APPLICATIONS OF TRAFFIC ENGINEERING
TECHNIQUES IN NEW GENERATION NETWORKS

A report submitted in partial fulfillment of the requirement of
the degree of B.Sc.

In
Electrical and Electronic Engineering
Under the Supervision of
Dr. Mohammed Ali Hamad

By
Mohamed Mamdouh Mohamed Saad El-Dien

To
The Department of Electrical and Electronic Engineering
University of Khartoum

July 2005
To my Aunt ...

Sister ...

Brothers ...

All people in my heart ...

I dedicate this work
ACKNOWLEDGMENT

I am greatly indebted to my supervisor Dr. Mohammed Ali Hamad for his guidance, unlimited help, and encouragement during this study research.

Thanks also extended to all my teachers in the Electrical and Electronic Engineering Department.


My huge thanks should be extended to my friend and my brother Eng. AbuBakr Hassan for his unlimited help.

Last and not at least I would like to express my sincere thanks and gratitude to my friends Mihuar Abd alla and Tamir Abd algader for their dispensable help and support through out my study.
# Contents

Abstract .......................................................................................................................... vi
الخلاصة ......................................................................................................................... vii
List of abbreviations ........................................................................................................ viii
Chapter One .....................................................................................................................
Introduction .................................................................................................................... 1
  1.1 Problems with existed networks: ........................................................................... 1
  1.2 Solution: .................................................................................................................. 1
  1.3 Report layout .......................................................................................................... 2
Chapter Two ....................................................................................................................
Section One: MPLS concepts and components ............................................................ 3
  2.1.1 Introduction ......................................................................................................... 3
  2.1.2 MPLS and IP ..................................................................................................... 3
  2.1.3 MPLS Components: .......................................................................................... 4
      2.1.3.1 LSRs and LERs ....................................................................................... 4
      2.1.3.2 FEC ......................................................................................................... 5
      2.1.3.3 Label Information Base ......................................................................... 5
      2.1.3.4 Labels and Label Bindings ................................................................... 5
      2.1.3.5 Label-Switched Paths (LSPs) ................................................................. 6
  2.1.4 Label Distribution Protocols ............................................................................ 6
  2.1.5 MPLS Operation ............................................................................................... 7
Section Two: Traffic Engineering with MPLS ............................................................... 9
  2.2.1 Introduction ....................................................................................................... 9
  2.2.2 Definition: ......................................................................................................... 9
  2.2.3 Traditional routed core for traffic engineering .................................................... 9
      2.2.3.1 Limitations of a traditional routed core for traffic engineering .......... 10
  2.2.4 Traffic Engineering Through an ATM Overlay Network ..................................... 10
      2.2.4.1 ATM Overlay benefits .......................................................................... 11
Abstract

The momentum toward voice and data convergence, each with their own unique set of requirements, is driving the Internet to cope with new realities. On top of the traditional data traffic, the addition of Hyper-Text Transfer Protocol (HTTP), voice, store and forward messaging, multimedia traffic and real-time e-commerce applications to the infrastructure are pushing toward ever-higher bandwidth requirements, as well as the ability to guarantee that bandwidth. There for the need for a true multi-service network that can support multiple different applications via a single carrier arises. This network is the next generation network (NGN).

Traffic engineering (TE) tools are designed to control the traffic flow through a network, offering services according to customers’ specific requirements while using network resources efficiently and economically. Carriers are using several mechanisms for TE. This project discusses and analyzes one of the most recent mechanisms called Multi-Protocol Label Switching (MPLS).

First we studied the main features of this new mechanisms and its differences from the existing ones what it gives more to core network and how it can affect and support traffic engineering in IP networks.

We then make a simulation program that simulates the operation of the MPLS network and we use this simulation to analyze the performance of the MPLS network in term of mean packet delay, throughput and lost packets. We also applied two mechanisms in the simulation network “Rerouting and re_optimization” as an example of traffic engineering features used in MPLS network.

The network simulation was implemented using C++ language, reasonable results were obtained showing that MPLS has a good performance and flexibility to support next generation network.
الخلاصة

إن الا تجاه نحو دمج البيانات والصوت، كل خصائصه ومتطلباته الفردية الخاصة يتقدم الإنترنت لم عالج حقائق جديدة. إن إضافة نظام إرسال النصوص الموصولة (HTTP) والصوت وتطبيقات الوسائط المتعددة والتجارة الإلكترونية يتطلب ساعات أكبر وضمان أكبر لعدم انقطاع الخدمة. ومن هنا جاءت الحاجة لشبكة متعددة خدمات يُمكنها أن تتعمّم تطبيقات مختلفة ومتنوعة عن طريق ناقل واحد. هذه الشبكة هي ما يعرف بالجيل المستقبل من الشبكات (Next Generation Network-NGN).

هندسة المرور هي عدد من المفاهيم والآليات المستخدمة في الشبكات لضمان عملها بكفاءة وبأقل تكلفة. هذا المشروع يناقش ويحلل أحد هذه الآليات (MPLS).

أولاً درستنا الميزات الرئيسية لهذه الآليات الجديدة ونلاحظ أنه الإتقانات السابقة ثم قمنا بعمل برنامج محاكاة يُقلّد عمل شبكة MPLS وقمنا باستخدام هذه المحاكاة لتَحليل أداء هذه الشبكة. طبّقتنا أيضاً آلية محاكاة كمثال لميزات هندسة المرور في سلسلة MPLS. تم استخدام لغة C++ لبرنامج المحاكاة، ثم الوصول إلى نتائج معقولة بينت إلى حد كبير فعالية عمل شبكة (MPLS).
## List of Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM</td>
<td>Asynchronous Transfer Mode</td>
</tr>
<tr>
<td>BGP</td>
<td>Border Gateway Protocol</td>
</tr>
<tr>
<td>CoS</td>
<td>Class of Service</td>
</tr>
<tr>
<td>CR-LDP</td>
<td>Constraint-based Routing Label Distribution Protocol</td>
</tr>
<tr>
<td>DLCI</td>
<td>Data Link Connection Identifier</td>
</tr>
<tr>
<td>DWDM</td>
<td>Dense Wavelength Division Multiplexing</td>
</tr>
<tr>
<td>EGP</td>
<td>Exterior Gateway Protocol</td>
</tr>
<tr>
<td>FEC</td>
<td>Forwarding Equivalence Class</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IGP</td>
<td>Interior Gateway Protocol</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
</tr>
<tr>
<td>L2</td>
<td>Layer 2</td>
</tr>
<tr>
<td>L3</td>
<td>Layer 3</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LDP</td>
<td>Label Distribution Protocol</td>
</tr>
<tr>
<td>LER</td>
<td>Label Edge Router</td>
</tr>
<tr>
<td>LIB</td>
<td>Label Information Base</td>
</tr>
<tr>
<td>LSP</td>
<td>Label Switch Path</td>
</tr>
<tr>
<td>LSR</td>
<td>Label Switch Router</td>
</tr>
<tr>
<td>MPOA</td>
<td>Multi-Protocol over ATM</td>
</tr>
<tr>
<td>MPLS</td>
<td>Multi-Protocol Label Switching</td>
</tr>
<tr>
<td>OSPF</td>
<td>Open Shortest Path First</td>
</tr>
<tr>
<td>PNNI</td>
<td>Private Network-to-Network Interface</td>
</tr>
<tr>
<td>PVC</td>
<td>Permanent Virtual Circuit</td>
</tr>
</tbody>
</table>
Chapter One

Introduction
Introduction

The explosion of data in all its forms and new applications such as voice and multimedia services have driven the demand for increased and guaranteed bandwidth requirements in the backbone of the network.

Also the ever increasing number of users (over 250 million in the next decade) using resource-consuming applications raised the need for service providers to handle QoS in the networks and to perform traffic engineering to enhance network utilization. Keeping in mind that the Internet will continue to see dramatic growth due to the ever increase in demand for more bandwidth the need for new traffic engineering techniques becomes more essential.

1.1 Problems with existed networks:

Carriers and service providers’ core networks run on impressive asynchronous transfer mode (ATM) backbones. Connections to these providers continue to be slow frame relay and point-to-point connections. This introducing latency and sometimes bottlenecks at the edge access points. Core network routers also contribute to latencies as each must make its own individual decision on the best way to forward each incoming packet.

Traditionally IP has been routed over ATM using IP over ATM via virtual circuits (VCs) or multi-protocol over ATM (MPOA). These forwarding methods proved to be cumbersome and complicated.

1.2 Solution:

The need for a simpler forwarding method one with the traffic management features and performance of traditional switches combined with the forwarding intelligence of a router is arise to support the next generation network.

All of these needs can be met with multi-protocol label switching (MPLS) because it integrates the key features of both Layer 2 and Layer 3. Most importantly it is not limited to any Layer 2 or Layer 3 protocol. In particular, MPLS has several applications and can be extended across multiple product segments.

This project studies the MPLS as one of the most powerful forwarding mechanism which shows great promise for traffic engineering. We developed a simulation program that perform the
operation of the MPLS network and use this simulator to do many test to the MPLS network and get results. The resultants are used as a performance indicator of the MPLS network.

### 1.3 Report layout

**Chapter 2**

This chapter is divided into 2 sections the first one introduces the concepts of MPLS and its basic components, the second one introduced the general concepts of traffic engineering (TE) and how MPLS support TE in the next generation network.

**Chapter 3**

Contains general description of the simulation program and its functions and how the program works. Flow charts of the program algorithm are also provided.

**Chapter 4**

Contains the result obtained from the simulation program.

**Chapter 5**

Conclusion and future work.
Chapter Two

Section One

MPLS concepts and components

Section Two

Traffic Engineering with MPLS
Section One
MPLS concepts and components

2.1.1 Introduction

MPLS is the latest step in the evolution of multilayer switching in the Internet. MPLS is based on the concept of label switching, an independent and unique “label” is added to each data packet and this label is used to switch and route the packet through the network. This concept has been around the data communications industry for years. X.25, Frame Relay, and ATM are examples of label switching technologies.

"Several label switching initiatives emerged in the mid-1990’s to improve the performance of software-based IP routers and provide Quality of Service (QoS). Among these were IP Switching (Ipsilon/Nokia), Tag Switching (Cisco), and ARIS (IBM). In early 1997, an Internet Engineering Task Force (IETF) Working Group was chartered to standardize a label switching technology." [1]

Multiprotocol label switching (MPLS) is a versatile solution to address the problems faced by present-day networks—speed, scalability, quality-of-service (QoS) management, and traffic engineering. MPLS has emerged as an elegant.

Solution to meet the bandwidth-management and service requirements for next-generation Internet protocol (IP)–based backbone networks. MPLS addresses issues related to scalability and routing (based on QoS and service quality metrics) and can exist over existing asynchronous transfer mode (ATM) and frame-relay networks.

2.1.2 MPLS and IP

Traditional IP packet forwarding uses the IP destination address in the packet’s header to make an independent forwarding decision at each router in the network. These hop-by-hop decisions are based on network layer routing protocols (layer 3 protocols), such as Open Shortest Path First (OSPF) or Border Gateway Protocol (BGP). These routing protocols are designed to find the shortest path through the network, and do not consider other factors, such as latency or traffic congestion. MPLS creates a connection-based model overlaid onto the traditionally connectionless IP routed networks. This connection-oriented architecture opens the door to new possibilities for
managing traffic on an IP network. MPLS builds on IP, combining the intelligence of routing, which is fundamental to the operation of the Internet and today’s IP networks, with the high performance of switching.

MPLS performs the following functions:

- Specifies mechanisms to manage traffic flows of various granularities, such as flows between different hardware, machines, or even flows between different applications.
- Remains independent of the Layer2 and Layer-3 protocols.
- Provides a means to map IP addresses to simple, fixed-length labels used by different packet-forwarding and packet-switching technologies.
- Interfaces to existing routing protocols such as resource reservation protocol (RSVP) and open shortest path first (OSPF).
- Supports the IP, ATM, and frame-relay Layer-2 protocols.

2.1.3 MPLS Components:

2.1.3.1 LSRs and LERs

The devices that participate in the MPLS protocol mechanisms can be classified into label edge routers (LERs) and label switching routers (LSRs).

An LSR is a high-speed router device in the core of an MPLS network that participates in the establishment of LSPs using the appropriate label signaling protocol and high-speed switching of the data traffic based on the established paths.

"An LER is a device that operates at the edge of the access network and MPLS network. LERs support multiple ports connected to dissimilar networks (such as frame relay, ATM, and Ethernet) and forwards this traffic on to the MPLS network after establishing LSPs, using the label signaling protocol at the ingress and distributing the traffic back to the access networks at the egress. The LER plays a very important role in the assignment and removal of labels, as traffic enters or exits an MPLS network." [2]
2.1.3.2 FEC

"The forward equivalence class (FEC) is a representation of a group of packets that share the same requirements for their transport. All packets in such a group are provided the same treatment in route to the destination. As opposed to conventional IP forwarding, in MPLS, the assignment of a particular packet to a particular FEC is done just once, as the packet enters the network. FECs are based on service requirements for a given set of packets or simply for an address prefix. Each LSR builds a table to specify how a packet must be forwarded. This table, called a label information base (LIB), is comprised of FEC–to-label bindings." [2]

2.1.3.3 Label Information Base

"As the network is established and signaled, each MPLS router builds a Label Information Base (LIB)—a table that specifies how to forward a packet. This table associates each label with its corresponding FEC and the outbound port to forward the packet to. This LIB is typically established in addition to the routing table and Forwarding Information Base (FIB) that traditional routers maintain."[1]

2.1.3.4 Labels and Label Bindings

A label, in its simplest form, identifies the path a packet should traverse. A label is carried or encapsulated in a Layer-2 header along with the packet. The receiving router examines the packet for its label content to determine the next hop. Once a packet has been labeled, the rest of the journey of the packet through the backbone is based on label switching. The label values are of local significance only, meaning that they pertain only to hops between LSRs.

Once a packet has been classified as a new or existing FEC, a label is assigned to the packet. The label values are derived from the underlying data link layer. For data link layers (such as frame relay or ATM), Layer-2 identifiers, such as data link connection identifiers (DLCIs) in the case of frame-relay networks or virtual path identifiers (VPIs)/virtual channel identifiers (VCIs) in case of ATM networks, can be used directly as labels. The packets are then forwarded based on their label value.

Labels are bound to an FEC as a result of some event or policy that indicates a need for such binding. These events can be either data-driven bindings or control-driven bindings. The latter is preferable because of its advanced scaling properties that can be used in MPLS.
Label assignment decisions may be based on forwarding criteria such as the following:

- Destination unicast routing.
- Traffic engineering.
- Virtual private network (VPN).
- QoS.

The generic label format is illustrated in figure 2.1.

![Figure 2.1 MPLS Generic Label Format](image)

2.1.3.5 Label-Switched Paths (LSPs)

Within an MPLS domain, a path is set up for a given packet to travel based on an FEC. The LSP is set up prior to data transmission. MPLS provides the following two options to set up an LSP.

- **hop-by-hop routing:** Each LSR independently selects the next hop for a given FEC. This methodology is similar to that currently used in IP networks. The LSR uses any available routing protocols, such as OSPF, ATM private network-to-network interface (PNNI), etc.

- **explicit routing:** Explicit routing is similar to source routing. The ingress LER (i.e., the LER where the data flow to the network first starts) specifies the list of nodes through which the ER–LSP traverses.

The LSP setup for an FEC is unidirectional in nature. The return traffic must take another LSP.

2.1.4 Label Distribution Protocols

MPLS architecture does have a single method of signaling for label distribution. Existing routing protocols, such as the border gateway protocol (BGP), have been enhanced to piggyback the label information within the contents of the protocol. The RSVP has also been extended to support piggybacked exchange of labels. The Internet Engineering Task Force (IETF) has also defined a
new protocol known as the label distribution protocol (LDP) for explicit signaling and management of the label space. Extensions to the base LDP protocol have also been defined to support explicit routing based on QoS and CoS requirements. These extensions are captured in the constraint-based routing (CR)–LDP protocol definition.

A summary of the various schemes for label exchange is as follows:

- LDP: maps unicast IP destinations into labels.
- RSVP, CR–LDP: used for traffic engineering and resource reservation.

2.1.5 MPLS Operation

"MPLS clearly separates the label based forwarding plane from the routing protocol control plane, so each one can be independently developed and modified." [3]

- The control component

  It used standard routing protocols to exchange information with other router to build and maintain a forward table. The control component centers around IP and must react when network changes such as link failure occur, but it is not involved in the processing of individual packet. Standard routing protocols are used to exchange routing information among the control component of the LSRs.

- The forwarding component

  As packet arrives forwarding component search the forward table to make a routing decision for each packet (examined information contained in the packet’s header and a set of local procedures) the forwarding component of LSR uses an exact match label swapping algorithm that uses the label in the packet and a label-based forwarding table to obtain a new label and output interface for the packet.

  Figure 2.2 shows the separation between the control component and the forwarding component in an MPLS domain and how does these two components work together to forward packets through the network.
Sequence of events is:

The network automatically builds routing tables participate in interior gateway protocols, such as OSPF, throughout the service provider network. LDP uses the routing topology in the tables to establish label values between adjacent devices. This operation creates LSPs, preconfigured maps between destination endpoints.

- A packet enters the ingress Edge LSR where it is processed to determine which Layer 3 services it requires, such as QoS and bandwidth management. Based on routing and policy requirements, the Edge LSR selects and applies a label to the packet header and forwards the packet.

- The LSR in the core reads the label on each packet, replaces it with a new one as listed in the table, and forwards the packet. This action is repeated at all core "hops."

- The egress Edge LSR strips the label, reads the packet header, and forwards it to its final destination.
2.2.1 Introduction

When the Internet began its growth, the demand for bandwidth increased faster than the speed of individual network links. Service providers responded to this challenge by simply provisioning more links to provide additional bandwidth. At this point, TE in the router-based core became even more important to service providers because it would enable them to use the aggregated bandwidth more efficiently by load balancing the traffic over the available parallel or alternative paths.

2.2.2 Definition:

Traffic engineering is a process that enhances overall network utilization by attempting to create a uniform or differentiated distribution of traffic throughout the network. An important result of this process is the avoidance of congestion on any one path. It is important to note that traffic engineering does not necessarily select the shortest path between two devices. It is possible that, for two packet data flows, the packets may traverse completely different paths even though their originating node and the final destination node are the same. This way, the less exposed or less-used network segments can be used and differentiated services can be provided.

The purpose of Traffic Engineering is to make the best possible use of existing resources.

2.2.3 Traditional routed core for traffic engineering

In IP router-based cores, TE was achieved by simply manipulating routing metrics. Metric-based control was adequate because Internet backbones were much smaller in terms of the number of routers, number of links, and amount of traffic.

"In Figure 2.3, assume that Network A sends a large amount of traffic to Network C and Network D. With the metrics shown in Figure 2.3, Links 1 and 2 might become congested because both the Network A-to-Network C and the Network A-to-Network D flows transit those links. If the
network operator changed Link 4’s metric to “2”, the Network A-to-Network D flow would be moved to Link 4, but the Network A-to- Network C flow would remain on Links 1 and 2. "[2]

![Network Diagram]

Figure 2.3 Traditional routed core metric manipulation

### 2.2.3.1 Limitations of a traditional routed core for traffic engineering

- TE based on metric manipulation is not scalable. As core networks become more richly connected (that is, bigger, more thickly meshed and more redundant), it is difficult to ensure that a metric adjustment in one part of the network does not cause problems in another part.

- TE based on metric manipulation offers a trial-and-error approach rather than a scientific solution to an increasingly complex problem.

- IGP route calculation is topology driven and based on a simple additive metric such as the hop count or an administrative value. IGPs do not distribute information such as bandwidth availability or traffic characteristics, so the traffic load on the network is not taken into account when the IGP calculates its forwarding table. As a result, traffic is not evenly distributed across the network’s links, causing inefficient use of expensive resources. Some of the links could become congested, while other links remain underutilized.

### 2.2.4 Traffic Engineering Through an ATM Overlay Network

"When IP runs over an ATM network, routers surround the edge of the ATM cloud. Each router communicates with every other router by a set of permanent virtual circuits (PVCs) that are configured across the ATM physical topology. The PVCs function as logical circuits, providing connectivity between edge routers. The routers do not have direct access to information describing the physical topology of the underlying ATM infrastructure; they recognize only the individual
PVCs that appear to them as simple point-to-point circuits between two routers. Figure 2.4 illustrates how the physical topology of an ATM core differs from a logical IP overlay topology."

2.2.4.1 ATM Overlay benefits

- ATM offers precise control over traffic as it flows across the core network.
- ATM core is also capable of transporting delay-sensitive broadband applications.
- ATM-based core fully supports TE.
- Per PVC statistics provided by ATM switches enable operators to monitor traffic patterns for optimal PVC placement and management.

2.2.4.2 Limitations of ATM Overlay for traffic engineering

- Extra network devices (cost).
- More complex network management (cost).
  - Two-level network without integrated network management.
  - Additional training, technical support, field engineering.
- IGP routing scalability issue for meshes.
- Additional bandwidth overhead (“cell tax”).
2.2.5 **MPLS solution**

MPLS brings connection-oriented forwarding techniques together with the Internet’s routing protocols by establishing a virtual connection between two points on an IP network. The simplicity and flexibility of an IP network remain intact, while the ATM-like advantage of a connection-oriented network is included. This hybrid architecture can emulate connection-oriented services, but normal datagram mechanisms are used to deliver IP services. **MPLS** provides simple and efficient support for explicit routing.

Advantages of explicit routing are:

- Operator has routing flexibility (policy-based, QoS-based).
- Can use routes other than shortest path.
- Can compute routes based on constraints.
- LSPs can be ranked so some reroute very quickly and/or backup paths may be pre-provisioned for rapid restoration.

The attractiveness of **MPLS** for Traffic Engineering can be returned to the following factors:

- Explicit label switched paths which are not constrained by the destination based forwarding paradigm can be easily created through manual administrative action or through automated action by the underlying protocols.
- LSPs can potentially be efficiently maintained,
- Traffic trunks can be instantiated and mapped onto LSPs,
- A set of attributes can be associated with traffic trunks which modulate their behavioral characteristics.
- A set of attributes can be associated with resources which constrain the placement of LSPs and traffic trunks across them.
- **MPLS** allows for both traffic aggregation and disaggregation whereas classical destination only based IP forwarding permits only aggregation.
- It is relatively easy to integrate a "constraint based routing" framework with **MPLS**,
A good implementation of MPLS can offer significantly lower overhead than competing alternatives for Traffic Engineering. Furthermore, through explicit routes, MPLS permits a quasi circuit switching capability to be superimposed on the current Internet routing model.

### 2.2.6 MPLS Traffic engineering requirements

- **Differentiating traffic trunks:**
  
  Large, ‘critical’ traffic trunks must be well routed in preference to other trunks.

- **Handling failures:**
  
  Automated re-routing in the presence of failures.

- **Pre-configured paths:**
  
  For use in conjunction with the off-line route computation procedures

- **Support of multiple Classes of Service**

- **re-optimize on new/restored bandwidth**

- **Ability to “spread” traffic trunk across multiple Label Switched Paths (LSPs).**

### 2.2.7 MPLS protocols for Traffic Engineering:

There are currently two label distribution protocols that provide support for Traffic Engineering:

- **Resource ReSerVation Protocol (RSVP)**

- **Constraint-based Routed Label Distribution Protocol (CR-LDP).**

#### 2.2.7.1 Constrained-based Routing LDP (CR-LDP):

- An extension to the base LDP protocol defined to support explicit routing based on QoS and CoS requirements.

- CR-LDP takes into account parameters, such as link characteristics (bandwidth, delay, etc.), hop count, and QoS.

- When using CR, it is entirely possible that a longer (in terms of cost) but less loaded path is selected.
• However, while CR increases network utilization, it adds more complexity to routing calculations, as the path selected must satisfy the QoS requirements of the LSP.

2.2.7.2 Resource Reservation Protocol-Traffic Engineering (RSVP-TE):

• RSVP-TE is an extension of the existing RSVP protocol.

• For LSP setup, the protocol exchanges PATH messages which do not reserve any resources in the network element but are intended to communicate resource requirements towards the end station.

• PATH messages include information with respect to :
  
  Peak data rate, average data rate, burst size, minimum and maximum packet size.
Chapter Three
Simulation Program
Simulation Program

3.1 Introduction

In this chapter we will talk about the simulation program of our MPLS network. The program was written using C++ OOP. The simulated network topology consists of 3 label edge routers (LERs) and 3 label switch routers (LSRs). Each label edge router connected with a end-terminal (USER) which acts as a traffic source to the network. The figure below shows the topology of the simulated network.

![Figure 3.1: Topology of the simulated network](image)

3.2 Program Assumptions:

- No queuing in the receive buffer.
- Propagation delay is negligible.
- Because of MPLS is very fast, processing delay is very small, thus it was neglected.
- All packet with the same size.
3.3 General description of the Program

In the program we first defined number of structures that will be used in the program. These structures are:

**Packet:** Consists of the following fields:

- **Label:** Contains the value of the label assigned to the packet and is used to forward packet by swapping this label.

- **Source-address:** Contains the IP address of the packet source and can be used to classify a packet in a certain FEC.

- **Destination-address:** Contains the IP address of the destination of the packet used to classify a packet in a certain FEC.

- **Source-port:** Contains the identifier of the port by which the packet will be sent.

- **Destination-port:** Contains the identifier of the port by which the packet will be received.

- **Priority:** Contains the priority of the packet. One means **High Priority**, Zero means **Low Priority**.

- **Sending-time:** Contains the time at which the packet was sent.

**Structure Port:** Consists of two fields:

- **Sending Buffer:** Contains the packets that will be sent.

- **Receiving Buffer:** Contains the received packets.

  (The Buffer is an array of packets)

**Structure Event:** Consists of the following fields:

- **Type:** Event type

- **Node_Id:** Identification number of the node which will perform this event

- **Time:** Time on which the event will take place

- **Pac:** The packet which will be included in this event or procedure.
3.4 Program Classes

OOP has been used in our program, we have created three classes; USER, LER and LSR as will be explained next.

Class USER:

It acts as a traffic generator and it contains:

- One port and has two functions:
  - Sending (packet)
  - Receiving (packet)

Class LER:

It consists of:

- Three ports each one has a counter to count the number of packets in the sending buffer of that port.
- Label information base table.
- Member functions in this class are:
  - Receiving (packet)
  - Processing (packet)
  - Sending (packet)
  - Get-table (table)
  - Shift-buffer

Class LSR:

It consists of:

- Four ports each one has a counter to count the number of packets in the sending buffer of that port.
- Table [20][5] : Label information base table.
• Member functions in this class are:
  o Receiving (packet)
  o Processing(packet)
  o Sending (packet)
  o Get-table (table)
  o Shift-buffer

**LER and LSR Functions:**

The general work of these classes functions are the same with some little differences. So, the general ideas about these functions are discussed below.

• **Receive(packet):**

It receives a new packet and places it in the receiving-buffer of the receiving port and then immediately sends this packet to be processed.

After the packet has been processed by the process function then it is placed in the sending-buffer of the port by which it is going to be send “source-port of the packet”.

Then the packet will be scheduled so as to be send when the resource is available for it.

• **Process(packet):**

It checks the label of the received packet if it has no label; it assigns to it a label by searching the **LIB** table using the packet destination address as a key entry.

If there is a label “the packet is already assigned to a label” this label is used as an entry in the **LIB** table to replace this label by a new one. Swapping label – in by label - out.

Then the processed packet is retuned to the forward function. And it will be placed in the sending-buffer of the port by which it’s going to be send “source-port of the packet”, the packet will be scheduled so as to be send when the resource is available for it.

• **Send(packet):**

It first shifts the sending buffer of the source port “meaning that the packet has been sent completely” and then it calls the forward function of the next node “router” to receive this packet. “Assuming that there is no propagation delay”
• Get-table(ID) :
This function updates the LIB tables in the node “router”. We use this function to update the entire LIB table in all the nodes periodically.

• Shift-buffer(No) :
It shifts the sending-buffer after sending a packet from it.

3.5 Main Program:

In addition to previous defined structures and classes, the main program deals with other global features and functions, these are discussed below.

• Event-list:
  It is an array of type event, it contains the list of events waiting to be executed and the time in which each event will be executed.

• General functions:
  o Schedule (event):
  It adds a new event to the event list and sort the events in the list according to their times. The event with the most recent time will be in the top of the list.
  o Do-event (event):
  It takes the first event in the list and does it “perform it, execute it”.
  o Shift-list:
  This function shifts the event-list after executing an event “after each event is done”.
  o Generate –traffic():
  It generates traffic to the network by sending packets from the user edges with [exponentially distributed] inter-arrival- time between packets.
  It schedules these packets to be sent each in its sending- time. The number of sent packets is according to an arrival rate.
3.6 The Simulation Algorithm:

We start and terminate our simulation by a virtual clock \{CK\} which starts from \( CK = 1 \) and ends with \( CK = \) termination time(TE).

When the clock is updated to a new value \( t \) the following steps are performed:

- Calling the function ‘generate –traffic’ which starts sending packets from the USER edge of the network and then these packets are forwarded through the network taking a certain path.

- The program checks the first event in the event list and see if the event time is equal to \( t \) or not, if it is equal to "t" the program calls the function Do_Event to execute this Event and then shifts the Event_list, this process is repeated again till the time of the first event is not equal to the clock time "t" when this happens, the program updates the clock to the next value \( CK = t + 1 \) and repeats all the previous steps.

The sequence of events is controlled by the event-list and the scheduled function, which adds new events to the event_list and sorts the events in the list according to their times.

When the clock reaches the termination time "TE" this means that the program is ended meaning that the simulation is stopped.

Re-route Algorithm

We made a mechanism to reroute traffic from the existing LSP to another one when a failure occurs and return the traffic to old LSP after it is repaired.

To perform this mechanism we defined a new variable "LSP-CO" which gives the cost of certain LSP at any time, and it is updated to the recent cost of the path periodically. We also defined two functions:

- \text{Ch\_LSP()}   
- \text{Reroute()}

\text{Ch\_LSP():}

This function compares the cost of each working LSP by a margin value "M" if the cost of the path exceeds this value this indicates that there is a problem with this path and we have to reroute to a new path by calling function reroute. Once the LSP cost returns to a value less than "M", the traffic is rerouted again to the old LSP.
Re-route():

This function is called by the function ch_LSP to reroute the traffic from a LSP to another one. It updates the LIB tables in all the nodes to carry the traffic by the new path "LSP", once the tables has been updated the traffic is automatically swapped to the new route.

re-optimize Algorithm

We use the same functions used in Reroute to perform the re-optimization mechanism.
Start

Enter terminated time (T)

Clock (ck) = 1

If clock = T

Yes
End

No
Check event list

If first event time = ck

Yes
Generate traffic

No
Ck ++

Do event

Shift event list
Figure 4.2: flow chart for the Do event function
Figure 4.3: Flow chart for the process function
Start

Enter arrival rate (a)

Enter mean inter-arrival time (m)

J = 1, t = 0

If J <= a

K = 1 - random(999)/1000

n = -m* (lnk)

t = t + n

Sent time = t

Schedule user send

Return to main program

Figure 4.4: Flow chart for the generate traffic function
Chapter Four
Results and Analysis
Results and Analysis

The network simulation program is tested so many times with changing a particular criterion at each time to see how this criterion will affect on the MPLS network performance. MPLS network performance is measured in terms of throughput, loss and delay time.

Three important criteria have been tested, buffer size of each node, speed of each node and the arrival rate (the rate by which packets arrive to the MPLS network).

As we can see from the results we obtained which were listed in tables below and also illustrated with figures, in all cases the throughput of high priority packets is higher than the throughput of the low priority packets. In addition, the delay time of the high priority packets is lower than the delay time of the low priority packets.

4.1 Effect of changing buffer size:

We change the size of the buffer for values between 2 – 50 with step of 2 at each time and record the corresponding values of throughput, loss and delay time. The results are shown in table 1 below:

Table 1: variation of throughput, loss and delay time with variation of buffer size

<table>
<thead>
<tr>
<th>Buffer size</th>
<th>low priority delay</th>
<th>high priority delay</th>
<th>mean delay</th>
<th>low priority throughput</th>
<th>high priority throughput</th>
<th>total throughput</th>
<th>Total-loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>476</td>
<td>704</td>
<td>1180</td>
<td>1320</td>
</tr>
<tr>
<td>4</td>
<td>0.38</td>
<td>0.29</td>
<td>0.33</td>
<td>564</td>
<td>854</td>
<td>1418</td>
<td>1082</td>
</tr>
<tr>
<td>6</td>
<td>0.83</td>
<td>0.63</td>
<td>0.73</td>
<td>622</td>
<td>930</td>
<td>1552</td>
<td>948</td>
</tr>
<tr>
<td>8</td>
<td>1.24</td>
<td>0.96</td>
<td>1.1</td>
<td>679</td>
<td>1019</td>
<td>1698</td>
<td>802</td>
</tr>
<tr>
<td>10</td>
<td>1.91</td>
<td>1.34</td>
<td>1.63</td>
<td>715</td>
<td>1058</td>
<td>1773</td>
<td>727</td>
</tr>
<tr>
<td>12</td>
<td>2.78</td>
<td>1.89</td>
<td>2.33</td>
<td>735</td>
<td>1101</td>
<td>1836</td>
<td>664</td>
</tr>
<tr>
<td>14</td>
<td>3.41</td>
<td>2.25</td>
<td>2.83</td>
<td>753</td>
<td>1125</td>
<td>1878</td>
<td>622</td>
</tr>
<tr>
<td>16</td>
<td>4.19</td>
<td>2.61</td>
<td>3.4</td>
<td>759</td>
<td>1151</td>
<td>1910</td>
<td>590</td>
</tr>
<tr>
<td>18</td>
<td>4.92</td>
<td>2.88</td>
<td>3.9</td>
<td>774</td>
<td>1170</td>
<td>1944</td>
<td>556</td>
</tr>
</tbody>
</table>
The results obtained in the table above are illustrated with figures; figure 3.1.a below shows the variation of the throughput with the buffer size. We can see that the throughput increases with the increase of buffer size when the arrival rate and the speed of different nodes are constants; until it reaches a maximum value then it becomes constant. This because the increase of buffer size means that each node has enough capacity to receive higher number of packets (i.e. small number of packets will be discarded).

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>5.61</td>
<td>3.23</td>
<td>4.42</td>
<td>780</td>
<td>1184</td>
<td>1964</td>
<td>536</td>
</tr>
<tr>
<td>22</td>
<td>6.3</td>
<td>3.5</td>
<td>4.9</td>
<td>793</td>
<td>1200</td>
<td>1993</td>
<td>507</td>
</tr>
<tr>
<td>24</td>
<td>7.07</td>
<td>3.84</td>
<td>5.45</td>
<td>799</td>
<td>1214</td>
<td>2013</td>
<td>487</td>
</tr>
<tr>
<td>26</td>
<td>7.94</td>
<td>4.19</td>
<td>6.17</td>
<td>809</td>
<td>1225</td>
<td>2034</td>
<td>466</td>
</tr>
<tr>
<td>28</td>
<td>8.67</td>
<td>4.47</td>
<td>6.57</td>
<td>816</td>
<td>1237</td>
<td>2053</td>
<td>447</td>
</tr>
<tr>
<td>30</td>
<td>9.29</td>
<td>4.72</td>
<td>7.0</td>
<td>821</td>
<td>1245</td>
<td>2078</td>
<td>422</td>
</tr>
<tr>
<td>32</td>
<td>9.68</td>
<td>4.91</td>
<td>7.29</td>
<td>825</td>
<td>1253</td>
<td>2091</td>
<td>409</td>
</tr>
<tr>
<td>34</td>
<td>10.87</td>
<td>5.17</td>
<td>7.67</td>
<td>827</td>
<td>1264</td>
<td>2091</td>
<td>409</td>
</tr>
<tr>
<td>36</td>
<td>10.62</td>
<td>5.33</td>
<td>7.97</td>
<td>833</td>
<td>1270</td>
<td>2103</td>
<td>397</td>
</tr>
<tr>
<td>38</td>
<td>11.04</td>
<td>5.5</td>
<td>8.27</td>
<td>838</td>
<td>1277</td>
<td>2115</td>
<td>385</td>
</tr>
<tr>
<td>40</td>
<td>11.46</td>
<td>5.62</td>
<td>8.54</td>
<td>841</td>
<td>1278</td>
<td>2119</td>
<td>381</td>
</tr>
<tr>
<td>42</td>
<td>11.73</td>
<td>5.74</td>
<td>8.74</td>
<td>843</td>
<td>1282</td>
<td>2125</td>
<td>375</td>
</tr>
<tr>
<td>44</td>
<td>12.13</td>
<td>5.86</td>
<td>9</td>
<td>846</td>
<td>1285</td>
<td>2131</td>
<td>369</td>
</tr>
<tr>
<td>46</td>
<td>12.48</td>
<td>5.99</td>
<td>9.23</td>
<td>847</td>
<td>1289</td>
<td>2136</td>
<td>364</td>
</tr>
<tr>
<td>48</td>
<td>12.68</td>
<td>6.07</td>
<td>9.37</td>
<td>848</td>
<td>1292</td>
<td>2140</td>
<td>360</td>
</tr>
<tr>
<td>50</td>
<td>12.93</td>
<td>6.15</td>
<td>9.54</td>
<td>850</td>
<td>1294</td>
<td>2144</td>
<td>356</td>
</tr>
</tbody>
</table>

Figure 3.1.a: variation of throughput with buffer size
Figure 3.1.b shows the variation of the loss according to the variation of the buffer size. As we can see from figure, the number of discarded packets decreases due to the increasing in the buffer size.

![Loss vs. Buffer size](image)

Figure 3.1.b: variation of loss with buffer size

Figure 3.1.c shows the variation of delay time with the buffer size. The delay time increases as the buffer size increases; this is due to the increasing in the queuing delay (i.e. more number of packets will queue in the buffer).

![Delay vs. Buffer size](image)

Figure 3.1.c: variation of delay with the throughput
4.2 Effect of changing router speed (i.e. link bandwidth):

At this time, we change the speed of different nodes for values between 1 – 20 with step of 1 at each time, and see how this variation affects the throughput, loss and delay time. The results are shown in table 3.2 below:

Table 3.2: variation of throughput, loss, and delay time with the router speed

<table>
<thead>
<tr>
<th>Router speed</th>
<th>Low priority delay</th>
<th>High priority delay</th>
<th>mean delay</th>
<th>Low priority throughput</th>
<th>High priority throughput</th>
<th>Total throughput</th>
<th>Total-loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>73.58</td>
<td>32.66</td>
<td>53.12</td>
<td>270</td>
<td>448</td>
<td>718</td>
<td>1782</td>
</tr>
<tr>
<td>2</td>
<td>30.54</td>
<td>12.74</td>
<td>21.64</td>
<td>602</td>
<td>905</td>
<td>1507</td>
<td>993</td>
</tr>
<tr>
<td>3</td>
<td>9.28</td>
<td>4.72</td>
<td>7</td>
<td>821</td>
<td>1245</td>
<td>2066</td>
<td>434</td>
</tr>
<tr>
<td>4</td>
<td>2.55</td>
<td>1.42</td>
<td>1.99</td>
<td>874</td>
<td>1332</td>
<td>2206</td>
<td>294</td>
</tr>
<tr>
<td>5</td>
<td>0.98</td>
<td>0.53</td>
<td>0.76</td>
<td>897</td>
<td>1347</td>
<td>2244</td>
<td>256</td>
</tr>
<tr>
<td>6</td>
<td>0.52</td>
<td>0.25</td>
<td>0.39</td>
<td>898</td>
<td>1348</td>
<td>2246</td>
<td>254</td>
</tr>
<tr>
<td>7</td>
<td>0.32</td>
<td>0.14</td>
<td>0.23</td>
<td>898</td>
<td>1348</td>
<td>2246</td>
<td>254</td>
</tr>
<tr>
<td>8</td>
<td>0.22</td>
<td>0.08</td>
<td>0.15</td>
<td>898</td>
<td>1348</td>
<td>2246</td>
<td>254</td>
</tr>
<tr>
<td>9</td>
<td>0.15</td>
<td>0.05</td>
<td>0.1</td>
<td>898</td>
<td>1348</td>
<td>2246</td>
<td>254</td>
</tr>
<tr>
<td>10</td>
<td>0.1</td>
<td>0.03</td>
<td>0.07</td>
<td>898</td>
<td>1348</td>
<td>2246</td>
<td>254</td>
</tr>
<tr>
<td>11</td>
<td>0.08</td>
<td>0.02</td>
<td>0.05</td>
<td>898</td>
<td>1348</td>
<td>2246</td>
<td>254</td>
</tr>
<tr>
<td>12</td>
<td>0.06</td>
<td>0.01</td>
<td>0.04</td>
<td>898</td>
<td>1348</td>
<td>2246</td>
<td>254</td>
</tr>
<tr>
<td>13</td>
<td>0.04</td>
<td>0.007</td>
<td>0.025</td>
<td>898</td>
<td>1348</td>
<td>2246</td>
<td>254</td>
</tr>
<tr>
<td>14</td>
<td>0.04</td>
<td>0.004</td>
<td>0.021</td>
<td>898</td>
<td>1348</td>
<td>2246</td>
<td>254</td>
</tr>
<tr>
<td>15</td>
<td>0.025</td>
<td>0.002</td>
<td>0.014</td>
<td>898</td>
<td>1348</td>
<td>2246</td>
<td>254</td>
</tr>
<tr>
<td>16</td>
<td>0.02</td>
<td>0.001</td>
<td>0.01</td>
<td>898</td>
<td>1348</td>
<td>2246</td>
<td>254</td>
</tr>
<tr>
<td>17</td>
<td>0.015</td>
<td>0.0007</td>
<td>0.008</td>
<td>898</td>
<td>1348</td>
<td>2246</td>
<td>254</td>
</tr>
<tr>
<td>18</td>
<td>0.01</td>
<td>0</td>
<td>0.005</td>
<td>898</td>
<td>1348</td>
<td>2246</td>
<td>254</td>
</tr>
<tr>
<td>19</td>
<td>0.007</td>
<td>0</td>
<td>0.004</td>
<td>898</td>
<td>1348</td>
<td>2246</td>
<td>254</td>
</tr>
<tr>
<td>20</td>
<td>0.005</td>
<td>0</td>
<td>0.003</td>
<td>898</td>
<td>1348</td>
<td>2246</td>
<td>254</td>
</tr>
</tbody>
</table>
These results are illustrated in the next three figures. Figure 3.2.a shows the variation of the throughput due to variation of the speed of different nodes. We can also notice that throughput is increasing rapidly as the routers' speed increases. Also the throughput will continue to increase up to certain which represent the maximum throughput.

![Throughput vs. Router Speed](image1)

Figure 3.2.a variation of throughput with the router speed

As we expected, figure 3.2.b, which shows the variation of loss according to variation of the routers' speed – is an inversion of the previous figure (i.e. throughput vs. router speed) since the loss starts with high value and then decreases rapidly as the routers' speed increases.

![Loss vs. router speed](image2)

Figure 3.2.b: variation of loss with the router speed
Figure 3.2.c shows the variation of the delay time with variation of routers' speed. As the speed increases the delay time decrease until become almost zero with high speeds.

Figure 3.2.c: variation of delay with router speed

4.3 Effect of changing the arrival rate:

Lastly we change the arrival rate for values between 2 – 60 and record the corresponding values of throughput, loss and delay time. Results are shown in table 3.3 below:

Table 3.3: variation of throughput, loss, and delay with the arrival rate

<table>
<thead>
<tr>
<th>Arrival rate</th>
<th>Low priority delay</th>
<th>High priority delay</th>
<th>Mean delay</th>
<th>Low priority throughput</th>
<th>High priority throughput</th>
<th>Total throughput</th>
<th>loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.01</td>
<td>0</td>
<td>0.005</td>
<td>500</td>
<td>500</td>
<td>1000</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0.1</td>
<td>0.06</td>
<td>0.08</td>
<td>500</td>
<td>1000</td>
<td>1500</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0.37</td>
<td>0.21</td>
<td>0.29</td>
<td>990</td>
<td>991</td>
<td>1981</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>0.98</td>
<td>0.53</td>
<td>0.755</td>
<td>897</td>
<td>1347</td>
<td>2244</td>
<td>246</td>
</tr>
<tr>
<td>6</td>
<td>0.69</td>
<td>0.22</td>
<td>0.455</td>
<td>655</td>
<td>656</td>
<td>1311</td>
<td>1689</td>
</tr>
<tr>
<td>7</td>
<td>0.9</td>
<td>0.4</td>
<td>0.65</td>
<td>357</td>
<td>474</td>
<td>831</td>
<td>2669</td>
</tr>
<tr>
<td>8</td>
<td>0.89</td>
<td>0.28</td>
<td>0.585</td>
<td>286</td>
<td>288</td>
<td>574</td>
<td>4574</td>
</tr>
<tr>
<td>9</td>
<td>1.1</td>
<td>0.47</td>
<td>0.785</td>
<td>206</td>
<td>207</td>
<td>412</td>
<td>4088</td>
</tr>
<tr>
<td>10</td>
<td>1.1</td>
<td>0.34</td>
<td>0.72</td>
<td>211</td>
<td>263</td>
<td>474</td>
<td>4526</td>
</tr>
<tr>
<td>15</td>
<td>1.3</td>
<td>0.51</td>
<td>0.905</td>
<td>106</td>
<td>122</td>
<td>228</td>
<td>7272</td>
</tr>
<tr>
<td>20</td>
<td>1.52</td>
<td>0.47</td>
<td>0.995</td>
<td>89</td>
<td>90</td>
<td>179</td>
<td>9821</td>
</tr>
<tr>
<td>30</td>
<td>1.68</td>
<td>0.52</td>
<td>1.1</td>
<td>53</td>
<td>61</td>
<td>114</td>
<td>14886</td>
</tr>
<tr>
<td>40</td>
<td>1.22</td>
<td>0.56</td>
<td>0.89</td>
<td>31</td>
<td>39</td>
<td>70</td>
<td>19930</td>
</tr>
<tr>
<td>50</td>
<td>0.44</td>
<td>0.23</td>
<td>0.335</td>
<td>18</td>
<td>22</td>
<td>40</td>
<td>24960</td>
</tr>
<tr>
<td>60</td>
<td>0.46</td>
<td>0.14</td>
<td>0.3</td>
<td>13</td>
<td>14</td>
<td>27</td>
<td>29973</td>
</tr>
</tbody>
</table>
The figures below show the results obtained in the last table, figure 3.3.a shows the variation of throughput with the variation of the arrival rate. Throughput begins to increase rapidly with the arrival rate. At certain point throughput stops increasing. After this point, any increase in the arrival rate causes throughput to go down.

![Throughput vs. Arrival rate](image)

Figure 3.3.a: variation of throughput with the arrival rate

Figure 3.3.b shows the variation of loss due to the variation of the arrival rate. As expected from the previous figure, the number of lost packets begins to decrease and after a certain point it will increase rapidly as the arrival rate increases.

![Loss vs. Arrival rate](image)

Figure 3.3.b: variation of loss with the arrival rate
4.4 Effect of applying TE mechanism:

In this test we applied the re-optimized mechanism to see how it will affect the performance of the network, we repeat the previous test of increasing the offered traffic "arrival rate" to the network and observing the amount of traffic that the network can handle "throughput of the network". Results are shown in figure 3.4 below:

![Throughput Vs Arrival rate](image)

Figure 3.4: variation of throughput with the arrival rate when applying re-optimize technique

As shown in figure 3.4 above the throughput first increased with the increase of offered traffic, when the offered traffic reaches a threshold value the network began to be unstable. However, with any more increase in the offered traffic, the network keeping with a certain value avoiding the breakdown status shown in figure 3.3.a in which re-optimize technique isn't applied.

From this results obtained using the re-optimize technique we can estimate how various TE techniques can improve the performance of the network.
Chapter five
Conclusion and Future Work
5.1 Conclusion

At the end of this project we can say that MPLS can be considered as one of the most powerful forwarding mechanism which shows great promise for traffic engineering.

One of the most important benefits of MPLS which make it a good solution for many ISPs is coast saving. MPLS doesn't need a special hardware to operate; instead MPLS can operate on any layer 2 hardware such as ATM or Frame Relay or any other available hardware. As a result to this, ISPs don't have to change their whole infrastructure; they just need to add simple software that supports MPLS to their existing hardware. The future evolutions and applications of MPLS technology is generalized multiprotocol label switching (GMPLS) which will support applying MPLS concepts into optical systems.

During our work in this project in order to complete it, we face many problems which lead us to work very hard deal with those problems. The first problem was the leakage of data sources. During our work we couldn't find any specialized book or reference in MPLS traffic Engineering. We depend mainly on files and white papers which we found in the internet or from persons who worked with MPLS before.

The second problem which we face was working with simulation program. At the beginning of our work we had to learn how to write a simulation program using C++ from the beginning. While the code was growing, for any addition of a new part or changing an existing part of the program, we had to stay with the program for too long times in order to discover errors which were hidden and scattered in the whole program.
5.2 Future work

The simulation program we wrote doesn't contain all features supported by MPLS, but rather it contains only the main features that can make the MPLS simulation network operate as label swapping based mechanism for packet forwarding.

In this project we divided the simulation program into two main components the forwarding component and the control component.

We do our best efforts in the forwarding component to make it reliable and acts as real MPLS network, although there is still a lot of work that can be done to simulate the operation of the MPLS network more accurately.

The control component can be extended to overcome more traffic engineering techniques, We only apply tow of the traffic engineering (TE) techniques in our simulation "re-route and re-optimize", great improvement can be achieved by applying more traffic engineering techniques.
References

Web ProForum Tutorials
Copyright © 2004, Ixia MPLS
http://www.iec.org

Web ProForum Tutorials
http://www.iec.org

Copyright © 2000, Juniper Networks, Inc.

A Paper in the Backbone Network and Business Solution Series
http://www.alcatel.com

Network Working Group
Internet Draft
Daniel O. Awduche
Joe Malcolm
Johnson Agogbua
Mike O'Dell
Jim McManus
http://www.ietf.com

http://www.nortelnetworks.com

http://www.alcatel.com

http://www.cisco.com
[9]. MPLS Tutorial and Operational Experiences-Peter Ashwood-Smith, Bilel Jamoussi, October, 1999 (PPT file)
http://www.nortelnetworks.com
