Implementation of Traffic Engineering in MPLS Networks Using Resource Reservation Protocol

SSivakumar*1, RSudha2 and Zumyla Shanaz3

1-3Electronics and Communication Engineering, Sri Shakti Institute of Engineering and Technology, L&T by-pass, Coimbatore.
Email: ssivakumar@siet.ac.in, R.Sudha@siet.ac.in, zumylashanaz@siet.ac.in

ABSTRACT
Traffic Engineering (TE) is the stage which deals with geometric design planning and traffic operation of networks, network devices and relationship of routers for the transportation of data. TE is that feature of network engineering which concentrate on problems of performance optimization of operational networks. It involves techniques and application of knowledge to gain performance objectives, which includes movement of data through network, reliability, planning of network capacity and efficient use of network resources. This thesis addresses the problems of traffic engineering and suggests a solution by using the concept of Multi-Protocol Label Switching (MPLS). Simulation has been done in GNS3 Software environment to compare the performance of MPLS against the IP network in a simulated environment. MPLS is a modern technique for forwarding network data. It broadens routing according to path controlling and packet forwarding. In this thesis MPLS is computed on the basis of its performance, efficiency for sending data from source to destination. GNS3 Software based simulation tool is developed to compare MPLS with IP network in a simulated environment. The results show the performance of MPLS network in comparison of IP network.


*Corresponding author
SSivakumar
Assistant Professor,
Electronics and Communication Engineering,
Sri Shakti Institute of Engineering and Technology,
L&T by-pass, Coimbatore.
Email: ssivakumar@siet.ac.in
INTRODUCTION

The term routing refers to tacking a packet from one device and sending it through the network to another device on a different network. Routers don’t really care about the host, they only care about the networks and the best path to each network. The logical network address of the destination host is used to get packets to a network through a routes network, and then the hardware address of the host is used to deliver the packet from a router to the correct destination host. A routing protocol is used by routers to dynamically find all the networks in the internetwork and to ensure that all routers have the same routing table. Basically, a routing protocol determines the part of a packet through an internetwork. Examples of routing protocols are RIP, OSPF, EIGRP and BGP. Once all the routers know about all networks, a routed protocol can be used to send user data in the form of packets through the established enterprise. Routed protocols are assigned to an interface and determine the method of packet delivery. Examples of routed protocols are IP and IPv6. Routing protocols are critical to a networks design. Dynamic routing protocols run only on routers that use them in order to discover networks and update their routing tables. Using dynamic routing is easier for the system administrator, than using labor intensive, manually achieved, static routing method but it will cost in terms of router CPU process and bandwidth in the network links. The source of the increased bandwidth usage and CPU cycles in the operation of the dynamic routing protocol itself. A router running a dynamic routing protocol shares routing information with its neighbouring routers, and it requires additional CPU cycles and additional bandwidth to accomplish that. The main objective of this project is to

- Study the routing protocols
- Simulate the routing protocols in a simulator
- Compare the performance of routing protocols

IP ROUTING

IP routing is the process of moving packets from one network to another network using routers. The IP routing process is important to understand because it pertains all routers and configurations that use IP. IP routing is used to forward packets from one node to other in the internetwork. IP routing is also used to determine the data has to follow to reach the destination node through the internetwork of nodes. The data is routed in the form of packets. The packets carries the date with information of source and the destination addresses. IP routing enables a router to build a forwarding table or also called as a routing table to determine the next hop that the data packets should be sent in order to reach the destination node. To be able to route packets from source to the destination, a router should contain the following information,
- Destination Address
- Neighbor routers from which it learns about all remote networks
- Possible routes to all remote networks
- The best route to each remote network
- How to maintain and verify routing information

**GNS3**

GNS3 is an open source software that helps simulate the complex networks as close as possible to the way real networks perform and such is achieved without having dedicated network hardware such as routers and switches. This software provides an intuitive graphical user interface to design and configure virtual networks, it runs on traditional PC hardware and may be used on multiple operating systems, including Windows, Linux, and MacOSX. In order to provide a complete and an accurate simulation, the GNS3 actually uses the following emulators to run the very same operating systems as in real networks:

- Dynamips, the well-known Cisco IOS emulator.
- Virtual Box, runs desktop and server operating systems as well as Juniper OS.
- Qemu, a generic open source machine emulator, it runs Cisco ASA, PIX and IPS.

**MULTI PROTOCOL LABEL SWITCHING**

To overcome most of the issues related to TE and IP Routing, MPLS is used. It is built by Internet Engineering Task force (IETF) to make the internet scalable, fast, carry heavy traffic, manageable and accept new routing architectures. MPLS uses TE which enables the network operators to reallocate packet flows to gain consistent distribution among different links. Making Network traffic to travel on specific directions permits to take the majority of the network capacity while making it easy to give uniform service levels to the users at the same time [12]. MPLS is a modern technique for forwarding network data. It broadens routing according to path controlling and packet forwarding. In MPLS packets are used for sending data. MPLS process is performed on two types of routers i.e. Label Edge Routers (LER) and Label Switch Routers (LSR). In figure 8 below, LER which is router 2 (Ingress router) works at the edge of the MPLS network and its interfaces are connected to the other networks. It routes traffic and works as an interface between the MPLS Network and the Layer 2 network. When router 2 receives a packet from the other layer 2 network, it attaches a label and sends the updated packet to the MPLS core network. The packet will then go through the path which is called Label Switched Path (LSP), going from one to another LER (egress) which is router 8. When the packet is received, the label is then removed from the packet and sent to the concern network. LER which sends the packet to the MPLS core network is called ingress while
LER which sends the packet to other dissimilar network is called egress. Both of these ingress and egress routers participate in the establishment of the LSPs before exchange of packets. The LSR (router 3, router 4, router 6 and router 7 in figure 8) comes in the core network of MPLS. They contribute in establishing the LSPs (links between two routers) and packet forwarding to the other MPLS routers. LSR receive packets from other connected LSR or LER, analyze its label and then forward the label according to the content of label.

MPLS consists of the following useful basic components.

- Path Management
- Traffic Assignment
- Network State Information Dissemination
- Network Management

**MPLS QUICK REROUTING**

There are two types of MPLS quick reroute methods. One to one backup method and another one is facility backup method. One to one backup method is used to create alternative route LSPs for every protected LSP on every point of failure, like node failure or link failure. The one to one backup failure is for every LSP which is individual. While facility backup method generates a bypass tunnel to guard a failure point. LSPs which are using the guarded facility are protected by the same bypass tunnel. That’s why facility back up method are pre-setup around guarded failure points in MPLS network and the bypass tunnels are predesigned in systems which are centralized to minimize the capacity of the network restoration. While one to one backup methods are established or deleted the backup LSPs on time when the protected LSPs come and enter in a distributed MPLS network.

**RESOURCE RESERVATION PROTOCOL**

Resource Reservation Protocol is used for soft state protocol which uses RESV and PATH commands to establish a LSP. In RSVP based on the protocol ID and destination IP address, packets are moved based on the raw IP datagram routing. Ingress LSR uses a PATH
message to notify each router along the selected LSP to acknowledge that this desired LSP is going to be established. After that, receiving LSR will use the message RESV with quality of service parameters and traffic traversing upstream to reserve the resources on every node along the desired LSP. Node along with the LSP will install the reservation for the concerned state by generating an entry on the label forwarding table. At each node along the path, RESV and path messages are used from time to time to refresh reservation states and the path [8]. RSVP-TE is developed and proposed to support Explicit Routed Label Switched Paths (ER-LSP) and to provide more features to RSVP. RSVP was developed to support MPLS LSP setups; some changes and extension are made to the original protocol to manage the TE requirements.

IMPLEMENTATION OF MPLS

GNS3 software is installed on a computer system. A simulation code is developed in GNS3 on the basis of which data will move from one router to another router. Routers take decision on the basis of routing table information. Figure 8 shows server “1”, router “1”, server “2” and router “6” belongs to IP networks which mean they are dissimilar with MPLS network. When the server wants to move data from server “1” to server “2” data will pass through MPLS network because server “1” is connected to server “2” through MPLS network. When the sender sends data from server “1” it will reach router “1”. Router “1” will look at its routing table and will send data to MPLS network where the packet will be received by the Ingress Label edge router. In Table 1, when packet arrives at router 3 which is the ingress label edge router. It will check the type of the data and will apply the Forward Equivalency Check (FEC). FEC refers to the collection of packets which are sent in a network on the same path, same manner and with the same forwarding treatment. Then Next Hop Label Forwarding Entry (NHLFE) will be applied to the IP packet which contains the following information.

- What will be the next router for the packet?
- Through which port the packet will move?
- Label number will be attached to the packet.

Label is a short fixed length important identifier which is used to identify FEC the label is binded with FEC by the LER. After that the packet is sent on a label switched path (LSP) to the Label Switched Router (Router 2). In router “2” the packet is received by Incoming Label Map (ILM). ILM is used to map incoming label to Next Hop Label Forwarding Entry (NHLFE). ILM will check the incoming label number and the port through which it is received and will map it with NHLFE. NHLFE will remove the incoming label number and will attach a new label number along with the port through which the packet will move to another router and the next
router address where the packet will be sent. Now router “2” will send the packet to the router “5” which is the Egress LER as well. Egress LER controls the traffic when it goes out of the MPLS network. So the router “5” will check the information in the packet along with the destination address when it checks that the destination address does not belong to the MPLS network so it will remove the labels of the MPLS network and will send the data to router 6 which will check the destination address and will send data to server “2”.

VIRTUAL PRIVATE NETWORKS (VPN)

Virtual Private Networks (VPNs) are a method of interconnecting multiple sites belonging to a customer using a Service Provider (SP) backbone network in place of dedicated leased lines. Each customer site is directly connected to the SP backbone. The SP can offer a VPN service more economically than if dedicated private WANs are built by each individual customer because the SP can share the same backbone network resources (bandwidth, redundant links) between many customers. The customer also gains by outsourcing the complex task of planning, provisioning and managing geographically distributed network to the SP. Unfortunately, existing VPN solutions are not all interoperable and may be tied to one equipment vendor and/or a single SP. This has created strong interest in IP-based VPNs running over the public Internet using standards-based interoperable implementations that work across multiple SPs. Many of these IP-based solutions require IP address-mapping or double encapsulation using two IP headers. This can require complex configuration management and requires additional processing at the entry to and exit from the SP’s networks. The new Internet technology, Multi-Protocol Label Switching (MPLS) forwards data using labels that are attached to each data packet. Intermediate MPLS nodes do not need to look at the content of the data in each packet. In particular the destination IP addresses in the packets are not examined, which enables MPLS to offer an efficient encapsulation mechanism for private data traffic traversing the SP backbone. MPLS can therefore, provide an excellent base technology for standards-based VPNs.

VPN TRAFFIC ENGINEERING

An LSP tunnel forms an excellent encapsulation scheme for VPN data flowing between two LSRs. But how do LSRs determine which LSPs to set up to provide connectivity for VPNs? In effect, how do LSRs decide which other LSRs provide access to the VPNs which they themselves serve? Even once this has been done, how should the different VPNs be mapped into LSP tunnels – a separate tunnel for each VPN, or a single tunnel for all VPNs? These are complex questions that do not have a single “right” answer. There is a number of factors that determine what VPN Traffic
Engineering (TE) scheme best suits the performance and scalability requirements of a particular customer and their SP.

1. Identifying VPN peers This is the first problem facing an LSR that has been configured to support a VPN. The simplest scheme is to use explicit manual configuration of the VPN peers. This is the traditional solution providing obvious and deterministic control of resources and security, but it does not scale well as the size and complexity of the VPN increases. Alternative schemes automate the process of discovering VPN peers using a directory or by overlaying VPN membership information on one or more routing protocols used on the SP network. This greatly simplifies the configuration task for a VPN since it means that each SP edge router need only be configured with information about the VPNs serviced by each of its customer interfaces. There is clearly a potential security trade-off here as rogue routers can pretend to give access to a VPN. In comparison, an IPSEC-based solution requires that each SP Edge router also be configured with security attributes for each peer in the VPN, which greatly increases the configuration complexity.

2. Multiplexing VPNs on an LSP Although LSRs in the core of the SP network do not have to examine the data flowing on VPN LSP tunnels, they are still aware of the existence of these tunnels. This can represent a scalability problem if a separate mesh of LSP tunnels is used for each VPN, because the core LSRs must at least maintain a forwarding table entry and associated resource reservation for each tunnel. If the SP supports thousands of VPN customers, the core LSRs could be required to maintain millions of LSPs. This is the same problem faced by VPN solutions based on ATM or Frame relay technology. Depending on the network topology, this large number of labels may also be beyond the capacity of the LSR switching hardware.

An alternative approach is to multiplex the traffic from multiple VPNs that share the same ingress and egress SP edge routers within a single LSP tunnel between those LSRs. This is achieved using label stacks, with a single outer tunnel set up across the core and an inner LSP that identifies the VPN for which the data is bound. The lower label in the stack is known only to the ingress and egress LSRs. This use of label stacks reduces the number of LSP tunnels exposed to the network core, but it ties VPNs together. The multiplexed VPNs cannot be routed separately or given different prioritization or drop priority by the core LSRs. The VPNs must also share a single network resource reservation within the network core, which may make it harder for the SP to guarantee the SLA for each individual customer.
3. Separating QoS classes: Multiplexing VPNs within a single tunnel helps to reduce the signaling load and forwarding table size in the core LSRs as the number and size of the VPNs increase. However, once the data for multiple streams has been clustered together in a single LSP, it is hard to provide distinct management of the different flows. The encoding of an MPLS label allows three bits to encode the Differentiated Services Control Point (DSCP). Thus, a total of eight Classes of service (CoS) can be set for packets within any one LSP. These bits can define queuing rules and drop priorities for packets carried on the LSP. In the case of an ATM-based network, there is just one bit available to encode the DSCP and this is usually used simply to indicate the drop preference. If a customer or SP needs to be able to differentiate more than eight DSCPs across the core, multiple outer LSP tunnels must be set up. Each outer tunnel carries a different CoS range and can be routed separately across the core. The interaction between setting up multiple outer tunnels across the core to carry more CoSs, and the need to minimize the number of such tunnels using VPN multiplexing on a single tunnel. The IETF draft ietf-mpls-diff-ext defines methods of signaling LSPs for CoS usage and ways of determining the interpretation of the DSCP bits. TE across the backbone MPLS TE can be used to distribute the load within a network and to guarantee bandwidth and QoS by controlling the routing of the outer VPN LSP tunnels across the SP backbone network.

**PROPOSED METHOD**
THE FISH PROBLEM

Let's make things more concrete by looking at a classic example of traffic engineering.

In this figure, there are two paths to get from R2 to R6:

- R2 → R5 → R6
- R2 → R3 → R4 → R6

Because all the links have the same cost (15), with normal destination-based forwarding, all packets coming from R1 or R7 that are destined for R6 are forwarded out the same interface by R2—toward R5, because the cost of the top path is lower than that of the bottom. This can lead to problems, however. Assume that all links in this picture are OC-3—roughly 150 Mbps of bandwidth, after accounting for SONET overhead. And further assume that you know ahead of time that R1 sends, on average, 90 Mbps to R6 and that R7 sends 100 Mbps to R6. So what happens here? R2 tries to put 190 Mbps through a 150 Mbps pipe. This means that R2 ends up dropping 40 Mbps because it can't fit in the pipe. On average, this amounts to 21 Mbps from R7 and 19 Mbps from R1 (because R7 is sending more traffic than R1). So how do you fix this? With destination-based forwarding, it's difficult. If you make the longer path (R2 → R3 → R4 → R6) cost less than the shorter path, all traffic goes down the shorter path. You haven't fixed the problem at all; you just moved it. Sure, in this figure, you could change link costs so that the short path and the long path both have the same cost, which would alleviate the problem. But this solution works only for small networks, such as the one in the figure. What if, instead of three edge routers (R1, R6, R7), you had 500? Imagine trying to set your link costs so that all paths were used! If it's not impossible, it is at least extremely difficult. So you end up with wasted bandwidth; in figure 3, the longer path never gets used at all.

What about with ATM? If R3, R4, and R5 were ATM switches, the network would look like figure 4.

![Figure 4: The Fish Problem in ATM Networks](image-url)
With an ATM network, the problem is trivial to solve. Just build two PVCs from R2 to R6, and set their costs to be the same. This fixes the problem because R2 now has two paths to R6 and is likely to use both paths when carrying a reasonably varied amount of data. The exact load-sharing mechanism can vary, but in general, CEF's per-source-destination load balancing uses both paths in a roughly equal manner. Building two equal-cost paths across the network is a more flexible solution than changing the link costs in the ATM network, because no other devices connected to the network are affected by any metric change. This is the essence of what makes ATM's traffic engineering capabilities more powerful than IP's. The problem with ATM TE for an IP network has already been mentioned—O(N^2) flooding when a link goes down and O(N^3) flooding when a router goes down.

**TRAFFIC ENGINEERING WITH MPLS**

MPLS TE combines ATM's traffic engineering capabilities with IP's flexibility and class-of-service differentiation. MPLS TE allows you to build Label-Switched Paths (LSPs) across your network that you then forward traffic down.

Like ATM VCs, MPLS TE LSPs (also called TE tunnels) let the headend of a TE tunnel control the path its traffic takes to a particular destination. This method is more flexible than forwarding traffic based on destination address only.

Unlike ATM VCs, the nature of MPLS TE avoids the O(N^2) and O(N^3) flooding problems that ATM and other overlay models present. Rather than form adjacencies over the TE LSPs themselves, MPLS TE uses a mechanism called auto route (not to be confused with the WAN switching circuit-routing protocol of the same name) to build a routing table using MPLS TE LSPs without forming a full mesh of routing neighbours.

Like ATM, MPLS TE reserves bandwidth on the network when it builds LSPs. Reserving bandwidth for an LSP introduces the concept of a **consumable resource** into your network. If you build TE-LSPs that reserve bandwidth, as LSPs are added to the network, they can find paths across the network that have bandwidth available to be reserved.

Unlike ATM, there is no forwarding-plane enforcement of a reservation. A reservation is made in the control plane only, which means that if a Label Switch Router (LSR) makes a reservation for 10 Mb and sends 100 Mb down that LSP, the network attempts to deliver that 100 Mb unless you attempt to police the traffic at the source using QoS techniques.

**SOLVING THE FISH PROBLEM WITH MPLS TE**

Figure 5 revisits the fish problem presented in Figure 3.
Like ATM PVCs, MPLS TE LSPs can be placed along an arbitrary path on the network. In figure 5, the devices in the fish are now LSRs.

The three major differences between ATM and MPLS TE are

1. MPLS TE forwards packets; ATM uses cells. It is possible to combine both MPLS TE
2. MPLS/ATM integration, but currently, this is not implemented and therefore is not covered here.
3. ATM requires a full mesh of routing adjacencies; MPLS TE does not.
4. In ATM, the core network topology is not visible to the routers on the edge of the network; in MPLS, IP routing protocols advertise the topology over which MPLS TE is based.

**USING MPLS IN REAL LIFE**

Three basic real-life applications for MPLS TE are

- Optimizing your network utilization
- Handling unexpected congestion
- Handling link and node failures

Optimizing your network utilization is sometimes called the *strategic* method of deploying MPLS TE. It's sometimes also called the full-mesh approach. The idea here is that you build a full mesh of MPLS TE-LSPs between a given set of routers, size those LSPs according to how much bandwidth is going between a pair of routers, and let the LSPs find the best path in your network that meets their bandwidth demands. Building this full mesh of TE-LSPs in your network allows you to avoid congestion as much as possible by spreading LSPs across your network along bandwidth-aware paths. Although a full mesh of TE-LSPs is no substitute for proper network planning, it allows you to get as much as you can out of the infrastructure you already have, which might let you delay upgrading a circuit for a period of time (weeks or months). This translates directly into money saved by not having to buy bandwidth.
Another valid way to deploy MPLS TE is to handle unexpected congestion. This is known as the \textit{tactical} approach, or \textit{as needed}. Rather than building a full mesh of TE-LSPs between a set of routers ahead of time, the tactical approach involves letting the IGP forward traffic as it will, and building TE-LSPs only after congestion is discovered. This allows you to keep most of your network on IGP routing only. This might be simpler than a full mesh of TE-LSPs, but it also lets you work around network congestion as it happens. If you have a major network event (a large outage, an unexpectedly popular new web site or service, or some other event that dramatically changes your traffic pattern) that congests some network links while leaving others empty, you can deploy MPLS TE tunnels as you see fit, to remove some of the traffic from the congested links and put it on uncongested paths that the IGP wouldn't have chosen.

A third major use of MPLS TE is for quick recovery from link and node failures. MPLS TE has a component called Fast Reroute (FRR) that allows you to drastically minimize packet loss when a link or node (router) fails on your network. You can deploy MPLS TE to do just FRR, and to not use MPLS TE to steer traffic along paths other than the ones your IGP would have chosen.

**CONCLUSION**

This thesis presents implementation and comparison of MPLS with IP network. To answer our research questions we have performed qualitative as well as quantitative study. Our thesis starts with literature review through which we have gained knowledge about our domain TE, IP and MPLS network. Literature review helped us in answering the first three questions of our thesis work. From our thesis we have concluded that MPLS is a better technique for traffic engineering as compared to IP network because of following reasons;

1. MPLS takes less time to send data to the destination
2. MPLS is efficient than IP networks
3. MPLS will be efficient if applied in the current internet architecture.

There are some concerns in our results as we have conducted a simulation for the validation of our results. This problem can be solved by two ways. One is performing this kind of experiment in reality and another is to perform simulation in an environment which is closer to the reality. We have selected the later approach but still real situation may produce different outcomes than the presented outcomes. The results presented in this thesis are validated for the text data. Therefore results obtained from this thesis can be different if different type of data is used in simulation. We try to overcome this problem by performing simulation in two steps. In order to find the differences in the results simulation was carried out in two steps. Our simulation shows consistency in the results obtained from two steps of simulation.
REFERENCES