egress LER, the label is removed and the packet is forwarded to the appropriate access network.

Fig 3 illustrates the generic label format as described in (E. Rosen et al , 2001). Each MPLS label is 32 bits long with components for the label, Class of Service (denoted as ‘CoS’), a bit to denote the bottom of the stack (denoted as ‘S’) and a time to live (TTL) field. The label gives

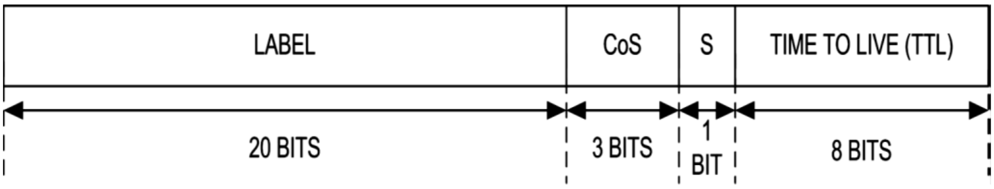


Fig 3 MPLS Label

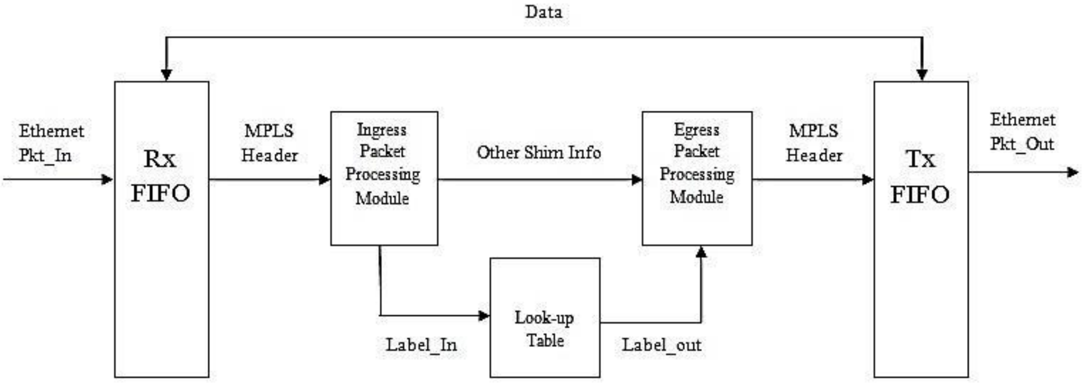
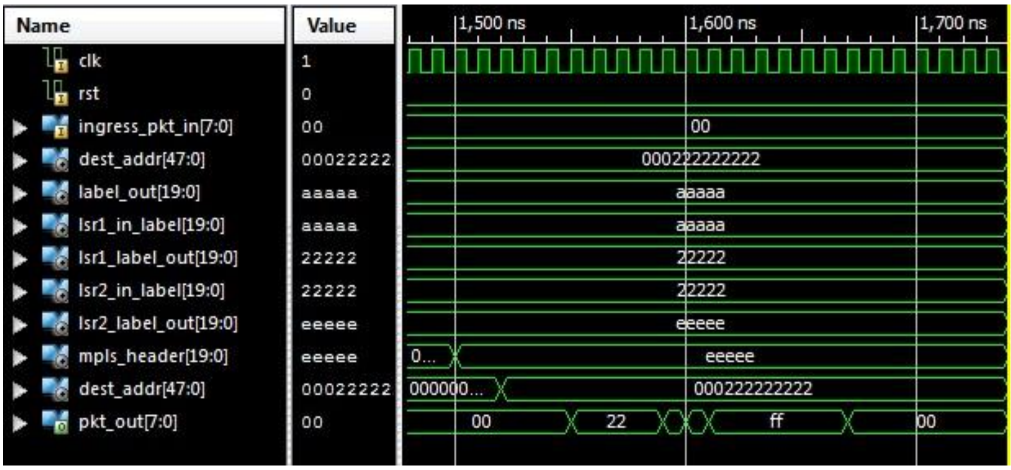


Fig 4 MPLS architecture



Name	Value
clk	1
rst	0
ingress_pkt_in[7:0]	00
dest_addr[47:0]	00022222
label_out[19:0]	aaaaa
lsr1_in_label[19:0]	aaaaa
lsr1_label_out[19:0]	22222
lsr2_in_label[19:0]	22222
lsr2_label_out[19:0]	eeeee
mpls_header[19:0]	eeeee
dest_addr[47:0]	00022222
pkt_out[7:0]	00

Fig 5.Simulation Result

information needed to forward the packet and is the basis upon which MPLS switching operations occur.

The CoS bits affect the scheduling and or discard algorithms applied to the packet as it is transmitted through the network. The S bit is set to one for the last entry in the label stack and zero for all other label stack entries. As described in (E. Rosen et al , 2001), the TTL is decremented by one each time the packet passes through a router. The packet is discarded when the TTL reaches zero.

4. Hardware Design

Fig 4 illustrated a high level description of the MPLS architecture. The architecture consists of Receive FIFO, Ingress Packet Processing Module, Look-up table Module, Egress Packet Processing Module and Transmit FIFO.

A. Ingress Packet Processing Module

The packet coming from input port is stored in Receive FIFO. The Ingress Packet Processing Module reads only the first 4 bytes which is the MPLS header attached to Ethernet packet. The Ingress Module separated the label and other shim info from the MPLS header. The Ingress Module checks the control information obtained from the MPLS header such as TTL and CoS, and identifies modification that has to be performed to the packet whether to forward, discard or modify the packet and sends the label to Look-up table Module where the new label is fetched.

B. Look-up Table Module

The Look-up Table stores and retrieves the label pairs. For each input label a corresponding output label is fetched from the look-up table. An Ingress LER does not have an incoming label and an Egress LER has not outgoing label. Those values are presumed to be zero.

C. Egress Packet Processing Module

The Egress Packet Processing Module gets the new label from the look-up table and other control information such as TTL, CoS from the Ingress module. The Egress Module constructs the 32-bit MPLS header and forwards it to Transmit FIFO where the packet data is stored. This complete packet which consists of the MPLS header and Ethernet packet data is sent through the output port.

5. Results

The MPLS architecture was implemented using Verilog hardware description language. These descriptions were then processed by standard Xilinx ISE 10.1 design tool suite, which performed synthesis, placement, routing, and bitstream (FPGA physical programming information) generation.

The bitstream generated was dumped onto XC2VP30 device of Xilinx Virtex 2 pro family. The number of slice flip flops used was 2396 and the number of 4 inputs LUT’s used was 965, representing 8% and 3% of the total resources available.

Fig 5 shows the simulation output of the implemented design. The figure shows the label that is attached to the packet when the packet is received at the Ingress LER. At LSRs the label switching is done where for every incoming label a new outgoing label is attached. At the Egress LER the label is removed from the packet and only the Ethernet packet is forwarded to the destination port.

Conclusion

Ethernet although initially conceived as a Local Area Network technology has been steadily making roads into access and core networks. This has led to a need for higher packet processing rate which are in gigabit range. In this paper, MPLS hardware architecture for Ethernet packets was proposed. This prototype implementation of MPLS on FPGA hardware has demonstrated the potential for 100x speedup in packet processing and forwarding when compared to software implementations.

References

E. Rosen, A. Viswanathan, R. Callon (Jan 2001) RFC 3031:Multiprotocol Label Switching Architecture

M. Abou-Gabal, R. Peterkin, D. Ionescu (July 2004)IS-IS protocol Hardware Architecture for VPN solutions, in *Proceedings of the 7th WSEAS International Conference on Communications*, Athens, Greece, July 12-15.

M. Abou-Gabal, R. Peterkin, D. Ionescu (2004) An Architecture for a Hardware Implementation of the OSPF Protocol, CAINE 2004 - *17th International Conference on Computer Applications in Industry and Engineering*, Orlando, Florida, USA, November 17-19

H. Wang, M. Veeraraghavan, R. Karri, T. Li (June 2004) A Hardware-Accelerated Implementation of the RSVP-TE Signaling Protocol, 2004 *IEEE International Conference on Communications*, Volume 3, Page(s):1609 – 1614.

T. Li, Z. Tao, H. Wang, M. Veeraraghavan (Jan 2005), Specification of a Subset of CR-LDP for Hardware Implementation.

Raymond Peterkin (2007), A Reconfigurable Hardware Architecture for VPN MPLS based Services *University of Saskatchewan Electrical Engineering* Master Thesis

S. Li (Aug 2003), System Architecture and Hardware Implementations for a Reconfigurable MPLS Router *University of Saskatchewan Electrical Engineering* Master Thesis

Peterkin.R, Ionescu.D (2005), Embedded MPLS Architecture Parallel and Distributed processing symposium, *IEEE*

Peterkin.R, ionescu.D (2005), A Hardware/Software Co-Design for RSVP-TE MPLS Parallel and Distributed processing symposium, *IEEE*