

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

**ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI[®]



QUALITY OF SERVICE ROUTING

BY

YASSIR OBEID MOHAMMED

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE
In
COMPUTER ENGINEERING

AUGUST 2001

UMI Number: 1407214

UMI[®]

UMI Microform 1407214

**Copyright 2002 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.**

**ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346**

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

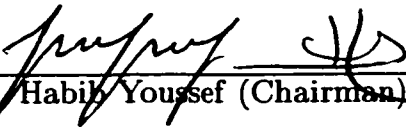
This thesis, written by

YASSIR OBEID MOHAMMED

under the direction of his thesis advisor and approved by his thesis committee,
has been presented to and accepted by the Dean of Graduate Studies, in partial
fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER ENGINEERING


Thesis Committee


Dr. Habib Youssef (Chairman)


Dr. Abdulaziz S. Almulhem (Co – chairman)


Dr. Sadiq M. Sait (Member)


Department Chairman


Dean of Graduate Studies

١٤٤٤/٥/١٧ هـ
Date



TO *My Beloved Mother*

TO *The Spirit of My Maternal Grandmother*

TO *My Dear Father*

TO *The Spirit of My Grandfather*

Acknowledgements

In the name of *God the most beneficial the most merciful*. It is all due to *ALLAH* that I could complete my work and courses and to do every other thing. All praise be to *ALLAH* for his mercy and for giving me the determination to do these all. Prayers and peace to the holy prophet, peace be upon him and his family.

My deepest appreciation and gratitude to my beloved mother to whom I owe all the achievements I have made in my life. I would as well acknowledge my great father. My parents role in my life is undescrivable, their prayers has brightened my life and their acceptance has gifted to me internal peace. I would also acknowledge my brothers and sisters. my parents, brothers, and sisters support and encouragement has strengthened me and their patient and sacrifices enable me to achieve this accomplishment and to carry out my academic career. Words can not express what they did, and words are not enough to thank them. Thanks to our family kids who motivated me and gave me moral support and special thanks to those who stand beside me specially Tayfour and Osman Hamid.

I would like to express my thanks to my thesis advisor, Dr. Habib Yousef, for his patient guidance and support. I also thank my thesis co-advisor, Dr. Abdul Azziz Elmulhim, who for his suggestions, comments and critique. Thanks are to my committee member Dr. Sadiq Sait, who has been helpful and supported me during my work and thesis.

Acknowledgment is due to King Fahd University of Petroleum and Minerals for

providing computing resources for this research.

Thanks to all my fellows and those in the department and the college who directly or indirectly supported and helped me in carrying out this work successfully specially my fellow Atif.

Contents

Acknowledgements	iii
Abstract (English)	xv
Abstract (Arabic)	xvi
1 Introduction	1
1.1 Connection-Oriented and Connectionless Routing	3
1.2 Distance-vector and Link-state Routing	5
1.3 Interior Routing and Exterior Routing	7
1.4 Forwarding IP Packets	11
1.5 Tunneling	14
1.6 Quality of Service Routing	16
1.7 Problem Statement	17
1.8 Proposed Work	19
1.9 Thesis Organization	20

2	Literature Review	21
2.1	Unicast Routing Algorithms	22
2.2	Multicast Routing Algorithms	29
2.3	QoS Routing Algorithms	31
2.4	Mechanisms for Quality of Service Routing	32
2.5	Resource Reservation Protocol	33
2.6	Traffic Prioritization (Class of Service (CoS))	36
2.7	Integrated Services (Intserv)	39
2.8	Differentiated Services (Diffserv)	41
2.9	Multiprotocol Label Switching (MPLS)	47
2.10	Comparison among Intserv, Diffserv, and MPLS	50
3	Multiprotocol Label Switching	56
3.1	Background	57
3.2	Motivation for MPLS	57
3.3	MPLS Terminology	58
3.4	MPLS Header and Label	59
3.5	Label Switching Path (LSP)	60
3.5.1	Single and Multi-Class LSPs	61
3.5.2	Route Selection	62
3.6	Label Assignment and Distribution	63

3.7	Label Distribution Protocols	65
3.7.1	Label Distribution Protocol (LDP)	65
3.7.2	Enhanced RSVP	66
3.7.3	Explicit LSP Setup	68
3.8	MPLS Networks Main Features	69
3.8.1	Partitioning of Functional Units	70
3.8.2	Label Forwarding-Swapping Algorithm	71
3.9	Routing in MPLS Networks	73
3.10	MPLS Applications	77
3.10.1	Traffic Engineering	77
3.10.2	Class of Service (CoS)	81
3.10.3	Virtual Private Networks (VPN)	81
3.11	Summary	82
4	Simulation and Results (1)	83
4.1	Performance Measures	84
4.1.1	Overhead Metrics	84
4.1.2	Latency Metrics	85
4.1.3	Resource Utilization Metrics	86
4.1.4	Flow Satisfaction Metrics	87
4.2	Traffic Types and Characterization	88

4.3	Simulation Environment and Objectives	88
4.4	MPLS Networks versus IP Networks	91
4.4.1	Experimental Model Description	92
4.4.2	Results and Discussion	93
4.5	Diffserv MPLS versus Vanilla MPLS	102
4.5.1	Experimental Model Description	102
4.5.2	Results and Discussion	103
5	Simulation and Results (2): Fault Tolerance in MPLS Networks	115
5.1	Link Failure in Diffserv MPLS	116
5.1.1	Experimental Model Description	116
5.1.2	Effect of Link Failure when Mapping Source 1 traffic to BE . .	117
5.1.3	Effect of Link Failure when Mapping Source 1 Traffic to EF .	120
5.2	Link Failure in Diffserv MPLS with Variable Loads	123
5.2.1	Experimental Model Description	123
5.2.2	Results and Discussion	125
5.2.3	Results for 1.7 Mbps Core Links Network	133
5.2.4	Results for CBR-EF Sources Network	139
5.3	Link Failure in Diffserv MPLS with Multiple LSPs	141
5.3.1	Two alternative LSPs Simulation and Results	142
5.3.2	Multiple Alternative LSPs Simulation and Results	143

6	Conclusions and Future work	145
6.1	Summary	145
6.2	Conclusions	148
6.3	Future Work	149
	Bibliography	150

List of Tables

1.1	Main Types of Routing Protocols.	10
2.1	Priority Values and Traffic Types.	38
2.2	Comparison among intserv, Diffserv, and MPLS.	51
3.1	Comparison of Hop-by-hop Routing and Explicit Routing.	63
4.1	Applications Traffic Types and Characterization.	89
4.2	Applications Bandwidth Requirements.	89
4.3	Traffic Sources for the IP versus MPLS Simulation Study.	93
4.4	Traffic Sources for MPLS Simulation Study.	103
4.5	Vanilla Mpls versus Diffserv MPLS Experiment Calculated Delays. . .	103
5.1	Mathematically Calculated Delays for the network with the Link. . .	118
5.2	Link Failure with Variable Loads Experiment Calculated Delays. . .	125

List of Figures

1.1	Routing in a simple data network.	2
1.2	IP Header Format.	11
1.3	Tunneling in an IP Network.	14
2.1	A Diffserv Router Basic Components.	42
2.2	2-bit Differentiated Services (modified Figure from [34]).	46
2.3	Tag Switching Packet Format.	48
2.4	Routing with Tag Switching.	49
2.5	MPLS Routing Domain.	50
3.1	Packet Format in MPLS Domain showing Shim Header.	59
3.2	Structure of MPLS Header.	60
3.3	A label Switching Path.	61
3.4	Upstream and Downstream LSRs.	63
3.5	Label Distribution.	64
3.6	Network Topology for LSP Setting Example.	68

3.7	MPLS Main Building Blocks.	70
3.8	Routing Function in MPLS Networks.	71
3.9	Label Swapping Process.	72
3.10	Time Sequence Diagram for MPLS Routing.	73
3.11	Routing in MPLS networks.	77
4.1	Test Model Configuration.	93
4.2	End-to-end Delay for Source 1 (VBR-500).	94
4.3	End-to-end Delay Frequency for Source 1 (VBR-500) in the MPLS Network.	95
4.4	End-to-end Delay Frequency for Source 1 (VBR-500) in the IP Network.	95
4.5	End-to-end Delay for Source 2 (VBR-400).	96
4.6	End-to-end Delay for Source 3 (VBR-200).	98
4.7	End-to-end Delay for Source 4 (CBR-1000).	99
4.8	End-to-end Delay for Source 5 (CBR-900).	100
4.9	Bandwidth Utilization in the IP Network.	101
4.10	Bandwidth Utilization in the MPLS Network.	101
4.11	MPLS-Diffserv versis Vanilla MPLS Test Model.	102
4.12	Vanilla MPLS: End-to-end Delay for the Network Traffic.	104
4.13	Diffserv MPLS: End-to-end Delay for the Network Traffic.	106
4.14	Diffserv MPLS: Queuing Delay for the Network Traffic at Router 2. .	107

4.15	Diffserv MPLS: Delay Variation for the Network Traffic.	109
4.16	Vanilla MPLS: Bandwidth Consumed by the Three Traffic Sources. .	110
4.17	Diffserv MPLS: Bandwidth Consumed by the Three Traffic Source. .	110
4.18	End-to-end Delay for the Network Traffic 10Mbps Core Links.	111
4.19	The Network Traffic end-to-end Delay 1.7 Mbps Core Links.	113
4.20	The Network Traffic end-to-end Delay 1.7 Mbps Core Links.	114
5.1	Simulated Network for Link Failure.	117
5.2	End-to-end Delay for Source 1 Traffic which is mapped to BE when Link Failure occurs.	119
5.3	End-to-end Delay for Source 2 Traffic with Source 1 Traffic mapped to BE when Link Failure occurs.	119
5.4	End-to-end Delay for Source 1 Traffic which is mapped to EF when Link Failure occurs.	121
5.5	End-to-end Delay for Source 2 Traffic with Source 1 Traffic mapped to EF when Link Failure occurs.	121
5.6	Link Failure with Variable Loads Simulation Network.	124
5.7	End-to-end Delay for the Network Traffic.	128
5.8	Queuing Delay for the Network Traffic.	128
5.9	Queue Length for Traffic of Source 2 (CBR-EF) at Routers 2 and 6. .	130
5.10	Queue Length for the BE Traffic at Routers 2.	132

5.11 Queue Length for the BE Traffic at Router 6.	132
5.12 End-to-end for the Network traffic (1.7 Mbps Links.)	134
5.13 Queuing Delay for the Network Traffic (1.7 Mbps Links.)	134
5.14 Traffic Sources Packet Loss at router 2 (1.7 Mbps Links.)	136
5.15 Queue Length for Traffic of Source 1 (CBR-BE) at Router 6.)	138
5.16 Queue Length for BE Traffic at Router 6.)	138
5.17 End-to-end Delay for the Network Traffic.	140
5.18 Queuing Delay for the Network Traffic.	140
5.19 Link Failure with Two alternative LSPs.	142
5.20 Link Failure with Multiple LSPs.	143

THESIS ABSTRACT

Name: YASSIR OBEID MOHAMMED
Title: Quality of Service Routing
Degree: MASTER OF SCIENCE
Major Field: COMPUTER ENGINEERING
Date of Degree: August 2001

In traditional routing, data flows are forwarded along the shortest routes selected on the basis of cost which might be hops count or delay, e.g., minimum-hops or traffic load, e.g., least-loaded. Networks nowadays are subjected to different types of traffic like audio, video, and images, besides the traditional data; these traffic types have variable quality of service requirements. Future networks are expected to handle even broader types of real-time applications, e.g., video-on-demand and tele-conferencing. Traditional routing is unable to provide services in such networks since it does not support characterization of traffic types as needed by the quality of service. This resulted in an increased need for a service aware routing that has the ability to handle different types of traffic and provide the applications with the quality of service required. In this regard, Multiprotocol label switching (MPLS) has been proposed to provide differentiated services in data networks. MPLS is expected to provide standard solutions for most of the issues related to quality of service routing. In this work we have experimentally evaluated MPLS and studied how it provides differentiated service. The performance criteria in our work are end-to-end delay, delay variation, and packet loss ratio.

MASTER OF SCIENCE DEGREE
KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS
Dhahran, Saudi Arabia
August 2001

ملخص الرسالة

الاسم: ياسر عبيد محمد
 عنوان الدراسة: امرار البيانات وفقاً لنوعية الخدمة
 التخصص: هندسة الحاسب الآلي
 تاريخ التخرج: اغسطس 2001

تقليدياً تقوم خوارزميات إمرار البيانات بالشبكات باختيار أقصر الطرق على أساس التكلفة والتي قد تكون عدد الموصلات المستخدمة أو زمن التأخير أو حال الزحام بالمرء، مثلاً باستخدام عدد الموصلات الأقل أو زمن التأخير الأقل أو الممر الأقل ازحاماً. شبكات البيانات في هذه الآونة عرضة لأنواع مختلفة من البيانات كالصور و الصوتيات هذه البيانات تتطلب نوعية خدمة خاصة ويتوقع أن تكون الشبكات في المستقبل عرضة لأنواع أكثر من شاكلة هذه البيانات كمثال المؤتمرات بالهاتف والفيديو عند الطلب. الأمرار أو التوصيل التقليدي للبيانات لا يستطيع خدمة هذه النواعيات من البيانات لأنه لايعتمد بتقييم البيانات وفقاً لنوعية الخدمة المطلوبة. هذا الأمر زاد من الحاجة للإمرار النوعي الذي يستطيع معالجة البيانات المختلفة وخدمتها وفقاً لنوعية الخدمة المطلوبة. في هذا الصدد تم إقتراح الإمرار المرقم متعدد الخوارزميات لتوفير خدمات متعددة بالشبكات و يتوقع ان يقدم حلولاً للعديد من المشكلات المتعلقة بالخدمة وفقاً للنوعية. في هذا البحث قمنا بتقييم الإمرار المرقم متعدد الخوارزميات عملياً و دراسة كيفية عمله لتوفير خدمات متعددة. معايير التقييم في عملنا كانت زمن التأخير الكلي لنقل البيانات والتغيير في زمن التأخير وكمية المعلومات المفقودة عند نقل البيانات.

درجة الماجستير في العلوم

جامعة الملك فهد للبترول و المعادن
 الظهران، المملكة العربية السعودية
 اغسطس 2001

Chapter 1

Introduction

The information exchanged in data networks is encoded in packets. Packets transfer between different source-destination pairs in the network need a sort of proper handling. Routing is the process of handling data packets. Routing is basically composed of two main functions [1]:

1. The construction of routing tables at each router in the network, and the selection of a route for a given source-destination pair.
2. The delivery of messages to their correct destinations via the selected routes.

The delivery of messages is straightforward using a variety of protocols and routing tables. Routing tables are data structures, which contain sufficient information to each reachable node with its next hop information.

Routers construct routing tables using routing algorithms. A routing algorithm selects paths to connect network nodes and its goal is to globally optimize the network performance, namely to maximize the network throughput. These algorithms generate routing tables based on the information that might be periodically exchanged among the nodes of a network and according to the methodology adopted in the routing algorithm. The methodology adopted could be to use the minimum number of hops or use the least loaded path.

Routing is one of the most complex and critical operations in data networks. The complexity is partially due to the following:

1. Routing requires coordination between all the nodes of the subnet rather than just a pair of modules, a source and a destination.
2. The routing algorithm must cope with link and/or node failures, that requires redirection of traffic and updating the databases maintained by the nodes.
3. The routing algorithm may need to modify the routes in the routing table in response to network traffic changes.

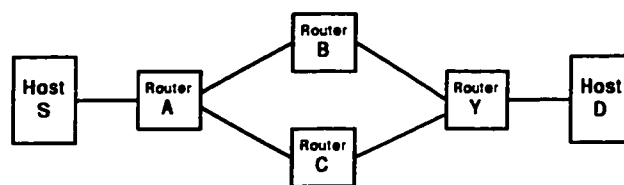


Figure 1.1: Routing in a simple data network.

To understand the routing process, consider the network shown in Figure 1.1.

Suppose that a data is to be transferred from the host S to the host D. When router A receives the data it has to decide whether to send it to router Y, on its way to D, through B or C. The path through router B will be the optimal if its link is shorter, faster, or wider in bandwidth compared to router C link. In this case, the routing protocol will give the decision to send packets through B. Now if router B fails or the link gets congested then router A, having received this information, runs its routing algorithm again which gives the decision to use the path through router C.

In the following sections, we provide details of some basic concepts related to routing in data networks.

1.1 Connection-Oriented and Connectionless Routing

Networks usually provide either connection-oriented or connectionless service. Routing in connection-oriented networks is performed utilizing the concept of virtual circuits. A virtual circuit (VC) is a cascaded sequence of links and nodes to connect a pair of end-nodes. These links may be shared partially or fully with other VCs.

In connection-oriented networks, data transfer is completed in three phases:

1. A source node (sender) establishes a connection by issuing a request.
2. The sender, after having established the connection, starts transmitting data.

3. At the end of transmission, the sender releases or tears down the connection.

In the connection establishment phase, a path is selected based on the requirements specified by the source router, resources are reserved at every router along the path to the destination. This path is uniquely identified in the router's routing tables. Connection-oriented networks need a sort of signaling protocol to be used in establishing the VCs states in switches and to close sessions.

Once a virtual circuit is established, packets are sent along this circuit with minimal routing information since they will follow the same path. In addition no sequencing information is needed to preserve the packet order since packets arrived in the destination in the same transmission order. This reduces the processing required to interpret and route each packet.

Reliable connection-oriented routing is of two types: message sequences and byte stream. There is a slight difference between the two types. In the first type the message boundaries are preserved, while in the other type data is treated as a stream of bytes.

In connectionless routing, no connection establishment phase is required and hence no signaling protocol is needed. Connectionless networks rely on little or no state being stored in the network. States indicate connections made and resources being reserved for them. In connectionless routing, each packet carries the full destination address, and each packet is routed independent of the others. Packets may take different routes since routing decisions are made differently for each packet.

As a result, packets may arrive at the destination out of order, which necessitates assigning sequence numbers to each packet to preserve the order. The connectionless model is also known as the datagram model.

Connectionless routing may result in better resource utilization when compared to connection-oriented routing. On the other hand connection-oriented routing guarantees better service than connectionless routing.

1.2 Distance-vector and Link-state Routing

Routing protocols can be categorized as distributed and centralized protocols [1]. Distributed routing implies that each node makes decisions regarding routing path selection independent of the other nodes in the network. In adaptive routing, routers employ a technique in which routes may change as a result of the dynamics in the network topology or traffic [2].

Distributed adaptive routing algorithms are commonly used in data networks and two classes of algorithms are broadly defined. In the first class, a node exchanges information with its neighboring nodes. Algorithms of this class are known as distance-vector algorithms. In the other class, a node exchanges information with all other nodes in the network. Algorithms of this class are known as link-state algorithms. The most popular distance-vector routing algorithm is the distributed Bellman-Ford algorithm [1].

The Bellman-Ford algorithm has been widely used in the Internet and it is known as the Routing Information Protocol (RIP). Using this protocol a node constructs its routing table with one entry per reachable network(s) address. One piece of information in the entry is the distance to the destination. The distance is expressed in metric form. The metric may represent the hops count to the destination or the routers queue length. It may also represent the delay experienced by a packet to reach its destination.

RIP algorithm has some limitations that can be summarized as follows:

1. It is limited to networks with small number of nodes.
2. It uses fixed metric (e.g., hop count), to compare alternative routes and does not account for the link bandwidth.
3. It takes long time to converge.

The link state algorithms have been developed to overcome the limitations of distance-vector algorithms. The most popular link-state protocol is the open shortest path first (OSPF) routing protocol [2].

Information exchange in OSPF is performed by *broadcasting* where each node broadcasts link state advertisement to every one of its neighbors. The link state advertisement contains information about all of the router interfaces. When an advertisement packet is received, it is retransmitted to every outgoing link except the one on which it has arrived. Broadcasting ensures a reliable transfer of informa-

tion and does not require any network topology information. Consequently each router obtains the entire network topology database and runs the shortest path first algorithm on its database to build its routing table.

OSPF supports different kinds of connections like, point-to-point and multi-access. OSPF routing is supposed to support a variety of distance metrics and to adapt automatically and quickly to changes in network topology. An important feature of OSPF routing is its support to quality of service [3]. For instance real-time traffic is routed differently from regular data. This is accomplished by having multiple routing tables. Each table is labeled with a cost according to the associated metric, e.g., one with the hop count metric, another with the queue metric, and a third with the delay metric. This complicates the computation and increases the hardware requirements but allows separate routing based on traffic type. When a connection is requested, the type of traffic is checked and the table labeled with the corresponding QoS metric is referenced accordingly to select the route to forward packets. It is worth mentioning that the IP protocol has a *Type of Service* (TOS) field, but it has not been used for routing decisions.

1.3 Interior Routing and Exterior Routing

Considering where the protocol can be deployed, there are two instances: Interior Routing Protocols (IRP), and Exterior Routing Protocols (ERP). Interior routing

protocols are used to exchange routing information within an *Autonomous System* (AS), while exterior routing protocols are used to exchange information between routers in different ASs.

An autonomous system is a system with the following characteristics:

1. It consists of a group of routers exchanging information using a common routing protocol.
2. These routers and networks are managed by a single organization or entity.
3. Each node is reachable from every other node.

RIP and OSPF for example are interior routing protocols. These protocols are not efficient for inter AS routing because different ASs may have different distance metrics, priorities, and restrictions. Therefore exterior routing protocols are used. These protocols simply provide information about networks that can be reached by a router and the ASs that must be traversed. Border Gateway Protocol (BGP) and Inter-domain Routing Protocol (IDMP) are examples of such protocols [1]. BGP uses path-vector routing technique [2]. The main features of this technique that makes it different from IRP are:

1. It does not include any distance or cost estimates, and each set of routing information sent by a router lists all the ASs visited to reach a destination network using this router.

2. The type of information provided enables a router to perform policy routing.

Policy routing allows a router to avoid certain paths to avoid transiting through a particular AS e.g., a router may have information about the performance or quality of an AS that encourages or discourages the usage of that AS.

3. Its design allows gateways in different ASs to exchange routing information.

BGP has three functional procedures: neighbor acquisition, neighbor reachability, and network reachability. Neighbor acquisition procedure is used to make two neighboring nodes in different ASs agree to exchange routing information. It is needed because one of the routers may not wish to participate in the operation. In this procedure, a router sends an **Open** message to another neighbor router, which may either accept or reject the offer. If the receiving router accepts the request, it retains a **Keep-alive** message as response.

Whenever the neighbor acquisition procedure is over, the neighbor reachability procedure is activated to maintain the agreement to exchange the reachability information. This procedure assures that routers still exist and engaged in the relationship which is confirmed by continuously exchanging **Keep-alive** messages.

Finally, each router maintains a routing table that contains the reachable networks and the preferred routes. Whenever a change is made to this table, the router issues an update message that is broadcasted to all BGP group members. This

process represents the final procedure of BGP, the network reachability procedure.

The Inter-Domain Routing Protocol (IDRP) is another exterior routing protocol. IDRP is also based on path-vector technique and it is advantageous over BGP [2]. The main differences between BGP and IDRP are:

1. IDRP operates over the Internet Protocol while BGP operates over TCP.
2. IDRP uses variable-length identifiers, while BGP uses a 16-bit AS numbers.
3. IDRP can deal with multiple internet protocols and address schemes.
4. BGP communicates a path by specifying the complete list of ASs visited, while IDRP aggregates this information.

Table 1.1 contains most of the important routing protocols. The four interior gate protocols are in use currently in the internet while the BGP is the only exterior gate protocol in use in the Internet.

Table 1.1: Main Types of Routing Protocols.

Protocol Type	Distance Vector	Link State
IGPs	RIP IGRP	OSPF Integrated IS-IS
EGPs	EGP BGP	IDPR

1.4 Forwarding IP Packets

Forwarding is the process of directing incoming packets of the ingress of a router to an outgoing port in their way to destinations. It is the second major process in data routing after the process of path selection. Forwarding is accomplished by examining the IP header of an incoming packet .

When a router receives a packet, it first checks the data-link layer header. If the link-layer header type indicates an IP packet, the router examines its header.

Figure 1.2 shows the IP header structure [3].

Vers	IHL	TOS
Length		
Identification		
Fragmentation		
TTL	Protocol	
Header Checksum		
Source IP Address		
Destination IP Address		

Figure 1.2: IP Header Format.

The router verifies the IP header contents by checking the Version, Internet Header Length (IHL), Length, and Header Checksum Fields. The version should be either 4 or 6 indicating IPv4 or IPv6, respectively.

The Header Length must be greater than or equal to the minimum IP header size (five 32-bit words). The Length, IP packet length, must be larger than the minimum

header size. Packet checksum is calculated and checked against the Checksum field. If any of these basic check parts fails, the packet is dropped.

The router then checks to assure that the TTL field value is greater than 1. TTL field is set to a value greater than or equal to the maximum number of hops, the packet is expected to traverse in order to reach its destination. Each hop decrements the TTL field by 1 when forwarding. When the TTL field value reaches 0 the packet is discarded and an ICMP TTL Exceeded message is sent to the sender. Using the TTL assures that a packet will not circulate forever.

Finally, the router checks the Destination IP address. The router uses this address as an index for the routing table lookup. The best-matching routing table is returned; it specifies the packet forwarding output port and the IP address of the next hop.

The forwarding process can further be modified [3]. For this purpose one or more of the following techniques could be considered:

1. *Multipath Routing*
2. *Type of Service (TOS) Routing*
3. *Extending IP options*

Multipath Routing: According to this technique, a router may have multiple paths to a destination. Any of these paths can be used as a route. This could lead to

balancing the load over the links. A router that employs this technique, will have a number of alternative paths when it consults its routing table for path selection.

Type of Service (TOS) Routing: The IP packet header TOS field may also be considered when selecting a path. Five TOS values have been defined for IP [3]:

1. Normal Service.
2. Minimize Monetary cost.
3. Maximize Reliability.
4. Maximize Throughput.
5. Minimize Delay.

Using TOS would enable a router to provide packets with better service and assign best paths. To implement TOS routing, a router keeps a separate routing table for each TOS. The router uses the packet TOS to specify the corresponding routing table then it runs the normal table lookup to choose the route. TOS routing has rarely been used and so far only OSPF and IS-IS protocols have been using TOS based routing. For the IPv6 the TOS bits have been removed.

Extending IP options: This technique implies extending the IP header by appending new fields to the header. These fields are used to assist or direct the path selection. The explicit route field is one of the options that might be considered in this regard.

Explicit route enables either *strict-source route* or *loose-source route*. The strict-source route is used to specify the exact path that should be followed by a packet. The loose-source route is used to assist in specifying the route path. It has been used for diagnosing the Internet routing problems and as a *tunneling* mechanism.

1.5 Tunneling

Tunneling is the process of configuring a path between two routers to be used in transferring data between them. In this case intermediate routers will only pass the data between the tunnel edge routers [3].

Consider Figure 1.3, assume that routers S and D know how to take packets from any source (e.g., Host N) to the destination Host R and the remaining routers in the network (X to Y) do not know how to perform that. In such case, a tunnel is configured between routers S and D.

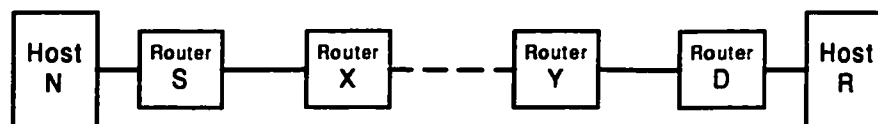


Figure 1.3: Tunneling in an IP Network.

The well known example of tunneling is the Internet *Multicast Backbone* (MBONE). The MBONE consists mostly of a collection of UNIX workstations running a common protocol that calculates path for multicast packets. The protocol used for this

purpose is the Distance Vector Multicast Routing Protocol (DVMRP) [4]. Since most of the routers in the Internet do not run DVMRP and hence do not know about multicast paths, the MBONE routers are interconnected by tunnels.

The situation could also be that the data application requires special treatment or handling that the intermediate routers do not support or do not support efficiently. Tunneling serves as a solution in this case.

To implement tunneling two mechanisms could be used: *Source Route Option* and *Packet Encapsulation*.

1. *Source Route Option*: Using this mechanism the router at the tunnel entry point uses the tunnel exit router address into the IP header as a destination address and moves the destination address to a loose route option. The router at the tunnel exit point recovers the original packet and forwards it to its destination.
2. *Packet Encapsulation*: In this mechanism the packet is encapsulated into an extra header addressed to the tunnel exit point router. This tells the tunnel exit point router to strip the IP header and forward the packet using conventional IP routing.

The encapsulation mechanism is preferred, due to the bad effect of using the source route option on the network performance [3]. Tunneling is generally avoided unless it is necessary because it has many drawbacks that can be summarized in the following:

1. The information adding and stripping at the two tunnel end points decreases the forwarding performance.
2. Tunneling makes traffic monitoring more difficult.
3. Tunneling may badly affect security issues since it can subvert firewalls.

1.6 Quality of Service Routing

Quality-of-Service (QoS) refers to the concepts and mechanisms that enable a network to give performance guarantees to traffic passing through it. Performance guarantees can be stated as guarantees on minimum throughput, maximum delay, maximum loss rate, or maximum delay variations (delay jitter).

QoS may be defined at the user-level or the network level. The main network QoS parameters include throughput, reliability, delay, and delay variation (jitter). The user QoS requirements specify the bandwidth that the user application needs, and the packet loss ratio and delay it tolerate.

The primary goal of QoS routing is to find for each traffic source a path that has sufficient resources to meet the QoS requirements of the source and to select a low cost path whenever possible.

1.7 Problem Statement

Traditional routing algorithms primarily make routing decisions based on available information and strive to satisfy some form of least cost criterion; the criteria could be to minimize the number of hops used, the monetary cost of using hops, the delay associated with the trip between a source and a destination, or some combination of these. The cost may be inversely proportional to the link capacity and proportional to the link traffic load. Routing protocols (e.g., RIP and OSPF) make use of a single metric such as hop-count or delay and they often use shortest path algorithms for path computation.

Traditional routing algorithms and routing protocols have shown efficient performance with traditional data networks where most applications are file transfer or electronic mail. However they fail to perform efficiently in current networks where there is different types of traffic with variable QoS requirements. This failure of providing QoS routing is mainly due to the following:

1. Routing algorithms are mainly concerned with connectivity and reachability.
2. Routing algorithms provide a best effort delivery service for all applications. In other words, they do not support service type routing and adopt same routing policy whatever the type of traffic is or whatever its QoS requirements.
3. Routing algorithms only consider optimal cost routes although acceptable paths may exist.

The routing problem can be defined as follows: Given a network $G = (V, E)$ where V is the set of nodes, E the set of edges or links, and given a set of sources S , a set of destinations D , a set of QoS constraints C , as well as a traffic profile, ϕ , describing the traffic behavior of each source e.g, a Constant Bit Rate (CBR) or a Variable Bit Rate (VBR) and related parameters, find the best existing path or route from S to D which satisfies C . A good routing algorithm to produce such route should seek to satisfy some or all of the following:

1. Improves the network performance by making the best utilization of the available resources.
2. Protects delay-sensitive type of traffic.
3. Takes the minimum possible time for route selection.
4. Has a reasonable complexity as well as a minimum possible computation overhead to have a feasible hardware implementation.

Current network applications such as audio, video, VoIP etc. have QoS concerns. Traditional routing does not have the ability to satisfy these concerns. This has lead research towards developing a new generation of routing algorithms and routing techniques that consider the type of application and its QoS requirements.

1.8 Proposed Work

Major efforts are being made to transform the Internet into an application-aware network. Problems addressed in this regard include:

1. Traffic differentiation: A means to differentiate between applications is needed to guarantee satisfying different applications requirements.
2. Traffic Engineering: Certain capabilities such as flow control and fast recovery in case of link or node failure should be provided.
3. Faster switching/routing: To reduce processing time at edge and core routers which results in reduced end-to-end delay.
4. Packet scheduling: New methods for scheduling based on traffic classes are needed.

The major technology solutions provided are: Traffic Prioritization [5], Integrated Services (Intserv) [6] with RSVP, Differentiated Services (Diffserv) [7], and Multi-protocol Label Switching (MPLS) [8, 9]. Mechanisms and methods of operation of these technology solutions are provided and detailed later.

MPLS is taking the lead among the proposed schemes to address the QoS related concerns. In this Thesis, we propose to work on the experimental evaluation of MPLS in supporting traffic flows with QoS requirements. Issues to be studied include:

1. To find out how an MPLS network behaves towards different types of applications under different network states.
2. To examine the MPLS reaction to link failure.
3. To evaluate the MPLS enabled network fairness and resources allocation.

1.9 Thesis Organization

The rest of this Thesis is organized as follows: Chapter 2 reviews the literature published addressing the issue of routing in general and QoS routing as well. Chapter 2 also introduces some basic concepts related to our work and covers the main techniques that supports QoS routing. Chapter 3 details the operation of MPLS and provides full coverage of related issues.

In Chapter 4 a review of the data traffic behavior and characterization is given and performance metrics are detailed. In this Chapter we have defined our test platform, tools used, and related details. In the rest of this Chapter and the following Chapter, Chapter 5, each simulation set is described and its objectives are stated. Finally results are provided, analyzed, and discussed in detail.

In the last Chapter, we conclude by a summary, conclusion, and suggestions for the future work.

Chapter 2

Literature Review

Many schemes have been proposed in the context of designing a powerful, efficient, and less complex routing algorithm for dynamic networks [10]. In some cases, such as in [11] and [12], the algorithm effectiveness depends on the traffic load conditions. Some algorithms, like in [13] and [14], are computational intensive and that may result into inefficient performance and hence affect connection acceptance rate.

Classical routing algorithms select the routing path with the least cost which may be presented as a function of the number of hops used, congestion state, delay, etc. For example hop-based routing algorithms include the minimum hops only (MHO) count or minimum hops (MH) count. The two algorithms are basically similar but they differ in that, the MHO strictly checks for the availability of the route with the least number of hops. If path is not available or the available bandwidth cannot handle the connection requested bandwidth, the request is rejected. However, when

MH is implemented, an alternate path is sought even at a higher cost. Studies such as [11] and [13] have shown that the minimum hops algorithm (named as alternate path algorithm in some references) is more efficient at high traffic load than at light traffic load. Routing algorithms based on congestion state such as the least-loaded (LL) algorithm are also implemented. The least loaded algorithm is more efficient at low traffic load than high traffic load. In short, this means both types of algorithms are not efficient under all states. Delay-based algorithms strive to guarantee the minimum delay but cannot avoid congestion.

Some of the proposed routing algorithms [15], [16], [17], and [18] work to support QoS requirements of applications. The primary goal of QoS routing is to find for each traffic source a path that has sufficient resources to meet the QoS requirements of the source traffic and to select a low cost path whenever possible.

In this chapter we will review some of the traditional routing algorithms as well as those proposed to support QoS routing. Some of the techniques developed to provide differentiation in services are also reviewed.

2.1 Unicast Routing Algorithms

T. E. Tedijanto and Onvural [11] considered the path total delay as the cost metric. They attempted to satisfy the end-to-end delay constraint; this means that their algorithm finds the path whose hops total delay causes the minimum possible delay

to the routed packets. The major drawback of this algorithm is that its optimizing object is based on a local load balancing function instead of the global optimization of the network performance. In their work they mentioned that it has a complexity of polynomial order but no analysis is presented.

N. F. Huang *et. al* [19] proposed a frame work that combines both concepts of minimum-hop and least-loaded. The decision on adopting which of the two methods is based on the available bandwidth threshold value. In their work, a threshold value equals to half of the link capacity is specified and used to find out whether the routing decision is to be based on satisfying the hop count using the minimum-hop algorithm, or to be based on load balancing using the least-loaded algorithm.

Antonios F. Atlasis *et. al* in [13] have proposed an algorithm based on a cost function that combines the use of load-balancing concepts and the MH route and employs the trunk reservation algorithm. The trunk reservation algorithm is a simple variation of the minimum-hop algorithm. It aims to enhance the performance of the routing algorithm in heavy traffic load conditions. The basic idea in the trunk reservation algorithm is to fix a value as a trunk utilization threshold ρ_{trunk} . When the expected route utilization ρ_{route} is higher than the trunk utilization, the call is accepted and assigned this route only if this is a minimum hop one. Otherwise, another route is tried. The route utilization ρ_{route} is defined as the maximum expected utilization of the links that constitute this route [13]. In this algorithm, routing tables are constructed at the source routers using any standard shortest-path algo-

rithm, while the trunk reservation technique is applied with a probabilistic linear equation [13]. This equation adopts the use of the trunk reservation at high traffic load where it performs very well and increases the throughput and hence performance, at the same time it is not used at light traffic load where it performs badly. Their simulation studies have shown that their algorithm performs better than the other algorithms whether these algorithms are based on minimum-hops, least-loaded or trunk reservation principles. Their algorithm is centralized and it is not tested under variable traffic load. The traffic source used for the simulation provides one type of traffic that is voice. The network they use is of small size with limited link capacity of 10.6 Mbps.

J. M. Jaffe [15] proposed an algorithm known as multi-label algorithm that finds the optimal path while satisfying multiple constraints. A router using this algorithm considers all links to the next node in a path in what is called a temporary label set. The link that satisfies the constraints and has the minimum cost is selected and added to the permanent label list. The process continues till the sink node is reached. If a node(s) in the path can not satisfy the constraints it is discarded and adjacent nodes are considered. The multi-label algorithm complexity is exponential because the comparison is between vectors when comparing elements in the temporary labels set since constraints and cost are all compared [12]. If no constraint is considered the algorithm tends to be the classical Dijkstra algorithm and its complexity is $O(n^2)$.

Jin Liu *et. al* [12] considered the exponential complexity which increases with

network scale as the major drawback in the multi-label algorithm. The authors proposed, as a solution, to limit the total number of label at each node to a certain value L_m . The authors presented analysis that show limiting the number of labels makes the algorithm complexity polynomial [12]. The problem with their work is that in the search process some labels in the set of temporary labels are discarded. If the optimal shortest path falls in the discarded labels the algorithm will not be able to find the optimal cost. They proposed to solve this problem by allowing the variance between output and optimal value of cost and constrain to be within a range of *DeltaCost* and *DeltaConstraint*, respectively. According to their statements this results in considering two paths that start and end at two same nodes, and their cost and constraint are close enough to each other, the same. In this case L_m is defined as follows:

$$L_m = (1/\textit{DeltaCost}) \times (1/\textit{DeltaConstraint})^c$$

where c is the number of constrains. For further reduction in L_m , they stated that the discard rule of temporary label is to minimize the potential increase on the path cost [12]. The authors in this work stated that their algorithm complexity is also polynomial and presented analysis that proves this. However the mechanism for discarding labels is not clear and does not guarantee avoiding to discard the optimal path label.

Antonios F. Atlasis *et. al* [20] have proposed an adaptive routing algorithm based on a learning automaton. A learning automaton is a finite state machine that in-

interacts with a stochastic environment, trying to learn the optimal action the environment offers, through a learning process [20]. The authors used a Stochastic Estimator Learning Algorithm (SELA). In operation, a standard algorithm is used to select routes. SELA is to be run in case of congestion and calls are blocked due to that. Their simulation study examines SELA performance in an ATM network in terms of the blocked calls and the successfully transmitted cells. Their results compared with the results of standard LL and trunk reservation algorithms have shown that SELA minimizes the number of blocked calls and maximizes the successfully delivered cells. SELA can be easily implemented according to the procedure described and it does not require complicated computation and other overhead. SELA can be used only in ATM networks which limits its application.

K. R. Krishnan and R. M. Cardwell [21] realized what there is a basic similarity between the virtual channel/virtual path concepts and the circuit-switched concepts where a certain link between two nodes in a network is reserved for some time. In this regard they made use of a single-link and two-link connection technique employed in circuit-switched networks in which the number of links used for routing is limited to one or two. They applied this technique to an ATM network and examined the effect of this restriction on the performance. They compared their results with the results collected by applying an algorithm that may use many routes as in virtual path. The results have shown that, limiting the routes to one or two links will not affect the performance significantly at high traffic load but, at low traffic load, it

has a clear negative effect on the performance. It has also been pointed out that network performance degrades as its size increases.

Turner and Tamir [14] modified the virtual channel connection basic idea in what they call a dynamic virtual circuit (DVC). In virtual channel once a connection is established, the assigned path is reserved for the connection life time. In their modification a virtual channel can be terminated during transmission to free the path for some other connections and the remaining packets are to be transmitted later or through a better alternative path. For this purpose, they introduced what they call Circuit Establishment Packet (CEP) and Circuit Destruction Packet (CDP). A CEP is inserted by the source at the initiation of the DVC, while the CDP is inserted to indicate the end of the connection. It is possible for a router to terminate a connection by inserting a pseudo CDP. To forward the remaining bits of the connection, the router inserts a new CEP. In order to keep the sequence of packets belonging to a particular DVC an Input Mapping Table (IMT) is introduced. IMT is used to keep information about the current connections passing through a router and it augments the packet header with a sequence number. Packet reordering is conducted at the router connected to the destination node. This technique may be efficient in improving the performance for some networks such as small full meshed networks but, it does not address the issue of QoS routing.

S. C. K. Nahrstedt [16] proposed a heuristic algorithm for multi-constraint routing. The idea behind the heuristic is to break the problem into two. The problem

is solved first considering one constraint then, the solution is repeated considering the other constraint. The final solution satisfies both constraints. Nahrstedt applied his algorithm on a network considering end-to-end delay and path cost as the two constraints. The problem is first solved for cost then, repeated for delay. The simulation he conducted results are compared to results obtained using an optimal algorithms. Figures presented for the algorithm performance under different scenarios, have shown that it was able to find the optimal path in most of the cases [16]. Again this does not constitute service based routing. An obvious drawback of the algorithm is the considerable amount of setting time and the complex hardware design.

D. Subramanian *et. al* [22] proposed an algorithm that was based on observing the behavior of ants in their search for a route between their house and a food source. In this algorithm every node periodically injects a short packet, known as an ant, to explore the network. An ant has the form (h_d, h_s, c) ; where h_d indicates the destination address (d), h_s indicates the source address (s) and c indicates the cost of the path from d to s . Initially the cost is zero and whenever a node receives this ant it states its address and adds the cost of the path from the source to itself. The ant is destroyed at some node when the cost exceeds a predefined value. According to the authors ants can be piggybacked with any transmitted data which reduces the amount of information exchanged between nodes. This algorithm is expected to save a considerable bandwidth that has been consumed in exchanging

control information. This improves the performance and increases the rate of call acceptance. However it has been tried on a limited number of networks and with a single type of data (voice). This algorithm does not provide QoS routing since only cost and not any other constrain is considered.

2.2 Multicast Routing Algorithms

In unicast routing data is routed from a source node to a single destination node; it is also known as a point-to-point routing. On the other hand broadcasting is where a message is received by all active destinations in a network. Multicast routing is a selective broadcast in which a message is sent to some members in the network. A critical design issue in multicasting is to minimize the number of copies of a message sent to the recipient. The multicast routing problem is to find the best feasible tree to cover a source node, s , to a set of destinations while satisfying certain constraints (c).

An N -unicast is a process that achieves multicast transmission by using N separate unicast transmissions, one to each recipient. An algorithm based on this technique is known as an N -unicast routing algorithm. N -unicast is not a multicast transmission, but it can be used as a reference for comparison. If a multicast routing algorithm performance results in worse than N -unicast, the proposed algorithm is not a practical multicast routing algorithm.

The Conference Call (CC) routing algorithm [23] is an example of multicast

routing algorithms. CC routing algorithms is used in the telephone system for voice conference calls. It uses a specialized conference hub, statically located in the network, as a synchronizing point for the sender. A sender sends one copy of a message to the central hub which replicates the message and delivers a copy to each member in the conference or the multicast group. The call blocking probability in the conference call routing algorithm increases with the mesh size. For small meshes with small multicast groups, conference call routing algorithm has a blocking probability rate smaller than that of unicast algorithms. On the other hand for small meshes with large multicast groups, conference call routing algorithm has a blocking probability rate higher than that of unicast algorithms.

The core-based trees (CBT) [24] is another example of multicast routing algorithms. As the situation in CC algorithm, CBT uses a hub to handle messages delivery to the multicast group members. Instead of using a static hub, it locates the hub dynamically for each multicast call depending on where sender and recipients are. Again the sender sends one copy of a message to the Hub which replicates the message and delivers a copy to each member the multicast group. The call blocking probability of the CBT increases with the increase in the mesh size.

2.3 QoS Routing Algorithms

QoS routing is the first step to guarantee QoS requirements of different applications. QoS routing tends to identify paths that meet QoS constraints, and selects the one that improves the network overall performance while making better resources utilization.

Chen *et. al* [17] proposed a multipath distributed heuristic for QoS routing. The basic idea of this algorithm is to send routing messages called ‘probes’ from a source (s) to some destination (d) to search for a path that satisfies the end-to-end delay constraint. Probes are guided along the best path using the state information of intermediate nodes. The number of probes generated depends on the contention level of the network and the degree of the QoS demands. According to the authors statements this algorithm search strategy achieves good tradeoff between routing overhead and routing performance, and considers QoS as well as path optimization. The algorithm does not consider other QoS parameters such as packet loss ratio and delay variation.

F. Xiang *et. al* [18] proposed to use genetic algorithm to guarantee QoS routing in multimedia applications environment. They have applied genetic algorithm to solve QoS unicast and multicast routing problems. In this work they established a fitness function that satisfies the following conditions:

1. The total cost of the selected route should be minimal.

2. There is only one route from source to destination.
3. Prevent non-existing links in the network to be selected.
4. Satisfy QoS constraints.

They considered one-point crossover and bit mutation. The elitist model is adopted as the selection operator, which is done by executing the selection operation according to the Monte Carlo method, then copying the chromosome with highest fitness to the next generation. Their simulation results have indicated that, the optimal solution has been achieved.

2.4 Mechanisms for Quality of Service Routing

The major technology solutions developed in response to the challenge of QoS routing are:

1. Traffic prioritization (Class of Service (CoS)).
2. Integrated services (Intserv).
3. Differentiated services (Diffserv).
4. Multiprotocol Label Switching (MPLS).

The following sections review these mechanisms which are developed to provide QoS routing. The main techniques used in these mechanisms are also covered.

2.5 Resource Reservation Protocol

Resource reSerVation Protocol (RSVP) is a mechanism that is designed to operate with routing protocols. RSVP was originally intended to assist QoS routing in networks with integrated services [25]. In operation, RSVP is used by a host on behalf of an application to request a specific QoS from the network. It is also used by routers to establish and maintain a state to provide the requested service. An RSVP request in general results in resources being reserved along the data path.

RSVP has four types of messages which are defined as follows:

1. **Path Message:** A Path message is initiated by the sender to set a path and forwarded to the next hop.
2. **Resv Message:** This message is a reply to the **Path Message** indicating that the requested path has been reserved. **Resv Message** is initiated from the receiver.
3. **Error Message:** When a path cannot be established an **Error Message** is initiated from the receiver and forwarded to the sender indicating the reason for path declination.
4. **Tear-down Message:** This message is initiated either from the sender or the receiver when a session is to be terminated. It results in releasing the resources along the path.

RSVP uses **Path** and **Resv** messages in setting a path. The **Path Message** is originated by a source node to install a *path-state* in every node along the route. The destination node, having enough resources, responds with a **Resv** message containing traffic descriptors and QoS parameters traversing the path to reserve resources at each node along the path.

The most important object of **Path** and **Resv** messages, considering QoS routing issue, is the *FLOWSPEC*. This object defines the flow and specifies its QoS requirements. If *FLOWSPEC*, also known as class of service CoS, is set to zero best effort mechanism will be used. *FLOWSPEC* object has three fields:

1. *Service number*: Specifies the type of service requested.
2. *RSPEC*: Describes the flow QoS special requirements. The description may be in a qualitative form (e.g., specifying controlled load service). It may also be in a quantitative form (e.g., asking for a maximum delay of 100 milliseconds, for audio/video real time applications).
3. *TSPEC*: Describes the flow characteristics. It specifies the bandwidth needed for the flow. The bandwidth required may not be specified as a single number as the case when the *token bucket* filter is used to define and specify the bandwidth required. The *token bucket* filter is described by two parameters: a token rate (r) and a bucket depth (B). In short, B specifies the longest burst that can be accommodated by a node/router and r specifies the rate at which

packets are sent. The *token bucket* filter allows the packets sending rate to change i.e., the sender can send in a rate faster than r but for a sufficiently long period of time the sender should not exceed r .

In connection-oriented networks resource reservation is carried by the sender according to the QoS requirements of a flow as the case in ATM networks. RSVP adopts receiver-oriented resource reservation approach in which the **Path** message figures the path and the receiver conduct the reservation [25]. This approach facilitates multicasting groups because it permits receivers to decide about the specifications of the application to receive. This is useful in multicasting groups that have receivers more than senders and receivers have different requirements.

The second main feature of RSVP beside the receiver oriented resource reservation, is the use of a *soft state* approach to manage the reservation state in routers and hosts. The *soft state* is created and periodically refreshed by **Path** and **Resv** messages [25]. The *soft state* principle enables the receiver to change the reservations it has made using the **Resv** refresh message to demand for a new reservation. RSVP relies on the soft state mechanism for tearing down since a state is automatically torn down if it is not refreshed within the time period fixed for refreshing.

2.6 Traffic Prioritization (Class of Service (CoS))

Traffic prioritization technology, also referred to as Class of Service CoS, is a product from 3Com's company. Its vendors believe it is an effective, yet simple, tool for providing differentiated services [5]. Traffic prioritization has two approaches to prioritize the traffic:

1. The switches and routers implement a packet classifier that identifies incoming traffic and prioritizes it.
2. Desktops mark their packets with a priority value, which is used by switches and routers to queue outbound traffic.

Once a packet is marked with a certain priority, traffic is tagged for this traffic priority. To simplify administration and prevent users from setting their own priorities, the network provides a centralized control where traffic priorities can be defined for the entire network. The network administrator can prioritize traffic based on the type of application, the protocol used, the user, and any other conditions.

To implement traffic prioritization all or at least some of the following components should be available in the network.

1. Desktops and servers: They must be able to:
 - (a) Receive and store centrally set priority policies.

- (b) Recognize applications (e.g., based on TCP and UDP port information) in order to prioritize them.

- (c) Label packets with priority markings.

3Com's *Dynamic Access software* (version 1.5), supports all these functions, with no impact on desktop or server performance [5].

2. Policy server and policy manager: The policy server will automatically distribute policies to desktops, servers, switches, routers, and access devices in order to provide a centralized point of control so that policy can be set across the entire network. Initially, standard management tools will set priority policies on a per-device basis.
3. Core layer-3 switches and routers: To prioritize the traffic effectively, these devices need to be able to:
 - (a) Read the priority marking of each incoming packet (e.g., read the priority bits set by the desktop or the prioritization server).
 - (b) Classify unprioritized packets, at wire speed, via a packet classifier that examines packet's header fields then, sets the priority bit(s) accordingly.
 - (c) Classify packets to multiple priority queues for each output port.
4. Edge layer-2 switches: To provide the ability to recognize priority tags and

build multiple queues. They are not needed if layer-3 switches are well provisioned.

5. Packet Schedulers that decide which packets get dropped during severe congestion periods.

Traffic prioritization defines eight traffic classifications based on IEEE 802.1p Priority Values and Associated Traffic Types. The 802.1p-enabled switches will use the 802.1p priority bits to prioritize traffic. The 802.1p standard has proposed specific associations of priority values with real traffic types. The proposed priority values and corresponding traffic types are presented in Table 2.1 [5].

Table 2.1: Priority Values and Traffic Types.

Binary	Traffic Types
111	Reserved
110	Interactive Voice
101	Interactive Multimedia
100	Controlled Load Applications (or Streaming Multimedia)
011	Excellent Effort (or Business-Critical)
010	Standard
001	Background
000	Best Effort (the Default)

These are proposed guidelines and network managers can define these levels differently to meet their specific needs.

2.7 Integrated Services (Intserv)

Intserv is an IETF service classification standard that relies on RSVP to setup and tear down reserved resources along the intended path for packets flow [6]. To support Intserv in a network, two fundamental mechanisms must be present:

1. A mechanism to control the QoS delivered to an application: This is provided by one of the QoS control services such as Controlled-Load [26] and Guaranteed Service [27]. Details of these two types of services will be provided later.
2. A mechanism to communicate the application requirements to the network resources: This is frequently implemented by a resource reservation setup protocol such as RSVP [25].

The following components are necessary to implement the mechanism needed to control the QoS in Intserv networks:

1. *Admission Control*: That enables the network to refuse customers when demand exceeds capacity.
2. *Packet Schedulers*: That enables the network to implement priority scheduling. *Fair Queuing* [28] is an example of the implementation of a packet scheduler.
3. *Flow Classifier*: Its primary function is to distinguish among different flows. Currently a flow classifier classifies flows based on the source and destination addresses and port numbers.

Intserv networks support two types of services in order to provide QoS:

1. Controlled load: For delay and delay jitter tolerant applications.
2. Guaranteed service: For delay and packet loss intolerant applications. For example, audio and video *play-back* applications. Guaranteed service guarantees that packets arrive within the guaranteed delivery time and do not get dropped due to queue overflow, provided that the traffic stays within its specified parameters.

Intserv is a ready to use technology and seems to be an advanced step towards providing QoS routing but has some drawbacks:

1. Lack of a mechanism for admission policy and admission policy control.
2. Routers periodically exchange **Path** and **Resv** refresh messages to preserve the reservation soft state which results in high control overhead flowing between routers. The increased control overhead degrades performance. It also requires considerable memory storage in routers which results in high complexity and uncertain maintaining costs.
3. Lack of scalability. RSVP exhibits many scaling and implementation challenges because it requires stateful routing and forwarding decisions.

2.8 Differentiated Services (Diffserv)

Instead of maintaining individual flows on all routers as in Intserv, flows in Diffserv are aggregated into a flow that receives specific treatment. The Diffserv mechanism for differentiation between flows employs a small and a well defined set of components from which a variety of treatments or *aggregate behaviors* may be built. A set of bits in a packet IP header is set to mark the packet to receive a particular forwarding treatment at each router. This specified forwarding behavior is known as a *Per Hop Behavior* (PHB) in DiffServ terms [29]. The IETF Diffserv Working Group in [30] has defined and standardized an architecture as well as specific PHB for Diffserv networks. In [7] the group has standardized a method for the general use of TOS.

To implement Diffserv a network must perform essential functions that are outlined in the following:

1. *Admission Control*: That enables the network to refuse customers when demand exceeds capacity.
2. *Packet Scheduling*: That should treat different customer's applications differently as needed.
3. *Traffic Classification*: That enables the network to reorganize data streams into substreams that receive different treatment.
4. *Policies*: That helps allocating the network resources.

A router in a Diffserv network performs three functions: it defines packet treatment classes, allocates the suitable resources for each class, and sorts all incoming packets into their equivalent classes. The router has two main functional units, a classifier which classifies packets dependent on the PHB and a conditioner as shown in Figure 2.1.

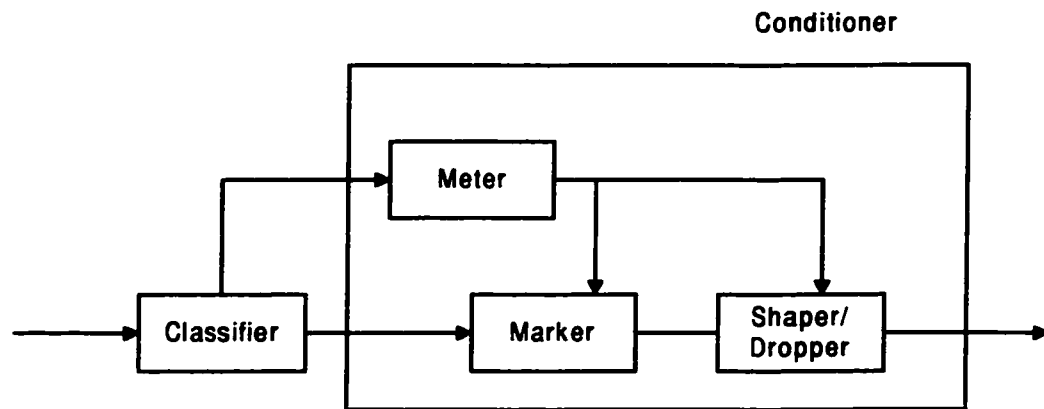


Figure 2.1: A Diffserv Router Basic Components.

The conditioner is composed of the following parts:

1. Profile meter: Measures the current state of a stream of packets.
2. Marker: Sets Diffserv field to a Diffserv behavior aggregate.
3. Shaper/dropper: Delay or discard some of the packets in order to shape a stream to comply with its traffic profile.

In Diffserv, the IP header TOS field, which was originally used to indicate a packet queuing precedence, is redefined to support differentiated services. The TOS

field is then renamed as Differentiated Service (DS) field. Diffserv operation is based on setting the renamed bits at the entry edges of the network. In this operation six bits of the DS field are used to convey the Diffserv Code Point (DSCP). Intermediate nodes use the DS field only to classify packets and use DSCP to determine how to forward packets.

Setting of the DS field and conditioning of marked packets behavior is performed in the conditioner and may vary in complexity according to the specifications of the PHB scheme used. Conditioning of marked packet behavior includes metering data streams and shaping them to confirm that they remain within agreed upon levels and drop the extra packet in case that A flow exceeds its limits. The Diffserv buffer management and packet scheduling mechanism are capable of delivering the specific packet forwarding treatment indicated by the DS field value.

The Diffserv working group proposes two types of premium services: *Expedite Forwarding* (EF) and *Assured Forwarding* (AF). The ability to indicate traffic priority and peak bandwidth is what differentiates premium service from best-effort (BE) service [31].

Expedite forwarding EF service is intended for jitter-sensitive traffic such as voice and video. EF traffic should experience less delay variation than either BE traffic or AF traffic which means EF traffic is forwarded in routers before other types of traffic (up to a certain rate). The user is assumed to not exceed the specified peak rate, while the network is supposed to provide the bandwidth guaranteed at anytime.

Assured forwarding AF service relies on buffer management within routers to provide a better-than-best-effort type of service. With the AF service, a customer gets some assurance his AF traffic is less likely to be dropped in the event of network congestion, as long as it stays within the agreed upon traffic profile [31]. If the number of assured packets in the queue exceed the agreed upon threshold, excess packets are dropped. This limits the amount of assured traffic, so that the best effort traffic is not completely suppressed. The assured service packets are not dropped even when the queue is full, if they do not exceeded the threshold.

Clark and Wroclawski [32] proposed a mechanism for providing assured forwarding service. Their principal idea is to define a service profile for each user, and to design a mechanism in the router that favors traffic that is within those service profiles. This mechanism can simply be described as: monitor the traffic of each user at its entering router and label its packets as being either *in* or *out* of its service profile. Then at each router, if congestion occurs, preferentially drop traffic that is labelled as being *out*.

Jacobson [33] proposed to provide expedite forwarding EF service using a single bit of the DS field. The DS bit, called a *Premium* service bit, is used to serve as traffic type indicator. Jacobson defined a premium (expedite) service that is provisioned according to peak capacity profiles that are strictly not oversubscribed and that is given its own high-priority queue in routers. Premium service levels are specified as a desired peak bit-rate for a specific connection. In this approach the edge router

examines packets entering the network to set the premium bit of those that match a premium service specification. Premium service scheduling mechanism enqueues premium service packets in a separate queue which means routers along a path will have two queues, one for premium service traffic and the other for best-effort traffic. Two simple actions only are needed in the forwarding path, to classify a packet into one of the two queues on a single bit, and to service the two queues using simple priority. Packets marked for premium service are sent first with high priority while unmarked packets are sent at the lower priority.

K. Nichols *et. al* [34] combined the concept of Assured service of reference [32] and the concept of Premium (expedite) service of reference [33] to benefit from the advantages of both in providing a more efficient differentiated services. They denoted each pattern by a bit, one they call the premium or P-bit, and the other is the assurance or A-bit. In this section, we will refer to the authors proposed architecture by the two-bit method.

According to the two-bit method, edge routers are configured with a traffic profile for a particular flow based on its packet header. An arriving packet A and P bits are cleared then, the packet is classified on its header. If the header does not match any configured values, the packet is immediately forwarded which means it will receive best effort service since its A and P bits are cleared. Matched packets pass through individual markers that have been configured from the usage profile of that flow: service class (EF or AF), rate (peak for EF, expected for AF), and permissible

burst size (optional for EF). AF traffic packets depart from the AF marker with their A-bits set. EF traffic packets depart from the EF marker with their P-bits set. Markers then pass packets to the forwarding engines.

According to the two-bit method, each router output port must have two queues one for EF traffic and the other for AF and BE traffic. To place an incoming packet in the appropriate output queue, each router must implement a test on the P-bit of the packet. EF traffic queue receives higher priority service and its traffic is never dropped unless it exceeds the peak rate. The forwarding decisions for the two-bit method are displayed in Figure 2.2.

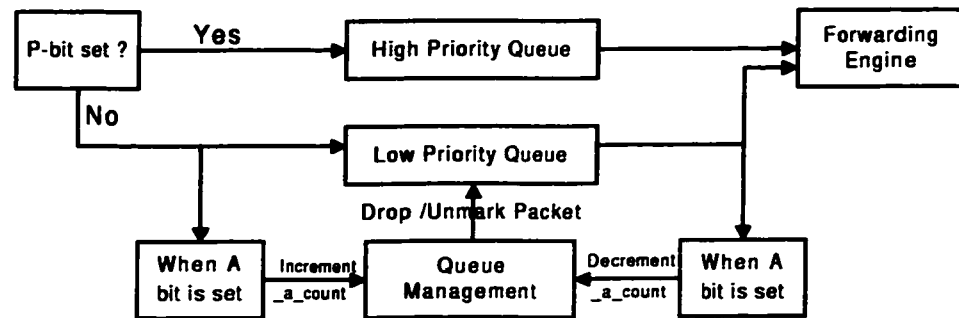


Figure 2.2: 2-bit Differentiated Services (modified Figure from [34]).

During congestion, packet drop takes place in the AF and BE traffic queue. In the two-bit method, routers output queues implement scheme called RIO as a packet drop mechanism. RIO is an extension to the RED algorithm. RIO stands for RED In and Out while RED stands for Random Early Detection. A RIO scheme uses two thresholds for when to begin dropping packets; a lower one is based on total

queue occupancy for ordinary best effort traffic and one is based on the number of packets enqueued that have their A-bit set. This means that any action preferential to Assured service traffic will only be taken when the queue capacity exceeds the threshold value for ordinary best effort service. In this case, only unmarked packets will be dropped. Keeping an accurate count of the number of A-bit packets currently in a queue requires either testing the A-bit at both entry and exit of the queue or some additional state in the router.

In two-bit method, the authors described the basic functional units required to implement their scheme. In the architecture they have modified the marker and the profile meter to include: a general classifier, a *Bit-pattern classifier*, a *Bit-setter*, policing token bucket, and shaping token bucket, besides the priority queues. These primitives constitute a Marker (either a policing or a shaping token bucket plus a bit setter) and a Profile Meter (a policing token bucket plus a dropper or bit setter).

2.9 Multiprotocol Label Switching (MPLS)

MPLS represents the latest technology developed to satisfy the need for QoS routing [35]. MPLS differentiated services are based on a *per hop model* where aggregate forwarding resources (buffer space, bandwidth, and scheduling policy) are allocated in each router for each Diffserv forwarding class. MPLS uses **Label Switched Path (LSP)** concept which is similar to the concept of VP/VC in ATM. An LSP

SETUP message needs to indicate which Diffserv forwarding class or classes an LSP belongs to. This information can be used as a constraint in the LSP route selection and in verifying that packets sent along an LSP belong to the correct forwarding classes. MPLS is produced as an extension to tag switching technology [36]. In tag switching a tag can either be explicit or implicit in layer-2 header. Figure 2.3 shows a tag switching packet format.

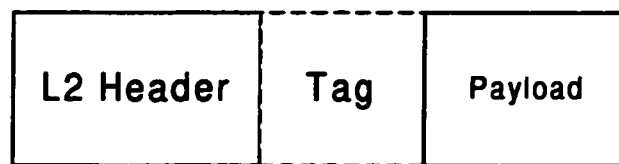


Figure 2.3: Tag Switching Packet Format.

Tag Switching technology uses *Label Swapping* mechanism. In Label Swapping the switch looks up the packets current tag in a table associated with the incoming port. Then, the outgoing port and tag are determined. The old tag is replaced with the new one, and the packet is forwarded on the designated out port. The routing process in tag switching can be summarized in three steps. first, an entry router attaches tags to packets based on the routes selected. Next packets are switched based on their tags. finally, an exit router strips off the tags and hands over the packets.

Routing in tag switching networks is illustrated by an example using the network shown in Figure 2.4. The router R1 receives packets from Host1 destined to Host2.

After classification R1 tags each packet of a flow with a particular tag according to the traffic type and the network tagging policy then it forwards tagged packets to the next hop in their route. Intermediate routers forward packets on the tag basis using label swapping mechanism. The network edge router towards the destination removes tags and forwards packets to their destination at Host2.

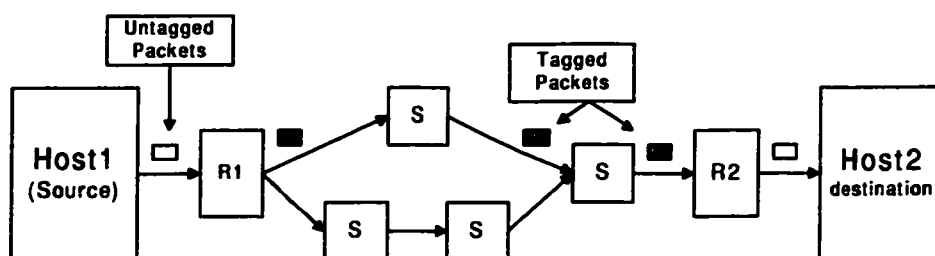


Figure 2.4: Routing with Tag Switching.

MPLS is expected to satisfy the need for service differentiation and to enable Internet Service Providers (ISPs) to answer customers demand for dependability. To provide these capabilities in networks, the basic traffic-forwarding paradigm of present-day IP networks must be enhanced to support traffic engineering [37]. Traffic engineering includes many aspects of network performance such as, guaranteed QoS, improved network resources utilization, and providing features for quick recovery in case of a node or a link failure [38].

MPLS does not consider computation of routes with constraints. To provide this facility MPLS designers are working on extending the existing IGPs like OSPF and

IS-IS to carry information about links such as maximum bandwidth and reserved bandwidth.

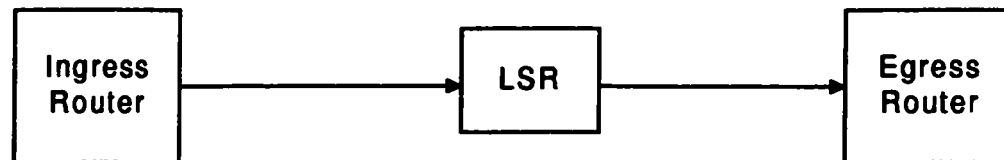


Figure 2.5: MPLS Routing Domain.

Figure 2.5 represents the essential parts in an MPLS enabled network. A detailed description of these components, their functions, and how they interact is given in Chapter 4.

2.10 Comparison among Intserv, Diffserv, and MPLS

The main features of the major three approaches that address the issue of QoS routing (Intserv, Diffserv, and MPLS) are compared and presented in Table 2.2. Differences among them affect everything from cost to all other performance aspects.

Intserv requires each router to maintain state information of each flow. This results in lack of scalability which is the major drawback of Intserv. Diffserv internal routers, on the other hand, do not have to maintain any state for reservations. The operation here is based on checking the service bits (Expedite, Assured or both). In MPLS resource reservation

Table 2.2: Comparison among intserv, Diffserv, and MPLS.

	Intserv	Diffserv	MPLS
Scalability	Not scalable	Scalable	Scalable
Layer of operation	Layer-3	Layer-3	Combines functions of layers 2 and 3 but remains independent of both
Compatibility	Compatible with ATM networks	Compatible with IP and ATM networks	Compatible with IP, ATM, frame relay
Router design	Complex	Simple due to reduced router state	Complex due to complicated router state
Router storage capacity required	High	Low	High
Scheduling	WFQ	Strict priority	WFQ and strict priority
Implementation	Complex	Simple	Complex
Advantages	Provides three service: best effort, controlled load, guaranteed delay	No packet classification and flow states at inner router. No signaling is required	No routing tables at inner routers. Simple forwarding mechanism
Disadvantages	Policy and security issues need to be addressed/resolved	An aggregate flow may receive same service level. No QoS absolute guarantee	Dynamic-connection admission-control and security issues need to be resolved
Protocols used	RSVP, IRP routing protocols	IRP routing protocols	Enhanced RSVP, CR-LDP, IRP routing protocols
Granularity of Differentiation	Fine granularity per flow basis	Coarse granularity on aggregate flows basis	Fine to Coarse granularity
Control overhead	High	Low	Low

is made by setting up states in routers and refreshing these states. Every router along the path of a connection has to support these reservations and has to maintain a state for each flow. The number of individual flows in a network can be very large which leads to lack of scalability but if flow aggregation [39] is allowed, MPLS is scalable.

Considering the router design, in Intserv each packet must be classified at each router. A lot of processing power is required to generate control messages to maintain reservation states which results in a complex router design. In Diffserv, no further classification is needed and no signaling is required which makes the router design simple. In MPLS, unless aggregation is enabled, core routers must preserve reservation state of each flow. This needs a considerable amount of signaling and processing which results in a more complex router.

Considering the memory storage capacity, the need to keep states for each flow requires large storage capacity in both Intserv and MPLS unless appropriate flow aggregation mechanism is adopted. Since no states are kept in Diffserv, the required storage capacity is minimal.

Considering scheduling, each router in Intserv and MPLS networks must keep a separate queue for each flow assuming that flow aggregation is not adopted. Organizing numerous queues makes scheduling very difficult. However in Diffserv scheduling is very simple and each router keeps only two queues one for EF service and the other for BE service (or may be BE and AF services).

Intserv has fine granularity in service differentiation since services are provided on flow basis. Diffserv has coarse granularity since services are provided on aggregate flow

basis. MPLS granularity could be fine or coarse based on the way of classing flows; fine granularity for flow based classing and coarse granularity for aggregate flow based classing.

The major advantage of Intserv is that it provides better service since it has fine differentiation granularity. Its disadvantages could be summarized in its complex operation. Diffserv advantages include simple operation and its disadvantages include lack of means to supply information about resources state in the core routers. MPLS main advantages are simple forwarding and separation of control/routing functions from forwarding. One of the major challenges facing MPLS is how to obtain the relevant QoS information.

In a workshop held to discuss issues related to QoS in Internet2, Van Jacobson raised a number of questions that list the major challenges for differentiated services [40]. The main questions which he presented are:

1. Who decides what users get to request special service?
2. Where is organizational policy on use of limited bandwidth implemented?
3. Who tells the edge router what to tag or label?
4. Who makes sure that simultaneous uses of special service fit within allocation?

It is of fatal importance for the Diffserv designers to face these challenges and to find answers to these questions in order to improve Diffserv performance as well as operation. Jacobson suggested, as a solution, the use of a **Bandwidth Broker (BB)**. The BB is a database where priority and limits for users as well as user credential are stored. It is included as part of the network infrastructure to have secure association with all routers. BB can resolve conflicts because requests go from user to the BB for authentication first,

then to the appropriate router. Jacobson in his presentation detailed the operation of the BB including where it could be located, who talks to the BB and how. This has been with an example of a campus with 10 Mbps every where. According to his analysis using a BB in that network, an allocation from a single 10 Mbps *Premium* bandwidth pool allows 300 simultaneous voice/video sessions with no topological knowledge in allocator. The effect of the BB on the network performance basically relies on the database developed there. With appropriate operation it is expected to enhance the differentiated services provided to users and improve the network performance.

In MPLS networks, the number of flows may be very large. The number of control messages for making resource reservation for this number of flows is also large. Maintaining state information for all the flows requires a large storage capacity. Managing the large number of queues and control signals requires a lot of processing power. In short this limits the MPLS scalability and raises the need for a solution which could be flow aggregation. Aggregation allows the network to deal with aggregated flows instead of dealing with each flow individually. Aggregation reduces the amount of signaling, storage capacity required and simplifies processing. However the major concern is who should be doing it? The network administrator or this should be done by some other means? If the administrator should aggregate how and to what limit he should aggregate?

Francois *et. al* [41] proposed as a solution to allow the administrator to select how to map Diffserv *Behavior Aggregate* (BA) to MPLS. Their solution relies on using LDP and combined use of single class and multi-class LSPs to give high flexibility in aggregation and at the same time allows for per flow service. Their solution operation steps can be

summarized in the following:

1. The Service Provider configures the LSRs to map between each *Per Hop Behavior* (PHB) of Intserv and a value of the *EXP* field to provide *BAndwidth reservation* (BAs) supported over the single/multi-class LSP.
2. The Service Provider configures the LSRs with the scheduling behavior for each single/multi-class LSP.
3. The LSRs signal the establishment of LSPs using LDP.

Their solution absolutely relies on the service provider or the network administrator. It has not been put to practice yet.

Chapter 3

Multiprotocol Label Switching

Multiprotocol Label Switching (MPLS) is an Internet Engineering Task Force (IETF)-specified framework that provides efficient classification, mapping, routing, forwarding and switching of traffic flows through the network [8]. MPLS is developed to satisfy the need for service differentiation and expected to play an important role in improving services provided by Internet Service Providers (ISPs) [9].

MPLS defines means to map layer-3 traffic to connection-oriented layer-2 transports. Deployment of MPLS necessitates labeling IP packets with labels that define both route and priority of a traffic flow. Packets are served in the MPLS network based on these identifiers.

3.1 Background

In conventional IP networks, each router in a path of a packet analyzes that packet's header and consults its routing table to select the next hop. Each router makes its selection independent of the other routers. Selecting next hop is actually a two step operation. Classification of the incoming packets and their assignment to a set of *Forwarding Equivalence Classes* (FECs), and forwarding each FEC to the next hop.

In MPLS networks, the classification of a packet and its assignment to a particular FEC is done once only at the ingress. Each FEC is assigned a label which is inserted in each packet header and all packets of an FEC follow the same path. Once a packet is labeled the rest of its forwarding in the network is done according to a process known as label switching. According to this process a label is used by a core hop as an index into a table that specifies the next hop and a new label. This means forwarding is driven by labels and no further classification or header analysis is done at each router. Label switching has major advantages: It results in a simplified hardware and software at the core routers, and it reduces packets delay and improves network efficiency.

3.2 Motivation for MPLS

MPLS networks has two major features. The partition of functional units and the use of Label Forwarding-Swapping Algorithm. Beside these distinguishing features, which will be described later, the following are considered to be the major motivations for MPLS:

- MPLS mechanism enables explicit route selection in IP networks. This option results

in a routing method similar to source routing. The slight difference between the two methods is that in source routing, a source requests a path where as in explicit path routing, the source enforces the selection of a certain prespecified route.

- MPLS eliminates the need for routing tables at the inner routers since only forwarding tables are needed.
- MPLS provides simplified mechanism for packet-oriented traffic engineering and multi-service functionality.

3.3 MPLS Terminology

In this section the widely used terms in MPLS technology are described.

Forwarding Equivalence Class (FEC): A set of packets to be forwarded in a network over the same path and treated in the same way at the inner routers.

Label: A fixed-length (20 bits) unit contained in the packet header to identify its FEC.

Label Switching Router (LSR): An MPLS network router. It is a router that supports MPLS-based forwarding and enables label switching.

MPLS Edge Router (ELSR): An MPLS router that connects an MPLS domain with a node which is outside of the domain.

MPLS Ingress Router: An ELSR that handles traffic as it enters an MPLS network.

MPLS Egress Router: An ELSR that handles traffic as it leaves an MPLS network.

Label Switching Path (LSP): It is a path in an MPLS network that is created by connecting one or more LSR.

Label Information Base (LIB): A table each LSR creates on the receipt of label binding. This table specifies the mapping between the input label and the output label.

Label Distribution Protocol (LDP): It is a protocol that is used to distribute labels in an MPLS network. It is a newly developed protocol produced by the MPLS designers to support label allocation, distribution, and binding, as well as transferring information about explicit routes.

3.4 MPLS Header and Label

MPLS can exist alone or with other network technologies such as ATM and frame-relay. When MPLS is used with ATM or frame-relay, a label is encapsulated in Layer-2 header of the packet. If the label does not fit there, it is contained in a separate MPLS header known as the shim header as in the case of IP networks. A shim header is a 32 bit header inserted between the IP header and layer-2 header as shown in Figure 3.1. MPLS header has the format shown in Figure 3.2.



Figure 3.1: Packet Format in MPLS Domain showing Shim Header.

Each field in MPLS header has specific functions. These fields and their functions are defined as follows:

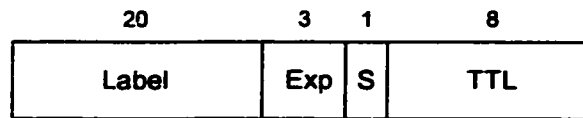


Figure 3.2: Structure of MPLS Header.

1. The label field, *label* (20 bits): Carries the actual value of the label. The label field is used by an inner LSR as an index to reference its forwarding table. This means a label has a local significance for the LSR which has created it. In other words labels correspond to links between LSRs only.
2. The experimental field, *Exp* (3 bits): Contains the class of service (CoS).
3. The stack field, *S* (1 bit): The ingress router may insert multiple labels (headers). In this case, labels are stacked and the stack field is used to indicate the end of the stacked labels. The label at the bottom of the stack is the level one label and the forwarding decision is based on the label at the top of the stack. The stack field bit is set to one for the level one label, and zero for all other stacked labels.
4. The Time-to-live field, *TTL* (8 bits): Provides a function similar to IP TTL function.

3.5 Label Switching Path (LSP)

An LSP is a path in an MPLS network that contains a set of LSRs including the two edge LSRs. LSPs allow traffic to flow in one direction only, from the source to the destination. For bidirectional traffic, two LSPs must be set, one for each direction.

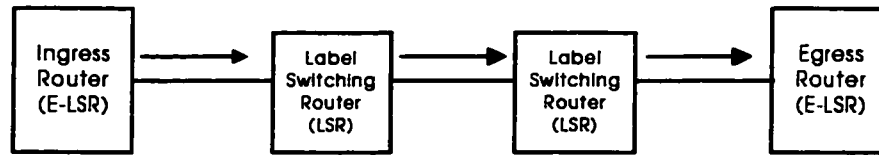


Figure 3.3: A label Switching Path.

Figure 3.3 represents an MPLS network that is composed of two edge LSRs (ingress and egress) and two core LSRs. The arrows indicate an LSP that starts at the ingress LSR, covers both core LSRs and ends at the egress LSR. An LSP may be developed using the IGRP routing information or it may diverge from that to facilitate traffic engineering implementation. LSRs included in an LSP may also be specified by the ingress router as the case with explicit route selection.

3.5.1 Single and Multi-Class LSPs

Classification policy decides which Diffserv forwarding classes would be supported by an MPLS network. It also specifies how many forwarding classes a single LSP may contain. MPLS supports two sorts of LSP classes: *single-class LSP* and *multi-class LSP*. A packet forwarding class is indicated in its MPLS header *Exp* field.

Single-Class LSP: When MPLS packets that are sent along an LSP belong to a single Diffserv forwarding class, then the LSP is a Single-Class LSP. In this case, there is no need to indicate the forwarding class of each packet because it can be derived from the label information.

Multi-Class LSP: When MPLS packets that are sent along an LSP belong to more than

one Diffserv forwarding classes, then the LSP is a Multi-Class LSP. In this case, service class of each packet, as well as possible drop precedence, need to be indicated. Each MPLS packet of such a Multi-Class LSP is forwarded according to the service class information contained in its MPLS header *Exp* field. If the service class of an MPLS packet is not among those classes listed in the corresponding LSP setup message, the packet is discarded.

3.5.2 Route Selection

Route selection refers to the method adopted to set routes or paths in a network; in MPLS it refers to the method used to specify an LSP for an FEC. MPLS supports two methods for route selection: *hop-by-hop* routing and *explicit routing*.

1. *hop-by-hop*: Allows each LSR to select next hop for each FEC independent of other LSRs as in traditional IP networks. An LSP that is set using this technique is a *hop-by-hop routed LSP*.
2. *explicit routing*: Does not allow each LSR to select the next hop independent of other LSRs in the network. Using this method, a single LSR, mostly the ingress router, specifies the entire LSP. Explicit routing has special importance for some applications such as, traffic engineering and policy routing.

Table 3.1 [35] lists the comparisons of the basic features of both routing methods.

Table 3.1: Comparison of Hop-by-hop Routing and Explicit Routing.

Hop-by-hop Routing	Explicit Routing
Distributed routing of control traffic.	Source routing of control traffic.
Builds a set of trees.	Builds a path from source to destination. Requires creation mechanisms.
Existing routing protocols are destination prefix based.	Has high routing flexibility, routing can be policy-based or QoS-based.
Difficult to perform traffic engineering.	Adapts well to traffic engineering.
Reroute on failure impacted by convergence time of routing protocol.	LSPs can be ranked which results in very quick rerouting. Backup paths may be set for restoration.

3.6 Label Assignment and Distribution

Label assignment and distribution is one of the fundamental operations in MPLS networks. The decision about binding a label to an LSP is made by the downstream/upstream LSR with respect to that binding. Upstream and downstream refers to the LSR which initiates the assignment and distribution of labels. To understand the upstream and downstream concepts, let us consider the setup shown in Figure 3.4. In this figure, router 3 is the downstream neighbor of router 2 considering host D as the destination. Router 2 itself is the downstream neighbor of router 1 and at the same time the upstream neighbor of router 3 towards D.



Figure 3.4: Upstream and Downstream LSRs.

Downstream Label Distribution Methods: Downstream label distribution can be performed by two methods: downstream and downstream on-demand. Any two adjacent LSRs must agree on the type of method used. It is possible to have both methods

simultaneously enabled in a network.

In downstream label distribution, when an LSR discovers a next hop for a certain FEC, it assigns a label for that FEC and distributes it to its upstream neighboring LSR. The procedure is performed at each LSR in the LSP till the ingress LSR.

On the contrary in downstream on-demand label distribution, an LSR generates a label but does not communicate it to its upstream neighbor LSR. When an upstream LSR needs a label for a particular FEC it requests the corresponding label from its downstream neighbor LSR. The downstream neighbor LSR replies by communicating the requested label.

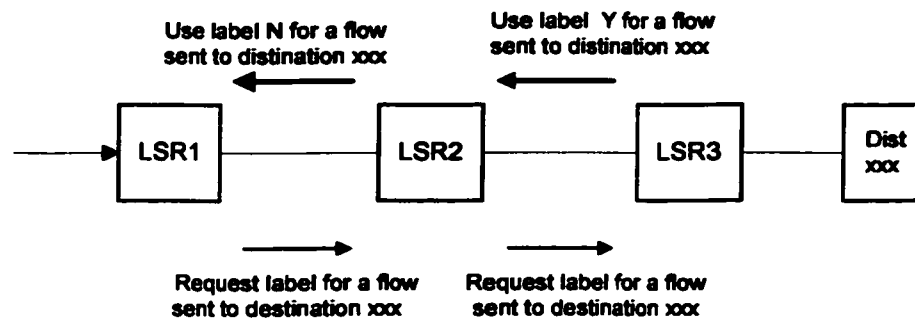


Figure 3.5: Label Distribution.

Figure 3.5 shows the distribution of labels in both methods. It is assumed that a flow heading to destination *xxx* has already been classified and mapped to an FEC. In the figure, the upper thick arrows indicate the downstream label distribution method. LSR3 communicates its binding label for an FEC to LSR2, which itself communicates its binding label to LSR1 (the upper thick arrows).

The lower thin arrows plus the upper thick arrows represent the downstream on-demand label distribution method. LSR1 requests a binding label for an FEC from LSR2

which requests a label from LSR3 (the lower thin arrows). LSR3 replies by communicating the requested label to LSR2, which communicates its label to LSR1 (the upper thick arrows).

3.7 Label Distribution Protocols

In MPLS networks, no specific protocol has been specified for label distribution. Two signaling protocols have been proposed for LSP establishment and label distribution:

1. Label Distribution Protocol (LDP) [42], and its extension Constraint-based Routing-Label Distribution Protocol (CR-LDP) [43].
2. Extended or enhanced RSVP [44] and extended RSVP for traffic engineering (TE-RSVP) [45].

3.7.1 Label Distribution Protocol (LDP)

LDP is used by the LSRs to exchange label/FEC binding information. LDP is a set of procedures and messages that enable the LSRs to map network-layer routing information directly to data-link layer switched paths and hence establish LSPs among them [42].

Any two LSRs that use LDP to exchange messages are known as *label distribution peers* with respect to the binding information they exchange, and they have a *label distribution adjacency* between them. LDP enables the negotiation between any two label distribution peers to establish connection or message exchange. LDP uses TCP, as a reliable protocol, for signaling.

LDP is extended to the Constraint-based Routing LDP (CR-LDP) to implement constraint-based routing [43]. CR-LDP is used to setup explicit routes that satisfies some constraint(s) such as bandwidth and/or delay to guarantee applications QoS requirement.

LDP Message types: The following are the main LDP message types:

1. **Discovery Messages:** Used by the LSRs to announce and maintain their presence in a network, for example, the hello message.
2. **Session Messages:** Used to establish, maintain, and terminate sessions.
3. **Advertisement Messages:** Used by the LSRs to request a label and to announce the existence of a label that may be used by other LSRs. Advertisement messages are also used to change and delete label binding.
4. **Notification Messages:** Notification messages are of two kinds:
 - (a) **Advisory notifications:** Provides information about the LDP session or the status of some of the exchanged messages.
 - (b) **Error notifications:** Detects and reports major signaling errors. Whenever an LSR receives an error notification message, it responds by terminating the current LDP session and deleting all labels recorded during that session.

3.7.2 Enhanced RSVP

Bruce Davie *et. al* [44] proposed to use enhanced RSVP for label distribution. Daniel Awduche *et. al* [45] have proposed extensions to RSVP to support traffic engineering

with MPLS. They argued that making label distribution a part of path/reservation setup process of RSVP is the most efficient method for label distribution. It also improves the network over all performance.

RSVP messages were augmented with new functions to support label allocation, distribution, and binding. The following objects are added or enhanced, if they already exist in RSVP, to support MPLS operations.

LABEL_REQUEST OBJECT (LRO): It is used to request a label for an LSP. There are three types of this object: one for ATM, the second for frame relay, and the third is general. The general type does not specify the label range, while the other two specify a particular range for label assignment. LRO is composed of the MPLS shim header and L3PID. The L3PID indicates the protocol to be used at layer-3 by an application.

EXPLICIT ROUTE OBJECT (ERO): Using this object, the ingress LSR can specify a predetermined LSP and enforces the use of that route instead of the conventional IP shortest path route.

RECORD ROUTE OBJECT (RRO): Using this object, the ingress router can receive Information about the LSP route. RRO has three major uses:

1. Discovers loops in layer-3.
2. Supplies detailed information about the LSP exact route.
3. Can be used as an input to the next path session.

LRO is a must object while ERO and RRO are optional objects.

Error messages constitute an important class of messages exchanged in MPLS networks

whether LDP or RSVP is in use as signaling protocol. An LSR normally generates an error message in one of the following three conditions:

1. The LSR is unable to assign a label for an LSP and hence it sends an error message that states MPLS allocation failure.
2. The LSR cannot support the L3PID. The error message in this case states an unsupported L3PID.
3. The LSR does not recognize the LRO and an unknown object error message is generated.

3.7.3 Explicit LSP Setup

The explicit path principle and the RSVP setup process are combined to setup explicit LSPs. To see how this is done, consider the network of Figure 3.6.

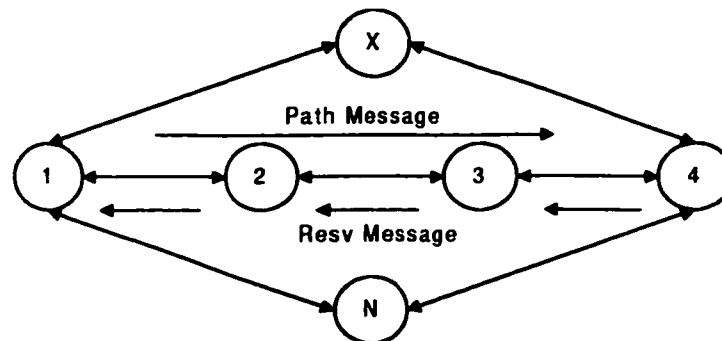


Figure 3.6: Network Topology for LSP Setting Example.

Assume that LSR1 wishes to transfer some data to LSR4. For this purpose it establishes an explicit LSP that contains LSRs 2 and 3 which means that it does not use any

of the two paths through router X or outer N although either may be the shortest path to LSR4. This could be for any of the following reasons:

1. To enforce some routing policies or to avoid a certain route.
2. The specified LSP satisfies some of the QoS requirements of the application to be served.

To setup the required explicit LSP, LSR1 generates an RSVP **path** message with (ERO) enabled. The use of this object enforces the selection of the specified path. The message also includes the addresses of the nodes that should be included in this subdomain. Each of LSRs 2, 3, and 4 explore the path and records the incoming port number and the upstream router IP address to be used for forwarding the **RESV** message it receives from the receiver. Finally, LSR4 checks the *Flowspec* of the path message, makes reservations, and generates a **RESV** message. This message specifies *Tspec* and *Rspec* that has been reserved to meet the application QoS requirements. Having this operation completed, an explicit LSP is set and LSR1 would use it to transfer the application. After completion, the reservation state at each LSR times out and the LSP is removed.

3.8 MPLS Networks Main Features

The two major features of MPLS that distinguish it from the other proposals for providing differentiated services like Intserv and Diffserv are: The partitioning of functional units and the use of Label Forwarding-Swapping Algorithm.

3.8.1 Partitioning of Functional Units

MPLS has two main building blocks, a control component and a forwarding component (see Figure 3.7) [9].

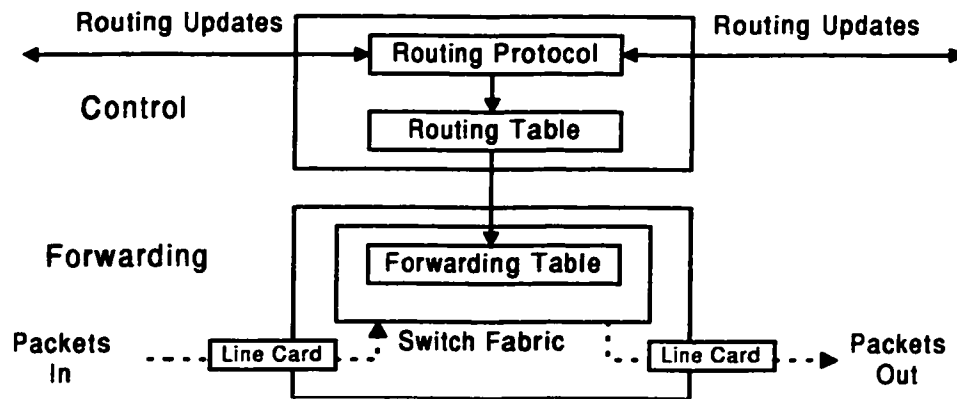


Figure 3.7: MPLS Main Building Blocks.

Control Component: It exchanges information with other routers to build and update routing and forwarding tables using standard routing protocols such as OSPF and BGP. The control unit provides a flexible packet classification capability that is expected to enable ISPs to improve their efficiency and to provide new services beyond those known in traditional networks.

Forwarding Component: It examines the IP packet header, then consults the forwarding table to specify the next hop and the output port and finally directs the packet from the input to the output port.

The partitioning results in a high flexibility in the design and modification of each of the two blocks. The two units should coordinate to manage routing table update. This is the only limit imposed on the separate design and modification of the functional blocks.

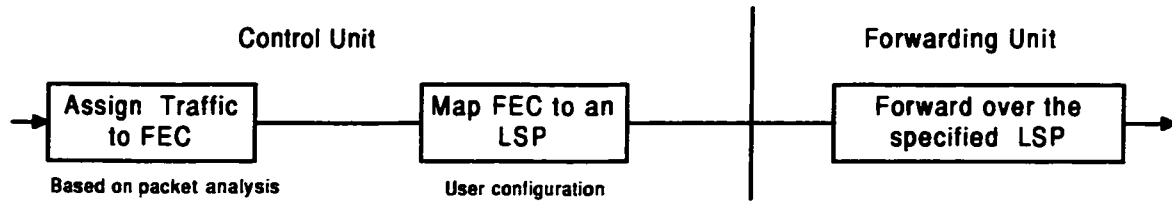


Figure 3.8: Routing Function in MPLS Networks.

Figure 3.8 [9] illustrates the way the two functional units cooperate to perform routing in MPLS networks. The control unit classifies an incoming packet and assigns it the corresponding FEC based on the policies adopted in the network. Then, it maps the FEC to an LSP using labels. Next, the forwarding unit forwards the packet using the specified LSP. The control component process represents the route selection and the forwarding component process represents the data transfer using the selected route. These two processes constitute routing in data networks. Further, forwarding is performed at each of the inner LSRs, where aggregate forwarding resources (buffer space, bandwidth, and scheduling policy) are allocated in each router for each Diffserv forwarding class.

3.8.2 Label Forwarding-Swapping Algorithm

Figure 3.9 depicts the label swapping process. In this scenario it is assumed that the ingress router has received an IP packet with some destination address. First, the classification, mapping to an FEC, label assignment and distribution processes explained earlier are completed. Then, the network inner LSRs forward the packet using the label-swapping algorithm.

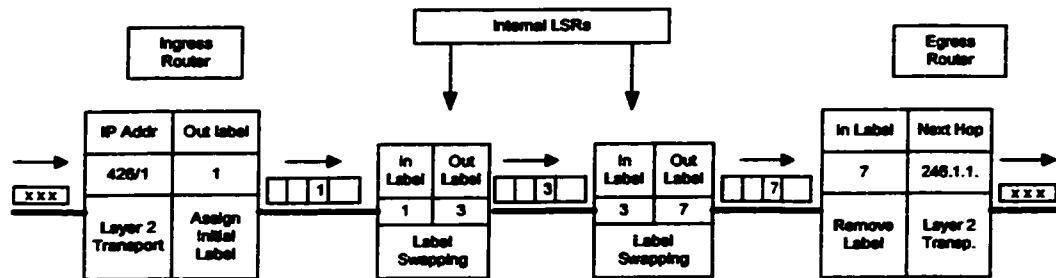


Figure 3.9: Label Swapping Process.

Using this algorithm, when a labeled packet enters a router, the router uses the input port number and the label as entries to search its forwarding table. As a result, the router specifies the outgoing label, the outgoing port, and the next LSR in the LSP. Next, the router swaps the labels and passes the packet to the outgoing port for transmission to the next LSR. At the egress router, the router discovers from label value that it is the destination LSR and hence removes the label and forwards the packet to the receiver.

Using Label swapping forwarding algorithm is advantageous in the sense that [9]: the service provider gains high flexibility in assigning packets to FECs, e.g, on the basis of the destination/source address, application type, CoS. It also enables the service provider to setup customized LSPs that support specific application requirements e.g, minimize number of hops or meet bandwidth requirements. Finally, it has the ability to map an FEC, assigned to a user type of traffic, to an LSP. This in short results in better service provider control over traffic flows and hence better network efficiency and resource utilization.

3.9 Routing in MPLS Networks

The essential steps that must be taken for a data packet to travel through an MPLS enabled network are: Label creation and distribution, forwarding table creation at each router, LSP creation, label insertion/table lookup and packet forwarding.

The time sequence diagram shown in Figure 3.10 represents the processes needed to establish a session and transfer an application in an MPLS network that consists of an ingress, an egress, and a number of core LSRs. The process starts by the sender issuing a

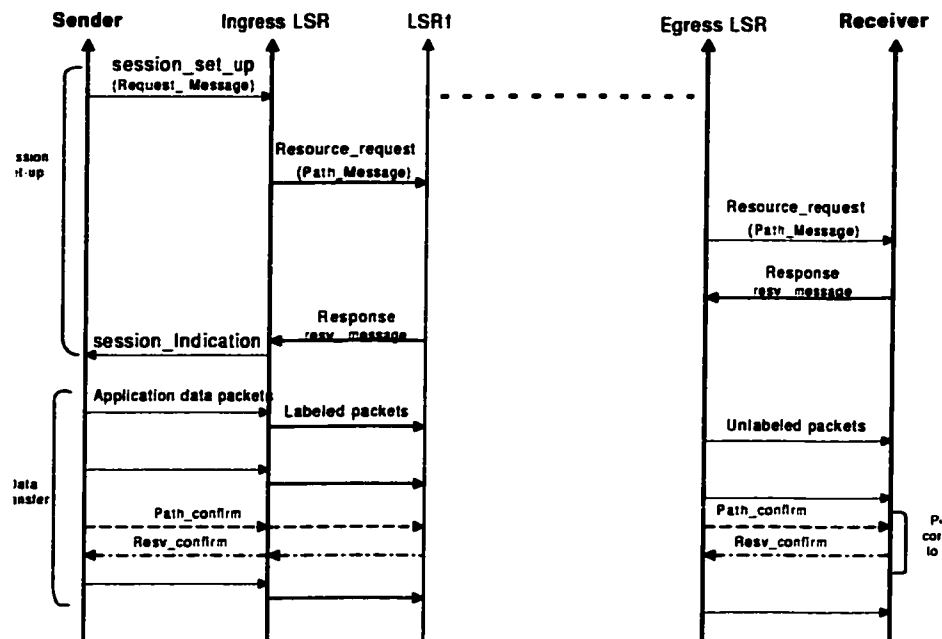


Figure 3.10: Time Sequence Diagram for MPLS Routing.

connection establishment request using a *session_set_up* message. This message contains the IP addresses of the sender and the receiver plus information about the requested service. The required service information might be in a qualitative form as *use a certain type of service*; it might also be in a quantitative form as *the application needs this much*

of bandwidth or a maximum delay of this amount.

The *a session_set-up* message is sent to the ingress router. The ingress router, upon receiving the request message and having full information of the application requirements and the network current state, uses the admission policy to decide whether to admit this application or not. Assuming that the network state indicates that it can handle the application, the router consults its routing table developed using OSPF algorithm to choose a route for this application. Next, the router generates an RSVP **Path** message and forwards it to the next LSR in the selected route. The **Path** message must contain, besides the normal RSVP **Path** message objects, an *LRO* and may include the *RRO*. In case of explicit route selection, the **Path** message must have the *ER* option set to enforce the selection of the prespecified route.

Router LSR1, upon receiving the **Path** message from the ingress router, records in its *Path State Block* (PSB) the IP address of previous hop, and the incoming port number. This information is used to forward the **RESV** reply message back to the ingress router. If the *RRO* and *ERO* are used, they are also recorded in the router's PSB. Then LSR1 forwards the **Path** message to the LSR in the way downstream towards the egress router and the receiver. Each of the intermediate LSRs repeats the same procedure of LSR1 till the **Path** message arrives at the egress router. The egress router, upon receiving the **Path** message from its upstream neighbor LSR, repeats the same procedure of the core LSRs and finally, forwards the **Path** message to the receiver.

The receiver, upon receiving the **Path** message from the egress router, examines the message *FLOWSPEC* object, generates a **RESV** message that specifies the *FLOWSPEC*

of the application it will receive. Receiver oriented resource reservation enables the receiver to decide about the type of application it wishes to have. The receiver sends the **RESV** message to the egress router.

The egress router, upon receiving the **RESV** message from the receiver, binds a label with a certain value for packets of this application. The egress router stores this value in its forwarding table and includes it in the **RESV** message. The egress router forwards the message using the information it has stored in its PSB during the **Path** message trip towards the receiver.

Router LSR1, upon receiving the **RESV** message from its downstream neighbor LSR, stores the label value received in the **RESV** message in its forwarding table. LSR1 also generates its own label and binds it with this application FEC. The assigned label value is stored in LSR1 forwarding table and inserted in the **RESV** message which is forwarded to the ingress router. Whenever a labeled packet arrives at LSR1, the label is replaced with the one previously received from the downstream neighbor LSR and stored in the forwarding table.

The ingress router, upon receiving the **RESV** message from the receiver, stores the label value it has received in the message in its forwarding table. The ingress router binds this label value to the FEC in which it has mapped the application traffic. Next, the ingress router issues a *session indication* message to the sender.

The sender having the required resources reserved and the LSP set, starts transmitting its data for as long as there is data to send. The ingress router receives packets from the source, maps them to their corresponding FEC, inserts the corresponding labels, and

forwards them to LSR1. Whenever LSR1 receives a packet it examines the label and consults its forwarding table to replace it with the label corresponding to the next hop then forwards it to the next LSR. For example, consider the network of Figure 3.11, The ingress router receives packets from the two hosts, 1 and 2. Packets received from Host1 are sent labeled with the corresponding label to LSR2 and packets received from Host2 are sent, also labeled, to LSR4. Host1 traffic LSP contains LSR2 and LSR3 besides the edge routers while Host2 traffic LSP contains LSR4 and LSR5 beside the edge routers. The core LSRs forward the arriving packets by consulting their forwarding tables and replacing the in-labels with the corresponding out-labels.

Whenever the egress router receives a packet, it examines its label. Discovering from the label value that it is the egress router for the application to which this packet belongs, it removes this label and forwards the packet using conventional IP routing to its destination. Again considering the network of Figure 3.11, the egress router LSR6 removes the labels from the arriving packets and uses conventional IP forwarding to take the packet to its final destination. Host1 traffic is forwarded to its destination at Host3, and Host2 traffic is forwarded to its destination at Host4.

This process continues till the end of transmission. Refreshing messages to maintain the reservation states in all the network routers are exchanged periodically among the nodes. In this regard, **Path_confirmation** messages are initiated from the source and upstream routers and sent to the downstream routers. The **Resv_confirmation** messages are initiated from the receiver and the downstream routers and sent to the upstream routers towards the sender. These messages are initiated every 30 seconds. At the end of

the transmission, either the sender or the receiver can tear the connection down by ceasing to send refreshing messages.

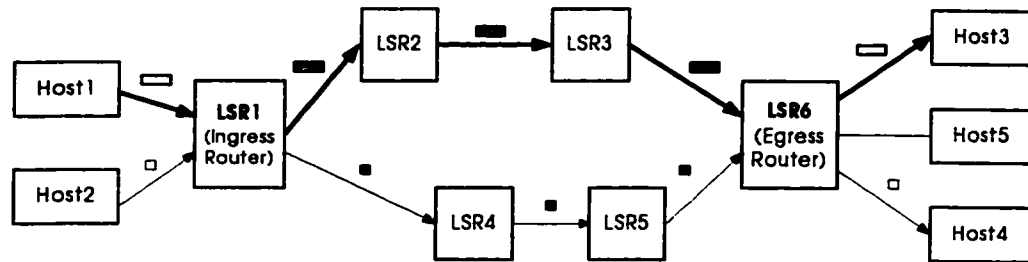


Figure 3.11: Routing in MPLS networks.

3.10 MPLS Applications

There exist three main applications for MPLS in ISP networks:

- Traffic Engineering.
- Class of Service (CoS).
- Virtual Private Networks (VPN).

3.10.1 Traffic Engineering

Supporting traffic engineering is the principal and most important application of MPLS.

Traffic engineering is the operation of controlling traffic flow in a network in order to

optimize resources utilization and improve performance. Traffic engineering performance objectives are either traffic oriented or resource oriented.

The objectives of traffic oriented performance are minimization of packet loss, minimization of delay, maximization of throughput, and enforcement of service level agreements. Resource oriented performance objectives, on the other hand, aims to the efficient management of network resources.

To satisfy traffic engineering requirements, a network should achieve these performance objectives as well as provide guaranteed QoS and fast recovery in case of link or node failure.

Four basic components are needed for traffic engineering in packet networks [38]. These components are outlined as below:

1. *Distribution of topology information:* There is always a need for a mechanism to advertise information about the initial network state, and to update it according to any changes in the network topology or traffic state.
2. *Path selection:* Nodes use the topology information to compute reachability cost. Results produced based on the algorithm applied are used to select paths. Paths may be selected on basis other than the shortest path for example to satisfy some constraints like bandwidth, delay, etc.
3. *Directing traffic along the computed paths:* After selecting and setting paths, traffic is forwarded along these paths.
4. *Traffic management:* It involves a framework and mechanism to enable the net-

work to provide the user traffic with QoS requirements. The mechanisms needed to provide traffic management include mechanisms for admission control, flow identification, and traffic policing and scheduling.

Traffic engineering enables the network manager to forward traffic on routes other than the shortest path calculated by traditional routing algorithms. This is normally done to avoid congestion and make better resource utilization. Applying traffic engineering principles results in better and more efficient services.

To build an MPLS system for traffic engineering, the following design parameters must be determined [46]:

1. The geographical scope of the MPLS system: The region to be contained in an MPLS network is specified by the authorized administrative policy.
2. The participating routers: The routers to be part of the MPLS system, i.e., the ingress, egress, and core LSRs. This could also be decided by the administrative authority. The system performance is greatly affected by this factor since it has effects on the amount of control overhead, routing complexity, number of LSPs, and link utilization.
3. The hierarchy of the MPLS system: The system administrator has to set the network hierarchy; small networks may be set in a full mesh resulting in a single LSP layer. For large networks this may complicate the situation, hence, the network may be divided into subregions. Each region is fully meshed to constitute the first layer; then, selected LSRs are meshed to form the second layer of LSPs and so on.

4. The bandwidth requirement of the LSPs: The LSPs traffic matrices should be specified in terms of bandwidth requirements.
5. The path parameters of the LSP: A constraint-based routing algorithm is used to compute the LSPs.
6. The priority of LSPs: Important LSPs are given higher priority and will gain an optimal path. This results in stable routing and improves resource utilization.
7. The number of parallel LSPs between each pair of endpoints: MPLS allows the existence of more than one LSP between the same pair of edge routers resulting in a number of parallel LSPs. The parallel LSPs can be used to routed more easily and to balance links loads.
8. The affinity of LSP and the links: Links and LSPs are assigned different colors to enable policy routing, which may result in preventing the LSPs from using some links or prefer certain links over the others.
9. The adaptability and resilience parameters of LSP: The ability to switch to a better path in case of availability i.e., LSPs reoptimization and rerouting in case of failure should be specified and agreed upon prior to LSPs setting.

3.10.2 Class of Service (CoS)

As discussed earlier, MPLS allows the existence of more than one LSP between the same two edge routers. Hence each LSP could be used to forward a different class of traffic. This enables the network to serve more than one application with different QoS requirements at the same time. This could be done by mapping each LSP to certain QoS. Then, the applications are mapped to a suitable LSP that satisfies the required CoS.

3.10.3 Virtual Private Networks (VPN)

A virtual private network VPN simulates the operation of a private wide area network (WAN) over the public internet [9]. An ISP desiring to provide his customers with VPN service needs to address the issue of using the IP address within a VPN and securing users data. VPNs should incorporate four basic functions:

1. Firewalling: To protect customer sites and provide safe access to the Internet.
2. Authentication: To assure legal information exchange between valid users.
3. Encryption: To secure and protect data transmitted across the Internet.
4. Tunneling: To provide a multiprotocol transport service and enable the use of the IP address within a VPN.

MPLS has a simple tunneling mechanism that would ease the matter of providing VPN service to the ISP customers. Tunneling in MPLS results in a set of LSPs that connect different edges of the network which can be used in offering VPN service.

3.11 Summary

MPLS addresses issues related to scalability and routing based on QoS. MPLS will play an important role in routing, switching, and forwarding packets through future networks as well as satisfying users service demands. MPLS performs the following functions:

- Provides a means to map IP addresses to labels that can be used by different technologies.
- Manages traffic flows of different applications.
- Operates with existing routing protocols such as RSVP and OSPF.
- Supports the IP, ATM, and frame-relay layer-2 protocols.

Chapter 4

Simulation and Results (1)

QoS provisioning implies segregation of traffic into distinct classes to provide special treatment to these classes. Implementing QoS concepts necessitates addressing the following issues:

- *Connection oriented service:* A mechanism is needed to provide conventional connectionless IP networks with connection oriented service.
- *Resource reservation:* To guarantee providing the required level of service, it is essential to have a method for reserving resources.
- *QoS routing:* It implies the ability to find routes that can satisfy QoS attributes (mainly bandwidth and delay) of the applications.
- *Admission control:* Refers to the policy adopted to handle applications requests for admission at the incoming edge routers.

- *Packet scheduling*: Refers to the method of queuing classes of service in the network core routers.
- *Real time transport protocol*: Transport protocol well suited to the transport of real time data is needed.

4.1 Performance Measures

Proper and accurate QoS provision requires the successful delivery of an agreed upon level of service. A level or class of service is characterized by a set of performance parameters. Performance parameters belong to four general categories:

- Overhead Traffic Metrics.
- Latency Metrics.
- Resource Utilization Metrics.
- Flow Satisfaction Metrics.

These metrics are used to differentiate the QoS level provided for a particular service. In this section, these metrics are detailed. Metrics used in this work are also specified.

4.1.1 Overhead Metrics

The extra overhead in MPLS networks is mainly due to the use of RSVP or LDP as a signaling protocol for setup and maintenance of LSPs. The performance metrics to measure overhead traffic consists of the following:

1. Volume of overhead traffic in bytes observed over the lifetime of a session. There is a fixed amount of overhead due to LSP setup and a variable amount that depends on the session duration since path confirmation messages are periodically exchanged.
2. Amount of memory needed at each router to maintain state information. This depends on the number of states needed to maintain resource reservations at a router.

MPLS provides services on per flow basis. As a result, the amount of overhead produced for reserving resources and maintaining reservation states (confirmation messages) increases with the number of flows. Considering the fact that the traffic flow in the Internet is increasing, the amount of refreshing or confirming reservation states in routers will increase severely. For example, consider the case of a network that has links of 100 Mbps capacity. Assuming all the flows in a certain time have a size of 64 Kbps each, the number of flows that can be accommodated in a link is about 1500 flows. Considering the number of links to a router, the amount of flows will be huge. Hence, signals exchanged to maintain states for these flows will consume relatively high bandwidth as well as router's memory.

4.1.2 Latency Metrics

Latency related metrics are end-to-end delay and delay variation. End-to-end delay, measured in milliseconds, includes: transmission, propagation, queuing, and switching delays. That is,

$$\text{End-to-end Delay} = T_x + T_p + T_q + T_s$$

where

T_x : The transmission delay. The transmission delay is equal to M/C where M is the average packet size and C is the link capacity.

T_p : The propagation Delay. The propagation delay is equal to d/v where d is the distance between two routers and v is the signal propagation speed. This speed is approximated to the light speed.

T_q : The queuing delay. The queuing delay is the delay experienced by a packet waiting its turn to be forwarded by a router.

T_s : The switching or processing delay. In IP networks the processing or switching delay is the time consumed in a router in analyzing a packet header and consulting its routing table to find the packet's next hop and output port. In MPLS networks, ingress router processing delay also includes the time to classify a packet, mapping it to a certain FEC, and labeling it. In the egress router, the processing time includes the time to strip off the label and conducting normal IP operation to send the packet to its destination. In the core routers, the processing time is the time the LSR spends consulting its forwarding table and swapping labels.

The packet end-to-end delay is measured in milliseconds. The delay jitter which is the variation in delay between successive packets is also measured in milliseconds.

4.1.3 Resource Utilization Metrics

In resource utilization, we are mainly concerned with the link utilization. Utilization refers to the percentage of a link being used to the total link capacity (bandwidth). The

bandwidth required by an application is measured in bits per second by counting the total packets delivered to a receiver times the packet size in bits. In this work, the bandwidth consumed by an application during its lifetime is measured.

4.1.4 Flow Satisfaction Metrics

Flow satisfaction is achieved by satisfying the flow QoS requirements. Satisfactory QoS in MPLS has the following main requirements:

- A means for labeling flows with respect to their priorities.
- Network mechanisms for recognizing the labels and acting upon them.
- Means for bandwidth reservation.

A flow is satisfied if a network meets its QoS requirements in terms of:

1. The bandwidth requirements of the flow are guaranteed.
2. The end-to-end delay remains within the end-to-end delay value tolerated by the flow.
3. The variation in delay does not exceed the maximum delay variation specified by the flow.
4. The packets dropped count does not exceed the packet loss limits specified by the flow.

Measuring end-to-end delay, delay jitter, and packet loss will show whether the network has the ability to meet the application requirements and hence achieves flow satisfaction.

4.2 Traffic Types and Characterization

Commonly used traffic sources in data networks are:

1. **Constant Bit Rate (CBR):** Applications that generate this type of traffic are audio traffic (e.g, telephony and voice mail) and non-compressed video traffic.
2. **Variable Bit Rate (VBR):** Applications that generate this type of traffic are compressed video traffic and LAN TV.
3. **Available Bit Rate (ABR):** Applications that generated this type of traffic are web browsing and E-mail.
4. **Unspecified Bit Rate (UBR):** Applications that generate this type of sources are traditional computer applications such as file transfer.

Table 4.1 gives current applications in the Internet. It also details each application traffic type and characterizes it in terms of traffic parameters and QoS attributes.

Table 4.2 gives bandwidth requirements for some of the applications listed in Table 4.1. The applications present in Table 4.2 are the most QoS demanding applications.

4.3 Simulation Environment and Objectives

The simulation tool used in this work is *NS* simulator [47], version 2.1b7a. To the basic NS simulator, we have also added the extensions made by Ahn [48] and the Diffserv patch developed by Sean [49].

Table 4.1: Applications Traffic Types and Characterization.

Application	Type	Traffic Parameters	QoS Attributes
Audio (non-compressed)	CBR	Rate and packet size or frequency of sending packets (typical packet size < 200 bytes)	Bandwidth (of 100-200 Kbps) end-to-end delay (≤ 200 milliseconds) packet loss (better than 1 in 10^4)
Audio (compressed)	VBR	Peak rate, on and off periods, and packet size (typical packet size < 200 bytes)	Bandwidth (of 128 Kbps) end-to-end delay (≤ 200 milliseconds) packet loss (better than 1 in 10^4)
Video (non-compressed)	CBR	Rate and packet size or frequency of sending packets	Bandwidth (80-400 Mbps) end-to-end delay (≤ 200 milliseconds) packet loss (better than 1 in 10^6 or 10^7)
Video (compressed)	VBR	Peak rate on and off periods burstiness and packet size (typical packet size < 200 bytes)	Bandwidth (1-7 Mbps) end-to-end delay (≤ 200 milliseconds) packet loss (better than 1 in 10^6 or 10^7) delay jitter (100 milliseconds for low quality and 5 milliseconds for high quality)
Web Browsing HTML	ABR	Rate on and off periods burstiness	Bandwidth provided should not be less than a prespecified value
Electronic Mail	ABR	Rate on and off periods burstiness	Bandwidth provided should not be less than a prespecified value

Table 4.2: Applications Bandwidth Requirements.

Application	Required Bandwidth
Voice conferencing (low quality)	64 Kbps
Voice conferencing (high quality)	128 Kbps
Video (non-compressed)	
EGA	80.6 Mbps
VGA	110.6 Mbps
SVGA	276.5 Mbps
NTSC	209.5 Mbps
PAL	400.2 Mbps
Video (compressed)	
Low quality video	256 Kbps
VHS VCR player quality	1.5 Mbps
Broadcast quality	5.0 Mbps
Studio quality	7.0 Mbps

We have performed several sets of experiments. These sets which are detailed in this Chapter and the following Chapter are outlined as below:

1. **Vanilla MPLS versus IP:** In this set of experiments, a vanilla MPLS network i.e., MPLS without differentiated service is evaluated against a traditional IP network.
2. **Diffserv MPLS versus Vanilla MPLS:** This set of experiments is targeted to evaluate MPLS combined with Diffserv against vanilla MPLS.
3. **Link Failure in Diffserv MPLS:** In this set of experiments, we intend to study the effect of a link failure on the network performance.
4. **Link Failure in Diffserv MPLS with Variable Loads:** This set of experiments examines the network performance under variable traffic loads of Best Effort (BE) type. The excess of Expedite Forwarding (EF) traffic on the network performance is also considered.
5. **Link Failure in Diffserv MPLS with multiple LSPs:** In this set of experiments the network reaction in case of link failure while there are more alternative LSPs is examined.

In all experiments admission control is performed manually. For the vanilla MPLS implemented in NS, the connection oriented service is provided by the LDP protocol. This is done by setting the LSPs which are used later by the traffic sources applications. Resource reservation is also done using the LDP protocol in the label mapping and distribution process. In vanilla MPLS experiments scheduling is not based on applications service types and hence a single queue, provided with best effort service, is built at each router.

In the Diffserv MPLS experiments the routing is based on (CR-LDP) of Ahn [48]. Scheduling in Diffserv MPLS simulation is performed according to Diffserv patch of Sean [49]. In this case scheduling is made per one queue for each traffic type. This means two queues are built at each router. One of the queues is provided with best effort service while the other queue is provided with expedite forwarding service. The expedite forwarding service class is assigned thirty percent of the network resources and the remaining seventy percent is reserved for the best effort traffic. The best effort traffic is given the higher share of the network resources because in reality most of the traffic is to be served as best effort.

Our simulation objectives can be summarized in the following:

1. To find out how an MPLS network behaves towards different types of traffic under different network states.
2. To test the effect of MPLS functionalities on the network performance and how differentiated services can be provided.
3. To examine and evaluate the MPLS reaction to link failure (see next Chapter).
4. To evaluate the fairness in MPLS enabled network and resources allocation.

4.4 MPLS Networks versus IP Networks

This simulation is conducted on MPLS enabled network, then repeated for a simple IP network with different simulation times (e.g., 5, 10, and 15 seconds). In both cases the

same simulation time and the same traffic loads and network conditions are applied. The network used in this simulation is shown in Figure 4.1.

Networks are subjected to a mix of traffic types to test the way MPLS handles different types of applications. Organizing the sources and destinations in the form shown in the figure results in having some sources sharing the same link and hence competing for the same resources. For example, in the MPLS network four sources are sharing links 3-5 and 5-7. In the IP network, three sources are sharing both links. This form of placing sources and destinations enables testing the networks fairness and examining the effect of competition for resources on the network performance.

Our objective in this experiment is to find out how MPLS basic mechanism affects the network performance. For each traffic source, we have measured the end-to-end delay and the delay variation. Bandwidth consumed by each traffic source is also measured.

4.4.1 Experimental Model Description

This simulation network consists of eight routers connected with links of 10 Mbps capacity and 10 millisecond propagation delay for each link. Five traffic sources, source 1 to source 5, are attached to different routers. Destinations are numbered according to their respective source numbers. Traffic characterization of the sources used in this experiment and their routes are provided in Table 4.3.

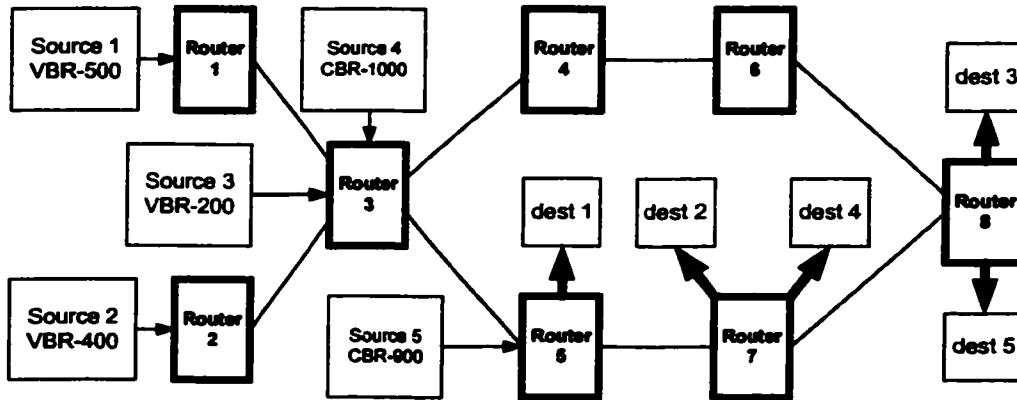


Figure 4.1: Test Model Configuration.

Table 4.3: Traffic Sources for the IP versus MPLS Simulation Study.

Source (Type)	ON - OFF Periods	Rate (Packet Size (Bytes))	Starting-stopping	Route	An Example
Source 1 (VBR)	400 - 600 msec	5 Mbps (500)	Randomly	Routers 1, 3, 5	Compressed video (broadcast quality)
Source 2 (VBR)	500 - 500 msec	1 Mbps (400)	Randomly	Routers 2, 3, 5, 7	Compressed video (low quality)
Source 3 (VBR)	400 - 1000 msec	400 Kbps (200)	Randomly	Routers 3, 5, 7	Web browsing HTML
Source 4 (CBR)	each 5 msec	200 Kbps (1000)	1.0 - 3.8	Routers 3, 5, 7	Voice (Noncompressed)
Source 5 (CBR)	each 5 msec	180 Kbps (900)	1.2 - 3.7	Routers 5, 7, 8	voice (Noncompressed)

4.4.2 Results and Discussion

First, the early stated mathematical formulae (see Section 5.1.2) is used to calculate the transmission and hence the end-to-end delay of the traffic of each of the five sources.

A packet of source 1 (VBR-500) has a transmission delay of 0.4 milliseconds at each router which means a total transmission delay of 0.8 millisecond. it will also experience a propagation delay of 20 milliseconds. This makes a minimum delay of 20.8 milliseconds plus queuing and switching delays. Figure 4.2 shows simulation results for traffic of source 1 (VBR-500) end-to-end delay in both networks.

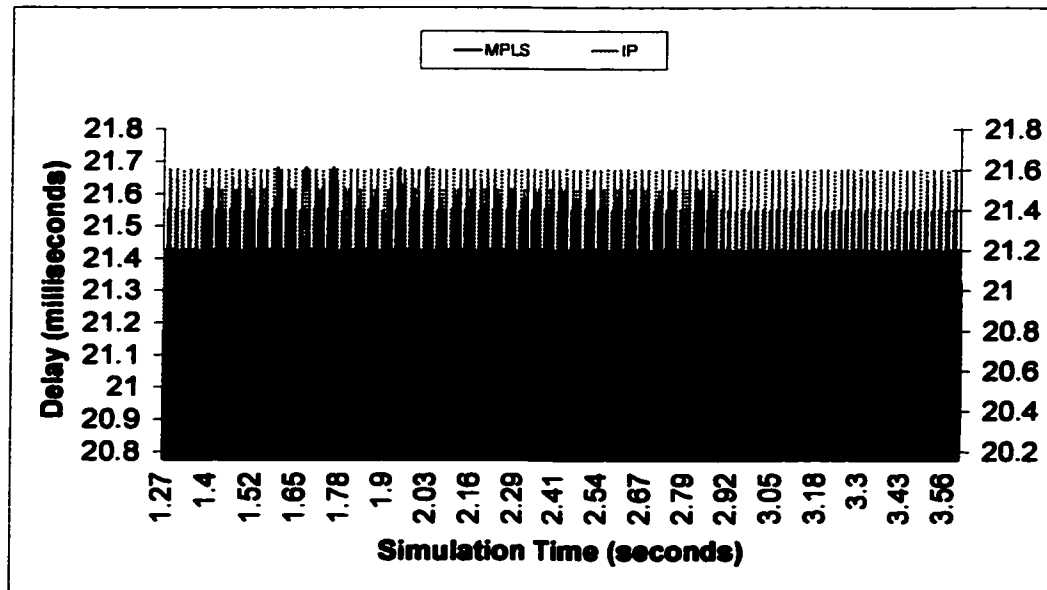


Figure 4.2: End-to-end Delay for Source 1 (VBR-500).

Vanilla MPLS graph shows that traffic of source 1 (VBR-500) has an end-to-end delay of 20.8 milliseconds at the beginning of transmission. This value increases as a result of the increase in the queuing delay when the traffic of the other sources join the network. Source 1 (VBR-500) maximum end-to-end delay reaches up to 21.5 milliseconds as seen in the graph.

In the IP network, traffic of source 1 (VBR-500) graph shows that it has an end-to-end delay of 20.8 milliseconds also at the beginning but this value increases upto 21.6 millisecond in many instances, as can be seen in the graph.

The frequency of having low end-to-end delay (20.8-21.2 milliseconds) in the MPLS network is higher than that in the IP network as seen in Figure 4.3. On the other hand, the frequency of having a little higher values for end-to-end delay (21.4-21.6 milliseconds) is higher in the IP network as seen in Figure 4.4. As a result the MPLS network has an

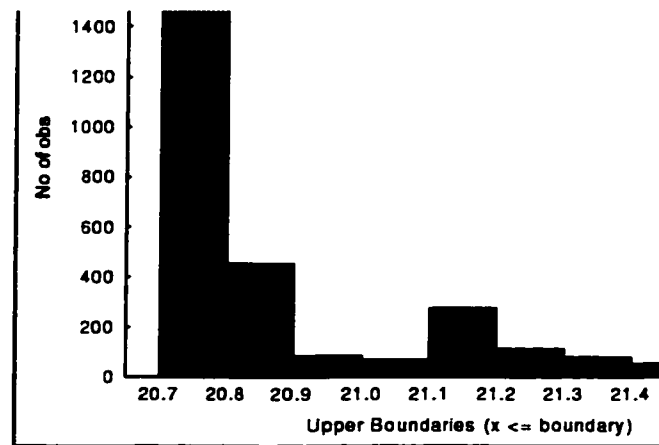


Figure 4.3: End-to-end Delay Frequency for Source 1 (VBR-500) in the MPLS Network.

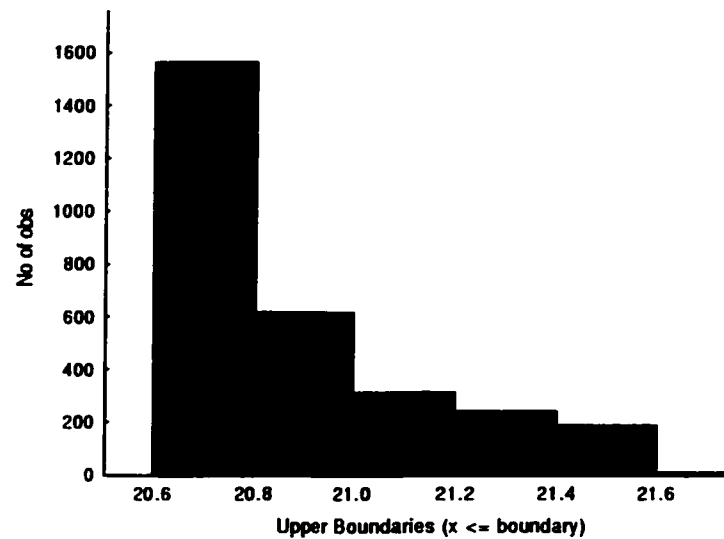


Figure 4.4: End-to-end Delay Frequency for Source 1 (VBR-500) in the IP Network.

average end-to-end delay lower than that in the IP network.

Source 2 (VBR-400) packets experience a transmission delay of 0.32 milliseconds at each router. Passing through three routers they suffer a transmission delay of 0.96 milliseconds. The propagation delay for the traffic is 30 milliseconds. Based on this, a packet of source 2 (VBR-400) has a minimum end-to-end delay of 30.96 plus queuing and switching delays. Figure 4.5 shows the end-to-end delay for the traffic of source 2 (VBR-400) in both networks.

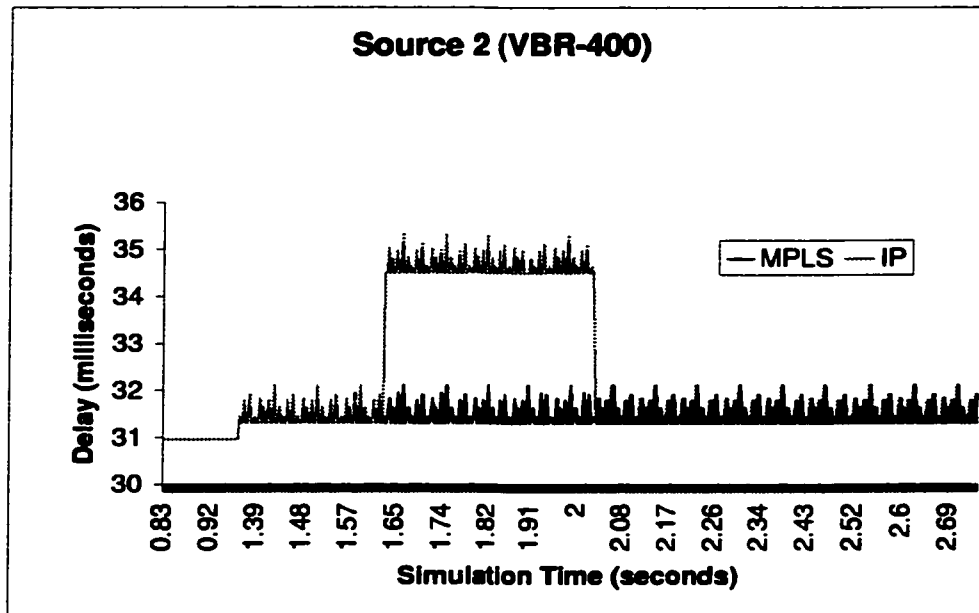


Figure 4.5: End-to-end Delay for Source 2 (VBR-400).

MPLS network graph indicates that traffic of source 2 (VBR-400) has an end-to-end delay values between 31 and 32 milliseconds according to the queuing delay it experiences.

In the IP network, source 2 (VBR-400) traffic has an end-to-end delay of 31 milliseconds that increases to 32 milliseconds when source 4 (CBR-1000) is turned on. This becomes

even worse when source 5 (CBR-900) becomes active at simulation time of 1.6 seconds where end-to-end delay reaches 35 milliseconds. Source 2 (VBR-400) end-to-end delay falls back and remains between 31-32 milliseconds when source 1 (VBR-500) goes off.

MPLS network graph shows that MPLS treats traffic of all sources the same way and does not favor any type of traffic. This is true since traffic of source 2 (VBR-400) end-to-end delay in this case is not seriously affected by the activity of the CBR sources.

A packet of source 3 (VBR-200) has a transmission delay of 0.16 milliseconds at each router. In the MPLS network, traffic of source 3 (VBR-200) is forced to take the route that contains routers 5 and 7 and ends at router 8. In the IP network, the situation is different and Source 3 (VBR-200) traffic travels the route that contains routers 4 and 6 and ends at router 8. In both cases packets of source 3 (VBR-200) suffer 0.48 milliseconds transmission delay and 30 milliseconds propagation delay. This sums to a minimum end-to-end delay of 30.48 milliseconds at best conditions. Figure 4.6 shows simulation results of the traffic of source 3 (VBR-200) end-to-end delay in both networks. MPLS network graph shows that traffic of source 3 (VBR-200) has an end-to-end delay that varies between 30.7 and 31.7 milliseconds. Source 3 (VBR-200) end-to-end delay drops to its minimum value of 30.48 only in the last second of the simulation when both CBR sources are off and only the VBR sources are active.

In the IP network, traffic of source 3 (VBR-200) has a fixed end-to-end delay of 30.48 milliseconds. Here, the path followed by the traffic of source 3 (VBR-200) is not used by any other source and hence, no competition for resources is taking place. This is why traffic of source 3 (VBR-200) experiences a minimum end-to-end delay.

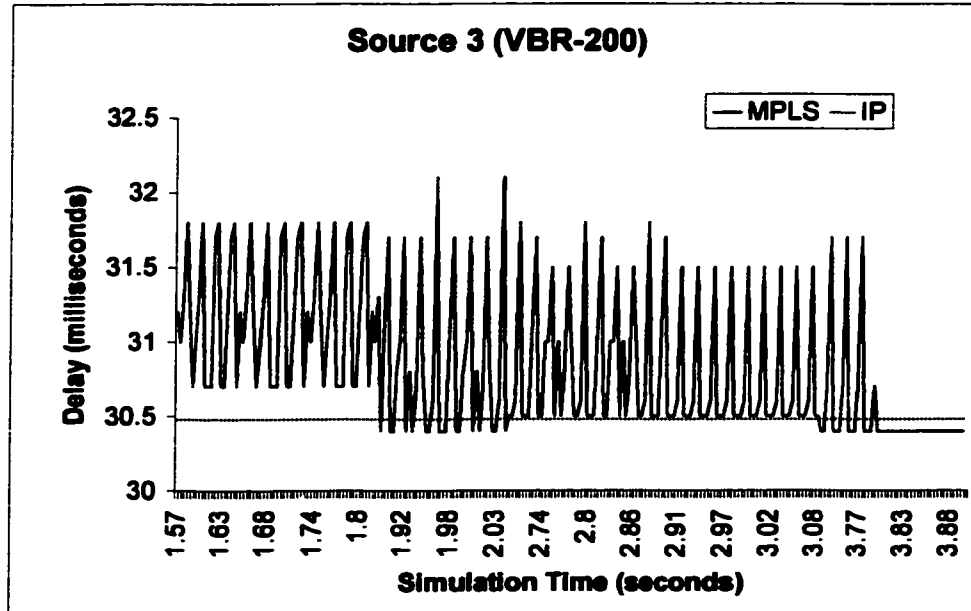


Figure 4.6: End-to-end Delay for Source 3 (VBR-200).

A packet of source 4 (CBR-1000) traffic has a transmission delay of 0.8 milliseconds per router. This produces 1.6 milliseconds transmission delay beside the 20 milliseconds propagation delay. The packet end-to-end delay for source 4 (CBR-1000) traffic is 21.6 milliseconds. Figure 4.7 shows simulation results for the traffic of source 4 (CBR-1000) end-to-end delay in both networks.

MPLS network graph shows that traffic of source 4 (CBR-1000) has an end-to-end delay between 21.6 milliseconds and 22.1 milliseconds most of the time. The moment source 2 (VBR-400) goes off, the maximum end-to-end delay of source 4 (CBR-1000) drops to 21.8 milliseconds and remains varying between 21.8 and 21.6 milliseconds which is the minimum end-to-end delay as seen in the graph.

As for the IP network, the traffic of source 4 (CBR-1000) has an end-to-end delay values between 21.6 and 21.7 milliseconds. When source 2 (VBR-400) becomes idle, source

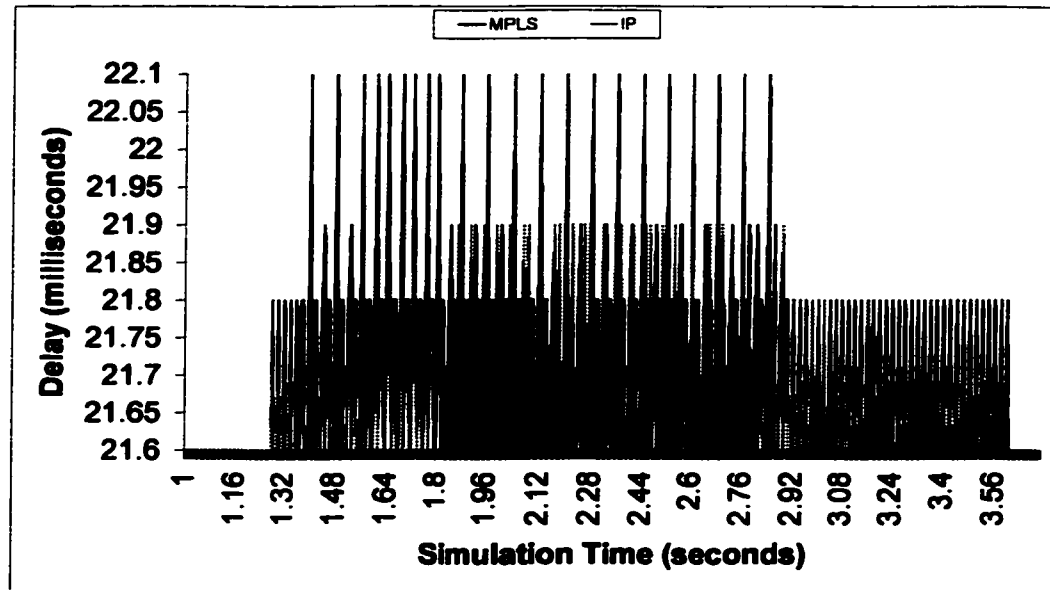


Figure 4.7: End-to-end Delay for Source 4 (CBR-1000).

4 (CBR-1000) end-to-end delay drops to 21.6 milliseconds and remains constant till the end of the experiment. Again, the IP network measures indicated a better handling of the traffic of CBR sources than that in the MPLS network.

A packet of source 5 (CBR-900) has a transmission delay of 0.72 at each router which results in 1.44 milliseconds of transmission delay and 20 milliseconds propagation delay. These values give a packet an end-to-end delay of 21.44 milliseconds. Figure 4.8 depicts the end-to-end delay experienced by the traffic of source 5 (CBR-900) in both networks. From the simulation results, traffic of source 5 (CBR-900) has almost the same end-to-end delay in both networks. The end-to-end delay varies between 21.44 and 21.55 milliseconds.

In general, MPLS networks reduce the end-to-end delay but provide no service guarantees. Further no differentiation in the treatment of flows have been observed. As a result in some cases IP performs better than MPLS when no route enforcement is engaged. The

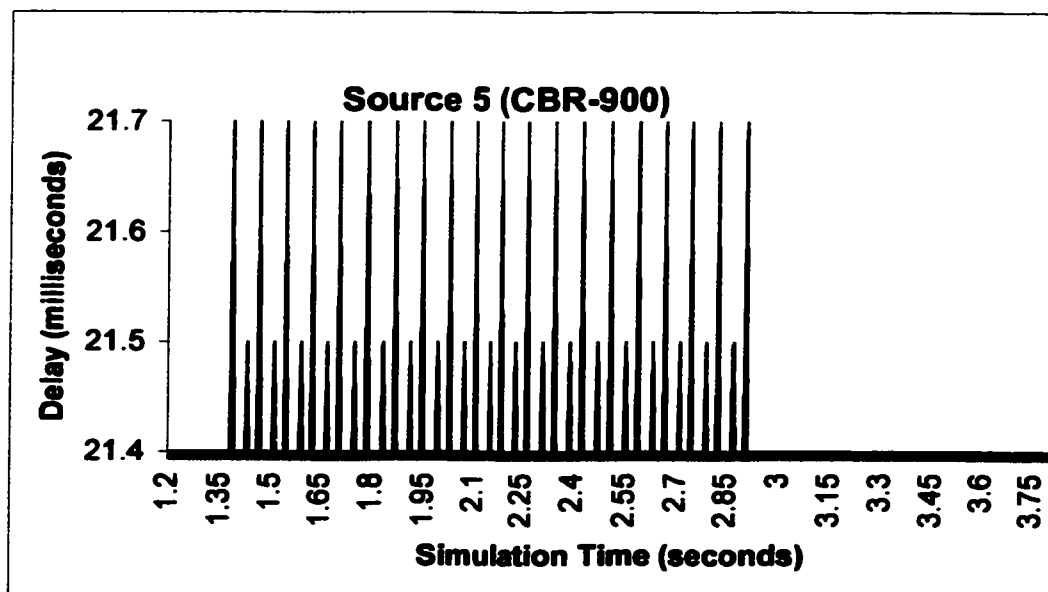


Figure 4.8: End-to-end Delay for Source 5 (CBR-900).

reduction in the end-to-end delay observed in the MPLS network is logical. It is justified by the fact that forwarding in the MPLS network core routers is only a matter of label swapping and then switching which consumes less time than the process of IP header check as performed in conventional IP forwarding.

The bandwidth consumed by the traffic of each source in the IP network is plotted in Figure 4.9. Figure 4.10 shows the bandwidth consumed by the traffic of each source in the MPLS network. In both figures the X-axis represents the simulation time in seconds and the Y-axis represents the bandwidth in bits/second. The bandwidth consumed in each of the two networks for the same source is almost the same. This means both networks are equally fair in allocating resources.

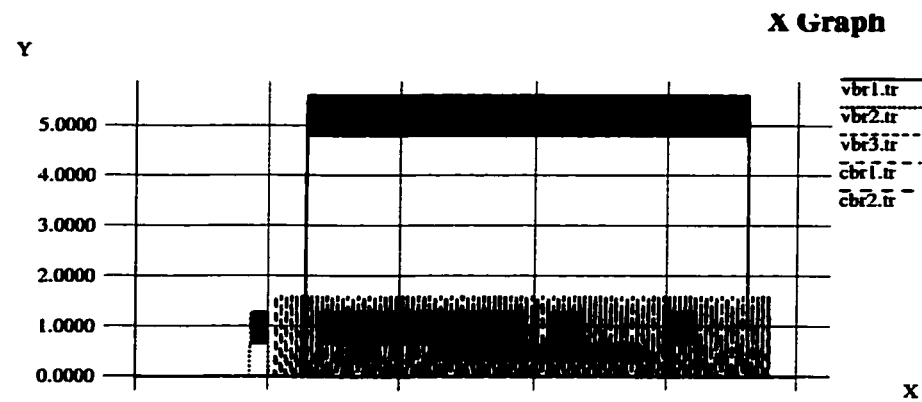


Figure 4.9: Bandwidth Utilization in the IP Network.

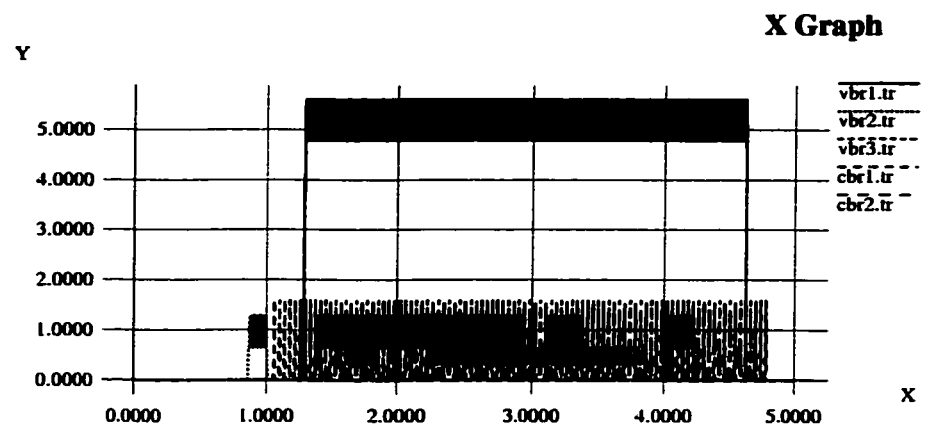


Figure 4.10: Bandwidth Utilization in the MPLS Network.

4.5 Diffserv MPLS versus Vanilla MPLS

In this experiment a Diffserv MPLS network is examined against a vanilla MPLS network. Diffserv is envisioned with two distinct classes of service: expedite forwarding (EF) and best effort (BE). This is implemented by having two logically independent queues. In Diffserv MPLS, schedulers enqueue an incoming packet according to its type in one of the two queues. Service is provided according to the traffic type, where EF traffic is served first. Measuring the end-to-end delay, delay variation and packet loss count in the two simulations enabled us to evaluate the effect of having differentiated service on MPLS network performance.

4.5.1 Experimental Model Description

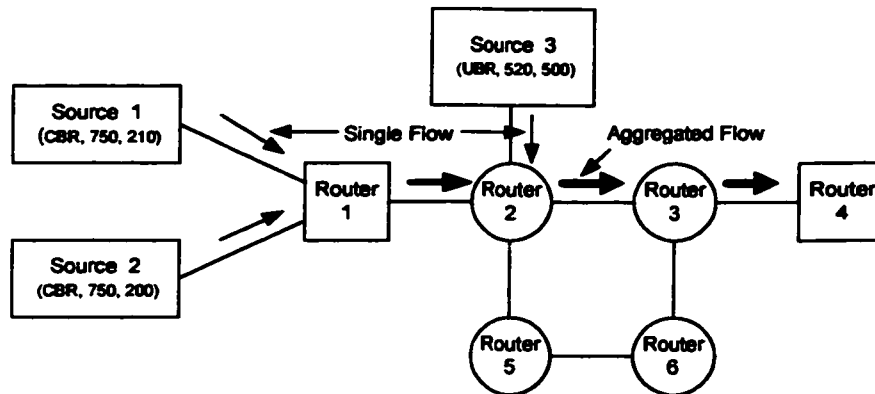


Figure 4.11: MPLS-Diffserv versus Vanilla MPLS Test Model.

The simulation network shown in Figure 4.11 consists of six routers; two (1, 4) are edge LSRs and the remaining (2, 3, 5, and 6) are core LSRs. Each edge LSR is connected to its neighbor core LSR using a link of 10 Mbps capacity and 0.1 millisecond propagation

delay. Core LSRs are connected to each other using links of 2.0 Mbps capacity and 1.0 millisecond propagation delay each. Source 1 and source 2 are attached to router 1 and source 3 is attached to router 2. Router 4 is the destination router for the three traffic sources. Router 2 is the bottleneck router and the link connecting this router to router 3 is the bottleneck link. Traffic characterization of the traffic sources used in all our experiments on MPLS is provided in Table 4.4.

Table 4.4: Traffic Sources for MPLS Simulation Study.

Source	Traffic Type	Rate (Packet Size (Bytes))	Starting-stopping Times	An Example
Source 1	Constant Bit Rate (CBR)	750 Kbps (210)	0.01-5.0	Voice (Noncompressed)
Source 2	Constant Bit Rate (CBR)	750 Kbps (200)	0.01-5.0	Voice (Noncompressed)
Source 3	Unspecified Bit Rate (UBR) 500 msec. ON and 500 msec. OFF	520 Kbps (500)	Randomly	Compressed Video (Low Quality)

4.5.2 Results and Discussion

Table 4.5 provides the transmission, propagation, and end-to-end (excluding queuing and processing (switching)) delays for the network traffic. The delays are mathematically calculated with different core link capacities.

Table 4.5: Vanilla Mpls versus Diffserv MPLS Experiment Calculated Delays.

Source	Link Capacity	Propagation Delay milliseconds	Transmission Delay milliseconds	End-to-end delay milliseconds
Source 1	1.7 Mbps	1.2	1.324	2.524
	2 Mbps	1.2	1.176	2.376
	10 Mbps	1.2	0.505	1.704
Source 2	1.7 Mbps	1.2	1.261	2.461
	2 Mbps	1.2	1.12	2.32
	10 Mbps	1.2	0.48	1.68
Source 3	1.7 Mbps	1.1	3.152	4.252
	2 Mbps	1.1	2.4	3.5
	10 Mbps	1.1	0.8	1.9

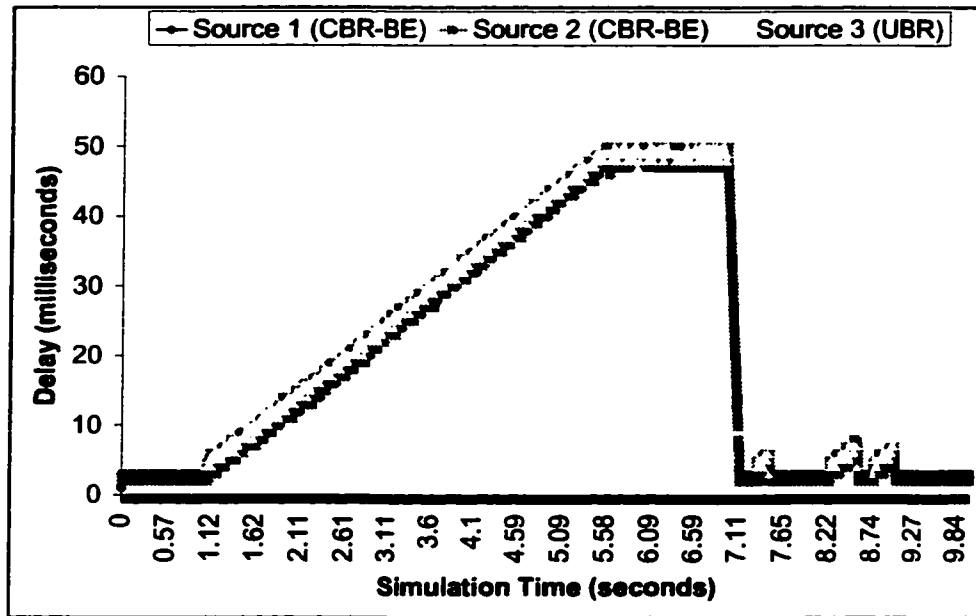


Figure 4.12: Vanilla MPLS: End-to-end Delay for the Network Traffic.

The end-to-end delay for the network traffic measured from the vanilla MPLS simulation is plotted in Figure 4.12. Till the end of the first second, the end-to-end delay of the traffic of both CBR sources remains at the minimum values (2.37 and 2.32 milliseconds). During the sessions of source 3 (UBR), traffic of both sources end-to-end delay increases as the case at 1.07 seconds when source 3 (UBR) becomes active. The maximum end-to-end delay value recorded for a packet of source 1 (CBR-210) is 49 milliseconds and for a packet of source 2 (CBR-200) is 48 milliseconds. These values are recorded at 5.5 seconds. The end-to-end delay of the traffic of both sources remains at these values till the end of this session of source 3 (UBR). When source 3 (UBR) goes off at the end of the seventh second, the end-to-end delay of the traffic of the two CBR sources drops to the minimum values. Their end-to-end delay is affected during source 3 (UBR) active sessions, for example

at 8.36 and 8.85 seconds, as can be seen in Figure 4.12. During source 3 (UBR) active sessions, the number of packets to be served at router 2 exceeds the capacity of the link connecting router 2 and 3 and hence packets are queued waiting for transmission. The resulting queuing delay increases the end-to-end delay of the traffic of the three sources.

The end-to-end delay of the traffic of source 3 (UBR) starts with a value of 4.0 milliseconds and increases rapidly until it reaches 49 milliseconds at 5.5 seconds. The end-to-end delay of the traffic of source 3 (UBR) remains unchanged till the end of this source session at the end of the seventh second. In each of the active periods, the end-to-end delay starts with a minimum value and increases over time as the queue grows up.

In this experiment, we have recorded a loss of five packets from source 1 (CBR-BE) and nine packets from source 2 (CBR-EF) and five packets from source 3 (UBR). Results measured in this simulation show that all the three sources are treated the same way and received best effort service.

Figure 4.13 depicts the end-to-end delay for the traffic of the network sources when Diffserv is incorporated with MPLS. Traffic of source 2 (CBR-EF) shows that it suffers the least end-to-end delay which remains between 2.32 to 3.0 milliseconds. From the graph, during source 3 (UBR) active sessions, for example at 0.22, 5.5, and 8.85 seconds, source 2 (CBR-EF) packets end-to-end delay goes up to 4 milliseconds. This means that traffic of source 2 (CBR-EF) faces a queuing delay of 1.0-1.68 millisecond during this period. This indicates that the end-to-end of the traffic of source 2 (CBR-EF) is not significantly affected by the introduction of the traffic of source 3 (UBR).

On the other hand, the end-to-end delay of the traffic of source 1 (CBR-BE) is sig-

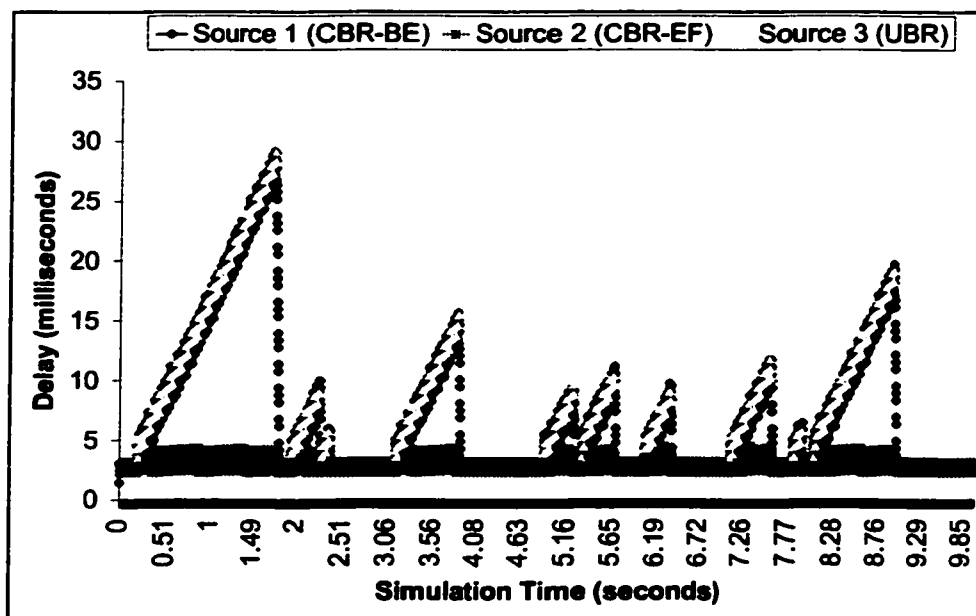


Figure 4.13: Diffserv MPLS: End-to-end Delay for the Network Traffic.

nificantly affected by the introduction of source 3 (UBR). Traffic of source 1 (CBR-BE) end-to-end delay increases during source 3 (UBR) active periods and decreases during source 3 (UBR) inactive intervals. This can be seen in source 1 (CBR-BE) graph at 0.22 seconds where the end-to-end delay of a packet jumps to 29 milliseconds. The same is true at 3.1 seconds where CBR-BE packet end-to-end delay goes to 15.5 milliseconds and also at 8.9 seconds where its delay reaches 20 milliseconds. These are the periods during which source 3 (UBR) is active (see the plot of the traffic of source 3 (UBR)). When source 3 (UBR) goes off, traffic of source 1 (CBR-BE) end-to-end delay drops to a very low value that is about 2.4 milliseconds as the case at 2.33, 3.8, and 9.0 seconds in Figure 4.13.

When source 3 (UBR) starts sending data, its packets have an end-to-end delay of 3.5 milliseconds. The packet end-to-end delay increases, as the source continues sending

traffic, till 29 milliseconds. At 1.7 seconds, source 3 (UBR) goes off. When it starts sending data again at 2.3 seconds, its packet end-to-end delay begins with 4.0 milliseconds. Again the end-to-end delay increases with time. The same phenomena is repeated at 4.1, 5.3, 8.0 seconds and so on as can be observed in the graph. The increase in the end-to-end delay is due to the increases in the queuing delay.

When analyzing the simulation results, it has been mentioned that: increase in the end-to-end delay is due to the queuing delay. To prove what stated here, we have measured the queuing delay of the traffic of the three sources at router 2. Traffic of the network queuing delay is depicted in Figure 4.13. Reading the queuing delay of a packet of any source and

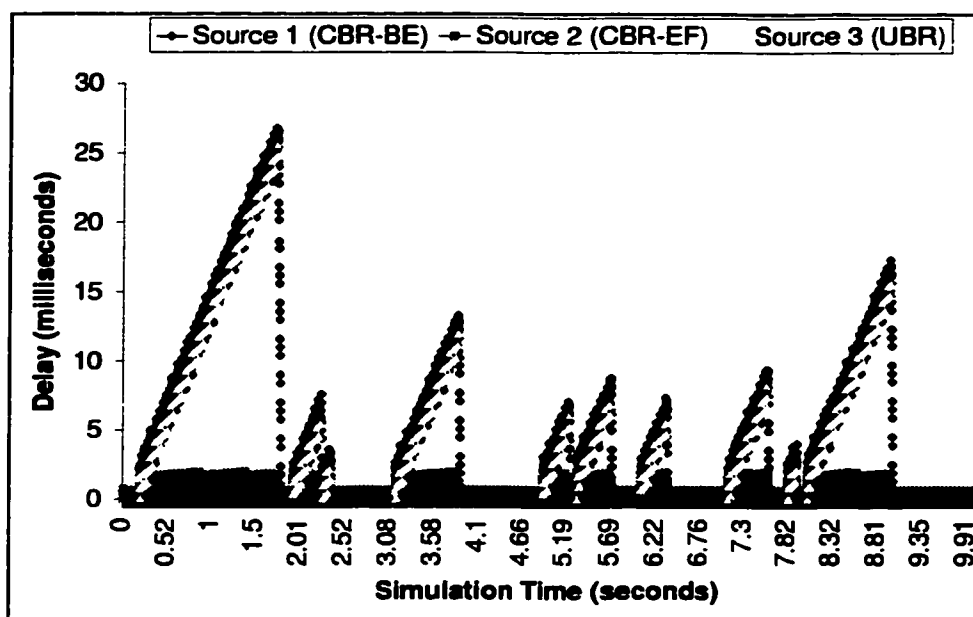


Figure 4.14: Diffserv MPLS: Queuing Delay for the Network Traffic at Router 2.

the corresponding end-to-end delay, we recognize the direct relationship between them. This can be seen by comparing the queuing delay of each of the two CBR sources from Figure 4.14 with the corresponding end-to-end delay in Figure 4.13. The same is true for

source 3 (UBR).

Traffic of source 2 (CBR-EF) suffers the minimum queuing delay. At the same time source 1 (CBR-BE) and source 3 (UBR) traffics suffer more queuing delay. This is because, source 2 (CBR-EF) packets have higher priority in service than the other two sources packets. This is also justified by considering that source 2 (CBR-EF) is the only source which is sending EF traffic and hence there is no high competition for the EF-service resources. It is also true that source 2 (CBR-EF) is sending its traffic in a constant rate which means there is no variation in the amount of the traffic arriving at router 2.

Traffic of source 1 (CBR-BE) queuing delay at router 2 increases with the increase in the traffic at the router. This is because packets arriving at the router when source 3 (UBR) injects its packets are of BE-service type and hence they will be added to the BE-service queue. The same is correct about the traffic of source 3 (UBR). The logical result of the competition for the BE-service resources is that, traffic of both suffer more queuing delay.

Delay variation for the three sources is also measured. Figure 4.15 shows delay variation of the traffic of the three sources. In the figure it is clearly seen that, traffic of

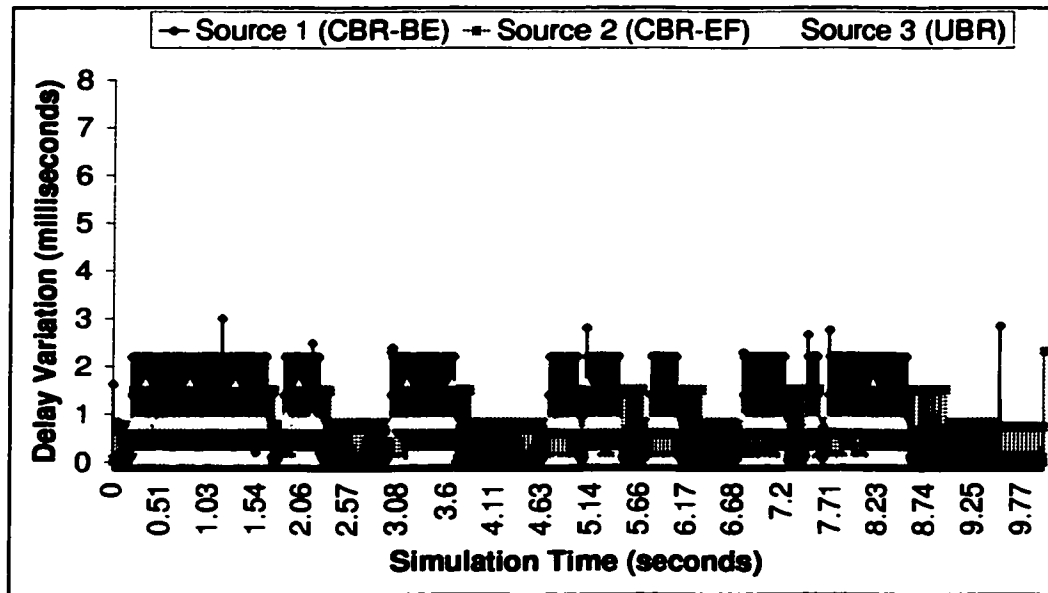


Figure 4.15: Diffserv MPLS: Delay Variation for the Network Traffic.

source 2 (CBR-EF) suffers the least delay variation while source 1 (CBR-BE) and source 3 (UBR) delay variations are affected by the competition for the resources. It is natural for source 2 (CBR-EF) traffic to have a very small variation since it has small and not much varying queuing delay. It is also natural for the traffic of the other two sources to have more variation since they experience variable queuing delay.

The bandwidth consumed by each of the three traffic sources for the whole simulation time is shown in Figure 4.16 for the vanilla MPLS network and Figure 4.17 for the Diffserv MPLS network.

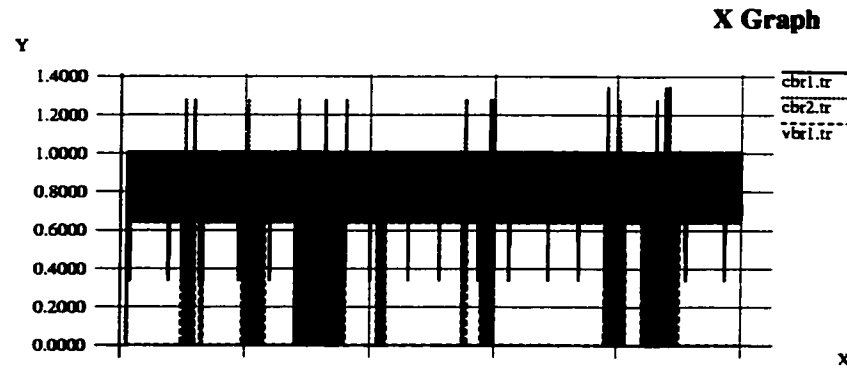


Figure 4.16: Vanilla MPLS: Bandwidth Consumed by the Three Traffic Sources.

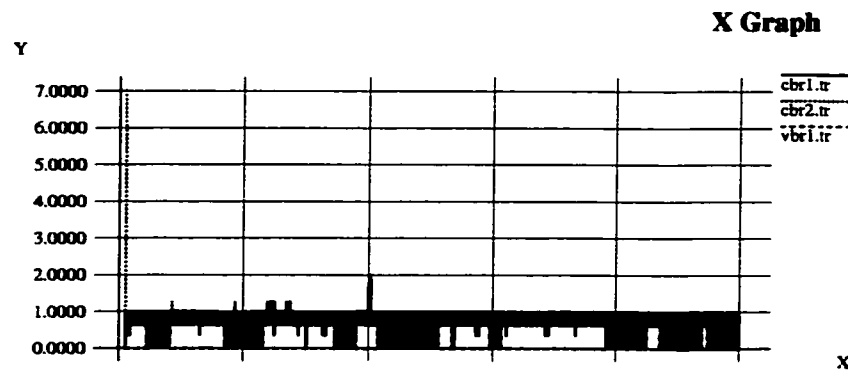


Figure 4.17: Diffserv MPLS: Bandwidth Consumed by the Three Traffic Source.

By studying the two graphs in detail, we conclude that in the vanilla MPLS network resources were fairly distributed between the traffic sources. In the Diffserv MPLS network resources were allocated according to the weights specified for each type of traffic. This means Diffserv MPLS network is fair in the sense that:

1. Limits of each source share are not exceeded.
2. Traffic sources are penalized the same way.
3. No source starves while others are using more bandwidth.

In this set of experiments, no packet loss is recorded for any of the three sources. The number of packets received from source 3 (UBR) at the destination increases significantly.

To study the effect of increasing and decreasing the network links capacity on the network performance we have repeated the same experiment two times. The first time, with core links capacity of 10 Mbps and the second time, with core links capacity of 1.7 Mbps.

Higher Link Capacities Results

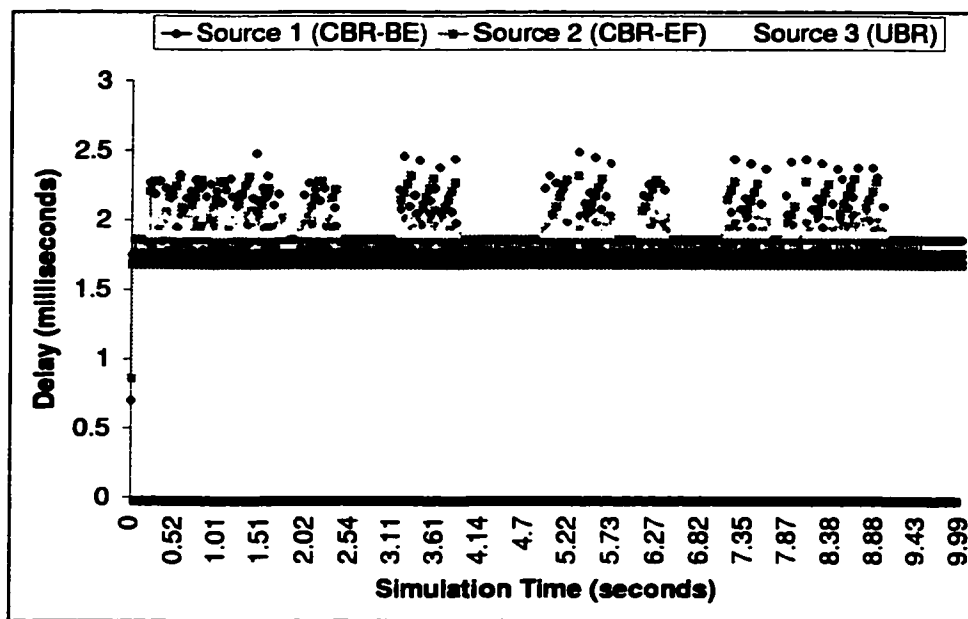


Figure 4.18: End-to-end Delay for the Network Traffic 10Mbps Core Links.

Simulation results are shown in Figure 4.18. With higher core links capacity, traffic of all of the three sources experienced reduced end-to-end delay. This is logical since reducing the transmission time leads to reducing the queuing delay and consequently reduces the

end-to-end delay. As a conclusion enriching the network resources such as by increasing link capacity improves the network performance.

Lower Link Capacities Results

When lower capacities are used for the core links, the network performance is expected to deteriorate as a result of higher competition for resources. At the same time, reducing core links capacity enables us to examine the effect of queuing delay on the end-to-end delay. Figure 4.19 shows the simulation end-to-end delay. As expected, results have shown that the end-to-end delay has a significant increase due to the reduction in the core links capacity. This resulted from the increase in the transmission delay which also results in increased queuing delay.

Traffic of the network sources queuing delay is plotted together in Figure 4.20. From the graphs, the increase in the end-to-end delay as a result of reducing the core links capacity is clear. Inspecting the graphs, proves once again the known fact about the relation between the queuing and end-to-end delays. In the instances where the queuing delay is small the end-to-end delay drops, and the increase in the queuing delay increases the end-to-end delay. The end-to-end delay increases and decreases linearly with the network link capacity. Increasing a link capacity reduces the transmission, queuing, and end-to-end delays. The opposite is also correct reducing a link capacity increases these delays.

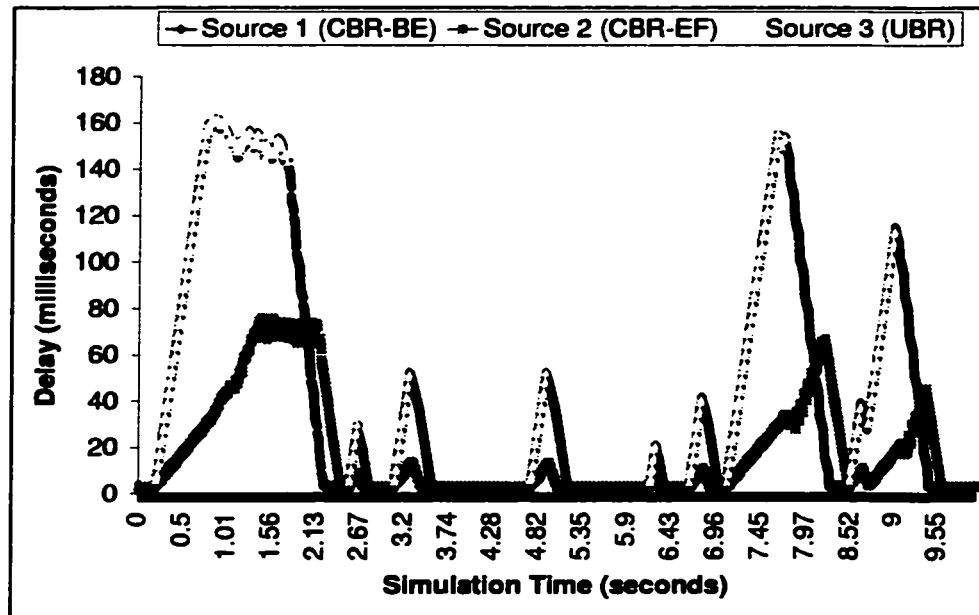


Figure 4.19: The Network Traffic end-to-end Delay 1.7 Mbps Core Links.

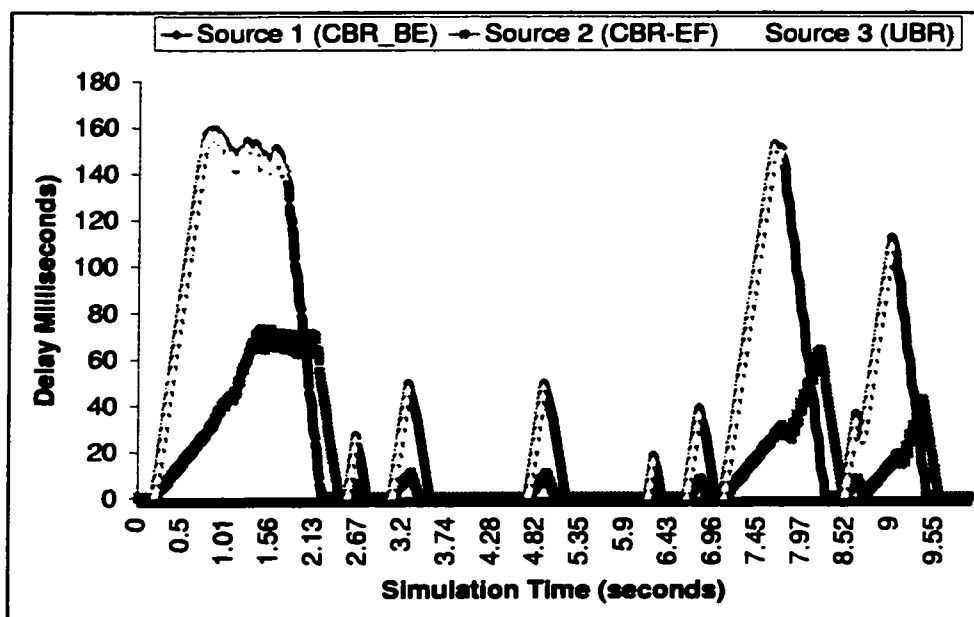


Figure 4.20: The Network Traffic end-to-end Delay 1.7 Mbps Core Links.

Chapter 5

Simulation and Results (2): Fault Tolerance in MPLS Networks

In this Chapter traffic engineering capabilities of MPLS such as rerouting of traffic in case of link failure is examined. In Section 6.1, we examine MPLS behavior in case of a link failure in the presence of a single alternative LSP. The objective here is to find out how fast MPLS redirects traffic and what the effect is of that on the traffic. Next to that MPLS performance with a link failure in the presence of variable loads is also tested. Fault tolerance in MPLS networks in the sense of the ability to guarantee almost the same level of service using alternative LSP is examined in the last set of experiments. The way MPLS picks the alternative LSP in the presence of multiple alternative LSPs is reported and the effect of that on the network performance is observed.

5.1 Link Failure in Diffserv MPLS

In this experiment, we cause the link connecting routers 2 and 3 (thereafter is called the link) of the network shown in Figure 5.1 to fail during the fourth second and resume to operation after that. The link failure causes traffic of source 1 (CBR-BE) to travel on link 6-7, following the path shown in the figure with thick arrows. As a result, traffic of both sources share link 6-7 which becomes the bottleneck link. This simulation is repeated three times. First time, with moderate resources using links of 2 Mbps capacity. Second, with extra resources using links of 10 Mbps capacity. Finally, with limited resources using links of 1.5 Mbps capacity. The objective is to observe the effect of the bandwidth capacity on the performance and to study the relation between the link capacity and the end-to-end delay.

In this set of experiments, the Diffserv MPLS network behavior during link failure is examined. The network performance is to be evaluated when a traffic of a certain type is merged with the traffic of a different type.

5.1.1 Experimental Model Description

The simulation network of this experiment is built using eight routers connected as shown. Routers (1, 4, 5, and 8) are edge routers (E-LSRs) and the other four routers (2, 3, 6, and 7) are core routers. Edge LSRs are connected to core LSRs using links of 10 Mbps capacity and 0.1 millisecond propagation each. The experiment is repeated for different core links capacities.

In this experiment only the two CBR sources of Table 4.4 are used. Source 1 is

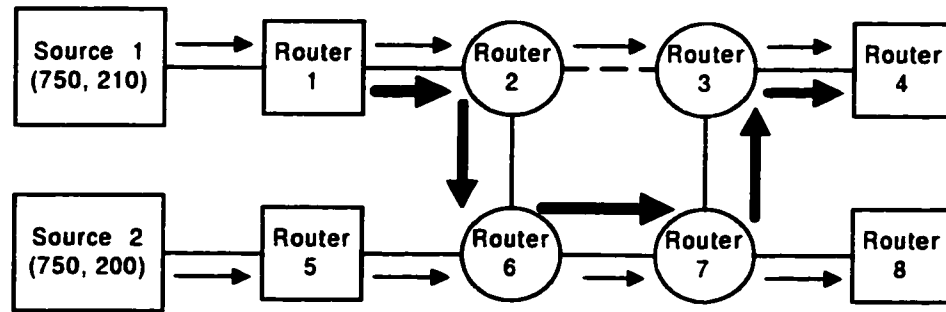


Figure 5.1: Simulated Network for Link Failure.

attached to router 1 and its destination is at router 4. Source 2 is attached to router 5 and its destination is at router 8. The thin arrows show both sources traffic paths under normal conditions.

5.1.2 Effect of Link Failure when Mapping Source 1 traffic to BE

Table 5.1 provides the mathematically calculated transmission, propagation, and end-to-end delays for the two CBR sources traffic under different link capacities. Note that the end-to-end delay values provided in the table does not contain queuing and processing (switching) delays.

The end-to-end delays experienced by the traffic of source 1 (CBR-BE) and source 2 (CBR-EF) are plotted in Figure 5.2 and Figure 5.3. With core links of 2 Mbps capacity, traffic of source 1 (CBR-BE) has an end-to-end delay of 2.376 milliseconds from the beginning of the simulation till the link fails. During the link failure period, traffic of source 1 (CBR-BE) end-to-end delay increases to values between 6.1 and 6.7 milliseconds.

Table 5.1: Mathematically Calculated Delays for the network with the Link.

		Default LSP			Alternative LSP		
Source	C	T_{prop}	T_{trans}	T_{e2e}	T_{prop}	T_{trans}	T_{e2e}
Source 1	1.5	1.2	1.456	2.656	3.2	3.696	6.896
	2	1.2	1.176	2.376	3.2	2.856	6.056
	10	1.2	0.48	1.68	3.2	0.8	4.0
Source 2	1.5	1.2	1.386	2.586	1.2	1.386	2.586
	2	1.2	1.12	2.32	1.2	1.12	2.32
	10	1.2	0.48	1.68	1.2	0.48	1.68
T_{prop} = Propagation Delay in msec. T_{trans} = Transmission Delay in msec. T_{e2e} = End-to-end Delay in msec. LC = Link Capacity in Mbps.							

This means, it has a maximum queuing delay of 0.65 milliseconds. Traffic of source 1 (CBR-BE) end-to-end delay returns to its minimum value of 2.376 milliseconds after the link is restored. Traffic of source 2 (CBR-EF) has an end-to-end delay of 2.32 milliseconds during normal operation. During the link failure, traffic of source 2 (CBR-EF) end-to-end delay increases by 0.68 millisecond corresponding to a maximum queuing delay.

With core links capacity of 10 Mbps, traffic of source 1 (CBR-BE) has an end-to-end delay of 1.7 milliseconds before the link failure. During the link failure, traffic of source 1 (CBR-BE) end-to-end delay is between 4 and 4.1 milliseconds. This means, its packet queuing delay is 0.1 milliseconds at most. Traffic of source 2 (CBR-EF) has an end-to-end delay of 1.68 milliseconds before the link failure. During the link failure, traffic of source 2 (CBR-EF) end-to-end delay increases to 1.8 milliseconds. This means, its packet queuing delay is 0.12 millisecond at maximum.

Using 1.5 Mbps core links, traffic of source 1 (CBR-BE) has an end-to-end delay of 2.66 milliseconds before the link failure. During its link failure, traffic of source 1 (CBR-BE) end-to-end delay jumps to values between 6.9 and 7.8 milliseconds. In this case, it

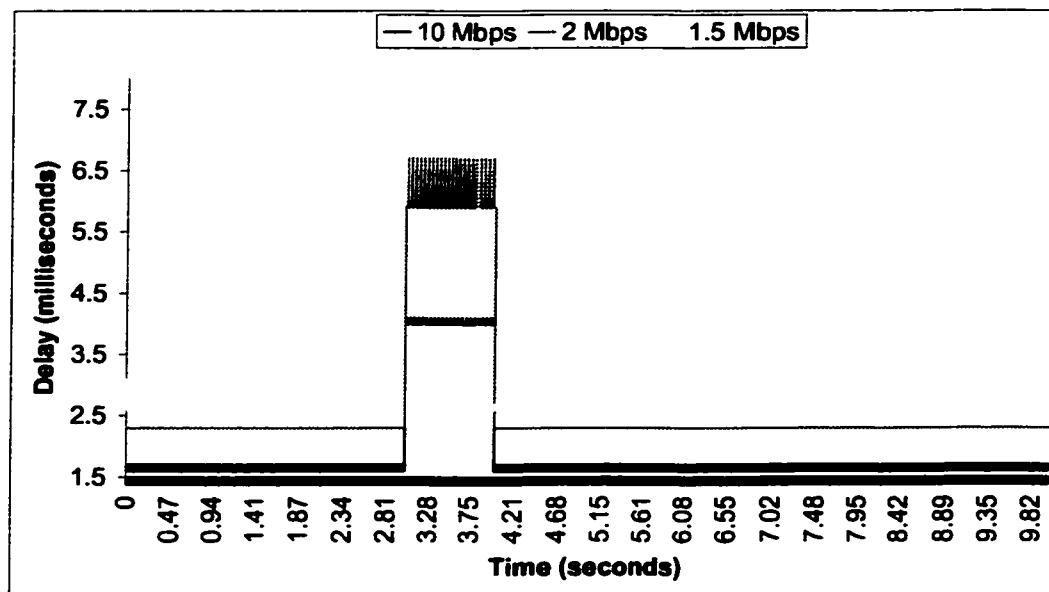


Figure 5.2: End-to-end Delay for Source 1 Traffic which is mapped to BE when Link Failure occurs.

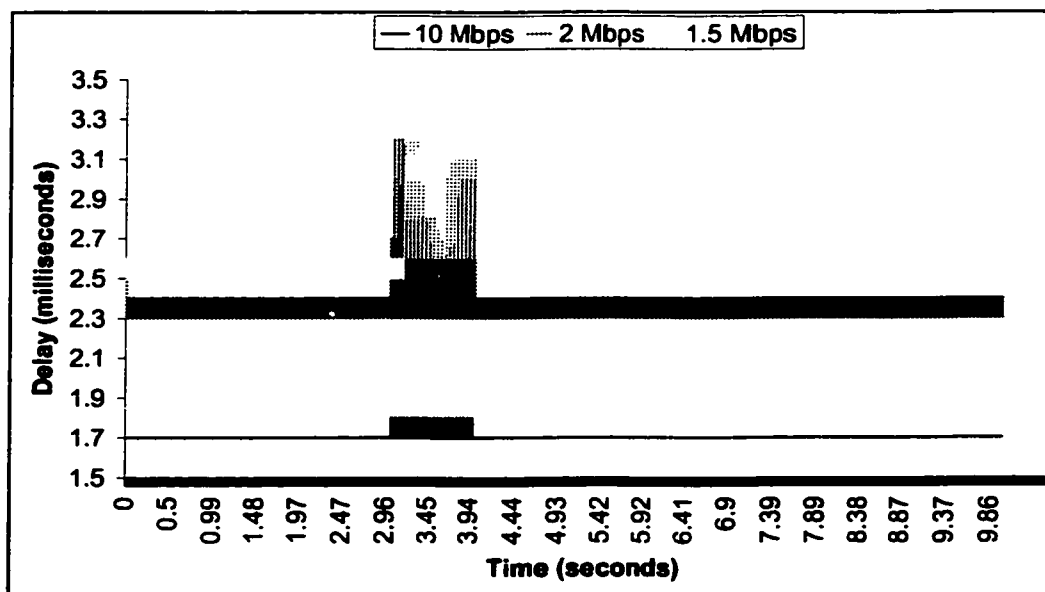


Figure 5.3: End-to-end Delay for Source 2 Traffic with Source 1 Traffic mapped to BE when Link Failure occurs.

has a queuing delay of 0.9 milliseconds at most. Traffic of source 2 (CBR-EF) end-to-end delay is 2.586 milliseconds before the link failure. During the link failure, traffic of source 2 (CBR-EF) end-to-end delay increases to values between 2.7 and 3.7 milliseconds. This means, it has a queuing delay of 0.1 to 1.1 milliseconds.

5.1.3 Effect of Link Failure when Mapping Source 1 Traffic to EF

We made the two CBR sources to exchange their class of service. Source 1 traffic is now mapped into EF while source 2 traffic is mapped into BE. Then, the same experiment is repeated under similar conditions. We mean to find out who MPLS handles traffic in this case and whether the EF traffic gets the same level of service.

The end-to-end delay experienced by the network traffic is plotted in Figures 5.4 and 5.5. With 2 Mbps core links, traffic of source 1 (CBR-EF) has an end-to-end delay of 2.3 milliseconds before its link failure. During the link failure, traffic of source 1 (CBR-EF) end-to-end delay is between 5.9 and 6.7 milliseconds. This means, it has a queuing delay of 0.8 milliseconds at maximum. Traffic of source 2 (CBR-BE) end-to-end delay jumps from 2.3 to 3.2 millisecond during the link failure. This means traffic of source 2 (CBR-BE) suffers queuing delay of 0.82 milliseconds.

Using 10 Mbps core links, traffic of source 1 (CBR-EF) has an end-to-end delay of 1.68 milliseconds before its link failure. During its link failure, traffic of source 1 (CBR-EF) end-to-end delay increases to values between 4 and 4.1 milliseconds. This means its has a queuing delay of 0.1 milliseconds at most. Traffic of source 2 (CBR-BE) has an end-to-end

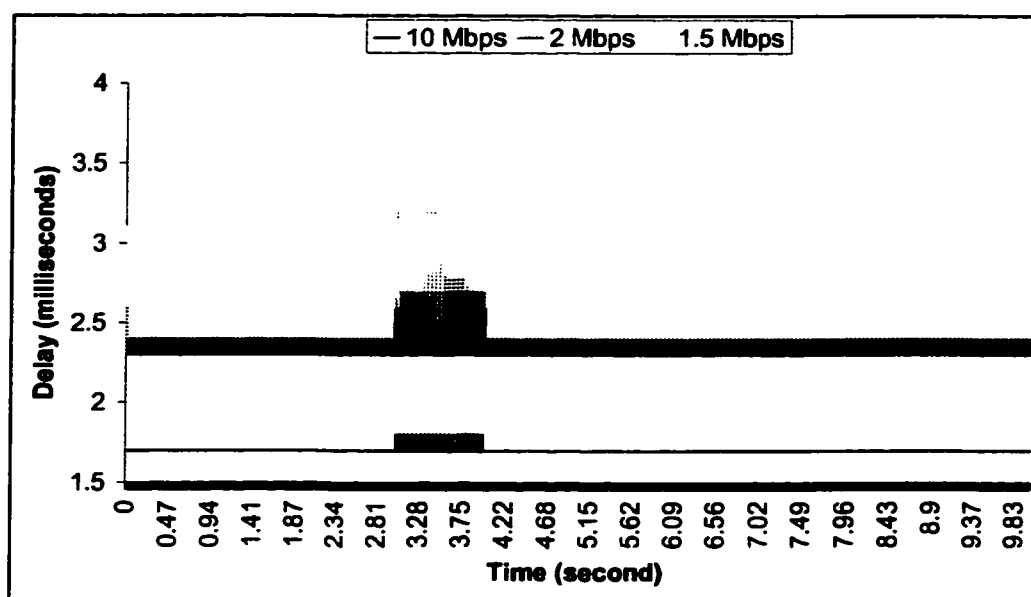


Figure 5.4: End-to-end Delay for Source 1 Traffic which is mapped to EF when Link Failure occurs.

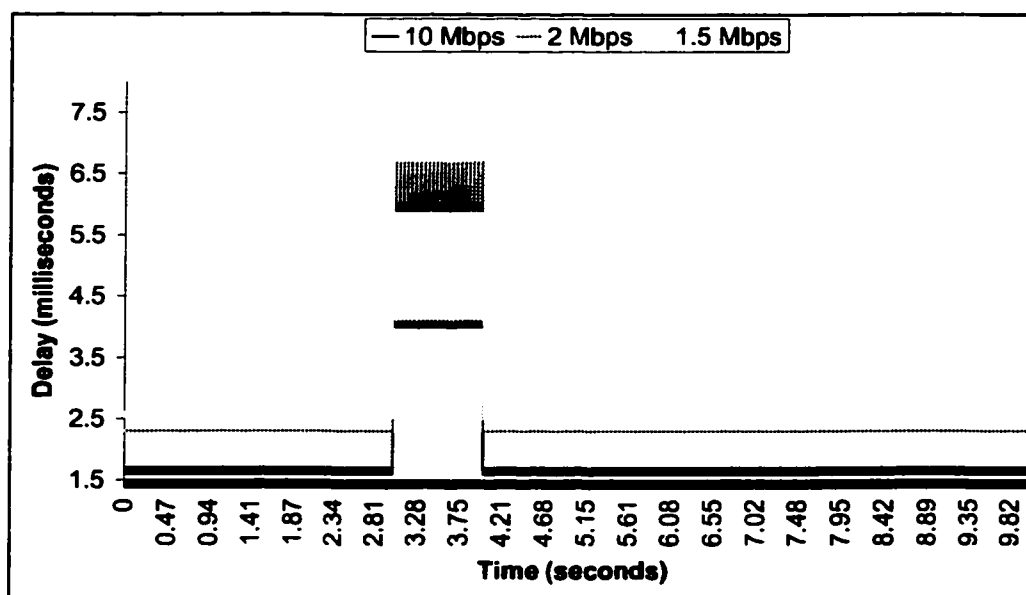


Figure 5.5: End-to-end Delay for Source 2 Traffic with Source 1 Traffic mapped to EF when Link Failure occurs.

delay of 1.7 milliseconds. Traffic of source 2 (CBR-BE) end-to-end delay increases during the link failure to 1.8 milliseconds. This means, traffic of source 2 (CBR-BE) suffers queuing delay of 0.1 milliseconds.

With core links capacity of 1.5 Mbps, traffic of source 1 (CBR-EF) has an end-to-end delay of 2.6 milliseconds before its link failure. During its link failure, traffic of source 1 (CBR-EF) end-to-end delay increases to values between 6.8 and 7.8 millisecond. This means, it has a queuing delay of 0.9 milliseconds at most. Traffic of source 2 (CBR-BE) has an end-to-end delay of 2.6 milliseconds before the link failure. During the link failure, traffic of source 2 (CBR-BE) end-to-end delay increases to 3.6 and at some instances 3.7 millisecond. This indicates that traffic of source 2 (CBR-BE) suffers queuing delay of 1.0 to 1.1 milliseconds.

The main observation in this experiment is that the end-to-end delay of the traffic of both sources in both cases is affected by the link failure. Traffic of the CBR-EF source end-to-end delay is slightly lower than that of the traffic of the CBR-BE source. This is because the traffic of the CBR-EF source has better service and hence suffers lower queuing delay than the traffic of the CBR-BE source. As seen in the results presented, the difference in delay measures is slight and not very distinct. The reasons behind this is that although the CBR-EF source traffic has higher priority in service the amount of resources reserved for the EF traffic is less than that for the BE traffic which makes the BE traffic posses almost the same level of service when its amount in the network is small as the case in this experiment. This is more observable in the case of moderate and high resources. Recognizing the effect of the link on the rerouting of the traffic it is seen that

the EF is affected after some time from the link failure while the the BE traffic is effected on the spot which confirms that the EF traffic is preferred in handling. This is more clear in the case of low resources (1.5 Mbps core links). This observation leads us to conclude that EF is more appropriate to be used in networks with high probability of link failure. To have a more close look to the situation in a more realistic scenario we work in a more loaded network that will be described in the following section.

5.2 Link Failure in Diffserv MPLS with Variable Loads

In this set of experiments the MPLS performance during link failure in the presence of variable traffic load as well as more different traffic types is examined. This experiment main objective is to find out how the presence of increased amount of BE traffic affects the service provided to EF traffic and whether the guaranteed level of service is still provided to the EF traffic. To achieve our objectives, a link is caused to fail for one second and the MPLS behavior is observed and recorded in the form of end-to-end and queuing delay. Delay variation and packet loss is also measured. Results are presented and analyzed.

5.2.1 Experimental Model Description

The simulation network of this experiments shown in Figure 5.6 consists of two source LSRs, a destination LSR, and four core LSRs. The two CBR traffic sources are attached to router 1 and the UBR traffic source is attached to router 5. Destinations of the traffic

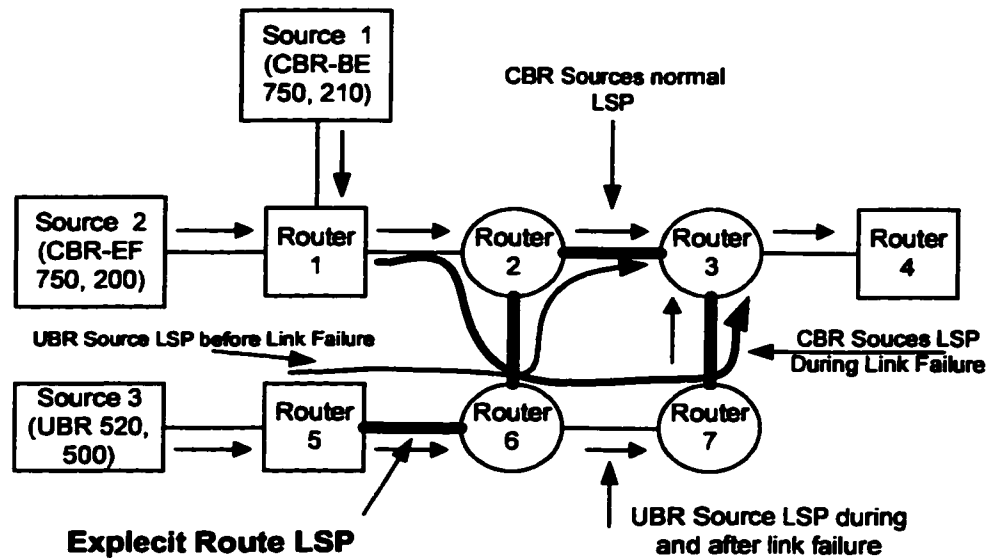


Figure 5.6: Link Failure with Variable Loads Simulation Network.

of the three sources are attached to the destination router (router 4). Each of the source LSRs and the destination LSR is connected to its neighbor core LSR using a link of 10 Mbps capacity and 0.1 millisecond propagation delay. Core LSRs are connected to each other using links of 2.0 Mbps capacity and 1.0 millisecond propagation delay each.

During normal operation and after the link restoration, traffic of the two CBR sources follow the LSP that starts at router 1 and contains routers 2, 3, and 4 to its destination. Traffic of source 3 (UBR) travels in the network following the explicit route LSP from router 5 to router 3 via router 2. At router 3, traffic of source 3 (UBR) departs from the ER-LSP and continues its way to its destination at router 4. In the first three seconds of the simulation, traffic of source 3 (UBR) shares the link and link 3-4 with the traffic of the CBR sources.

During the link failure, traffic of the CBR sources uses its alternative LSP which

contains routers 1, 2, 6, 7, 3, and 4. At the same time, traffic of source 3 (UBR) also uses its alternative LSP which contains routers 5, 6, 7, 3, and 4. In this case, traffic of source 3 (UBR) shares links 6-7, 7-3, and 3-4 with the traffic of the CBR sources. After the link restoration, traffic of source 3 (UBR) sticks to this LSP and does not return to its default LSP.

5.2.2 Results and Discussion

The transmission, propagation, and end-to-end delays for the network traffic with core link capacities of 1.7 and 2.0 Mbps is presented in Table 5.2. Values in this table are mathematically calculated and the end-to-end delay does not include and processing (switching) delays. Calculated values are also valid for the EF traffic sources case that to be presented in the last section of this experiment.

Table 5.2: Link Failure with Variable Loads Experiment Calculated Delays.

Source	<i>LC</i>	Default LSP			Alternative LSP		
		T_{prop}	T_{trans}	T_{e2e}	T_{prop}	T_{trans}	T_{e2e}
Source 1	1.7	1.2	1.324	2.524	3.2	3.3	6.5
	2	1.2	1.176	2.376	3.2	2.856	6.056
Source 2	1.7	1.2	1.261	2.461	3.2	3.143	6.343
	2	1.2	1.12	2.32	3.2	2.72	5.92
Source 3	1.7	2.2	5.502	7.702	2.2	5.502	7.702
	2	2.2	4.8	7.0	2.2	4.8	7.0
T_{prop} = Propagation Delay in msec.				T_{trans} = Transmission Delay in msec.			
T_{e2e} = End-to-end Delay in msec.				LC = Link Capacity in Mbps.			

The simulation results with core link capacity of 2 Mbps are shown in Figure 5.7. Traffic of source 1 (CBR-BE) has an end-to-end delay of 2.4 to 3.1 milliseconds. These values are recorded before the link failure and having UBR source off. Notice that, these

values are in accordance with the mathematically calculated value (2.376 milliseconds) since the lower limit (2.4 milliseconds) is recorded when there is no queuing delay and the upper limit (3.1 milliseconds) is recorded after the queue is accumulated and hence the queuing delay increases. During the first session of source 3 (UBR) between 0.1413 and 0.3591 seconds, traffic of source 1 (CBR-BE) end-to-end delay increases up to 8.7 milliseconds. Traffic of source 1 (CBR-BE) has its maximum delay of 26.6 milliseconds during the third session of source 3 (UBR) which is between 1.2029 and 2.556 seconds. During the link failure and the fourth session of source 3 (UBR), a packet of source 1 (CBR-BE) delay reaches up to 24.5 milliseconds at 3.958 seconds. After the link restoration, and in the absence of the traffic of source 3 (UBR), traffic of source 1 (CBR-BE) end-to-end delay remains at 2.4 to 3.1 milliseconds according to the queuing delay values (see Figure 5.8).

The increase in the end-to-end delay of the traffic of source 1 (CBR-BE) is due to the queuing delay. This can be confirmed by considering queuing delay graph of Figure 5.8. For example, traffic of source 1 (CBR-BE) has an end-to-end delay of 26.6 milliseconds at 2.5625 seconds. At the same time, in the queuing delay graph it has a queuing delay of 24.4 milliseconds, which when added to the transmission and propagation delay sums to 26.6 milliseconds. Also, consider traffic of source 1 (CBR-BE) maximum queuing delay during the link failure which is 17.2 milliseconds. When this value is added to the total propagation and transmission delay which is this time 6.056 ($3.2 + 2.856$), it produces 23.256 milliseconds. If we read the value at the corresponding point in the end-to-end delay graph, we find that it is 24.5 milliseconds. This value is almost the same calculated

value considering queuing and switching delay in the remaining routers of the LSP.

Traffic of source 2 (CBR-EF) has a minimum end-to-end delay of 2.3 milliseconds in some instances when there is no queuing delay. Traffic of source 2 (CBR-EF) end-to-end delay most of the time ranges between 2.4 and 2.9 milliseconds as in its end-to-end delay graph of Figure 5.7. Keeping in mind that these values are measured in the absence of the traffic of source 3 (UBR) and off the link failure period. During source 3 (UBR) sessions, traffic of source 2 (CBR-EF) delay increases to values between 2.7 milliseconds at minimum and 4.0 milliseconds. Its delay reaches up to 4.1 and 4.3 milliseconds in some cases (see the values corresponding to 0.2858 and 1.5509 seconds in traffic of source 2 (CBR-EF) end-to-end delay graph of Figure 5.7). During the link failure, traffic of source 2 (CBR-EF) end-to-end delay reaches between 6.4 and 9.0 milliseconds. Its delay reaches to 9.7-9.8 milliseconds during the link failure and source 3 (UBR) fourth session (at 3.1808 and 3.985 seconds). After the link restoration, traffic of source 2 the CBR-EF end-to-end delay decreases and remains between 2.4 and 2.9 milliseconds depending on the queuing delay measures.

The traffic of source 2 (CBR-EF) suffers less queuing delay compared to the traffic of source 1 (CBR-BE) and source 3 (UBR). This is observed in the queuing delay graphs of the traffic of the CBR sources shown in Figure 5.8. During the link failure such as at 3.5413 seconds, a packet of source 2 (CBR-EF) has a queuing delay of 2.2 milliseconds and at the corresponding point in the end-to-end delay graph it has a delay of 9.2 milliseconds. This is equal to the sum of the transmission, propagation, queuing, and switching delays. Recall that the total propagation delay for the CBR-EF source, using the alternative LSP,

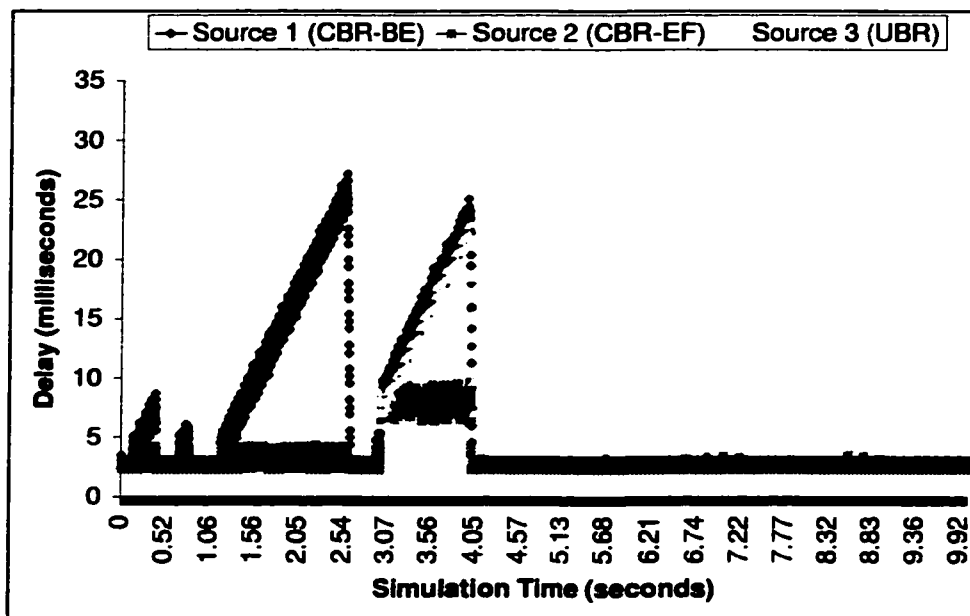


Figure 5.7: End-to-end Delay for the Network Traffic.

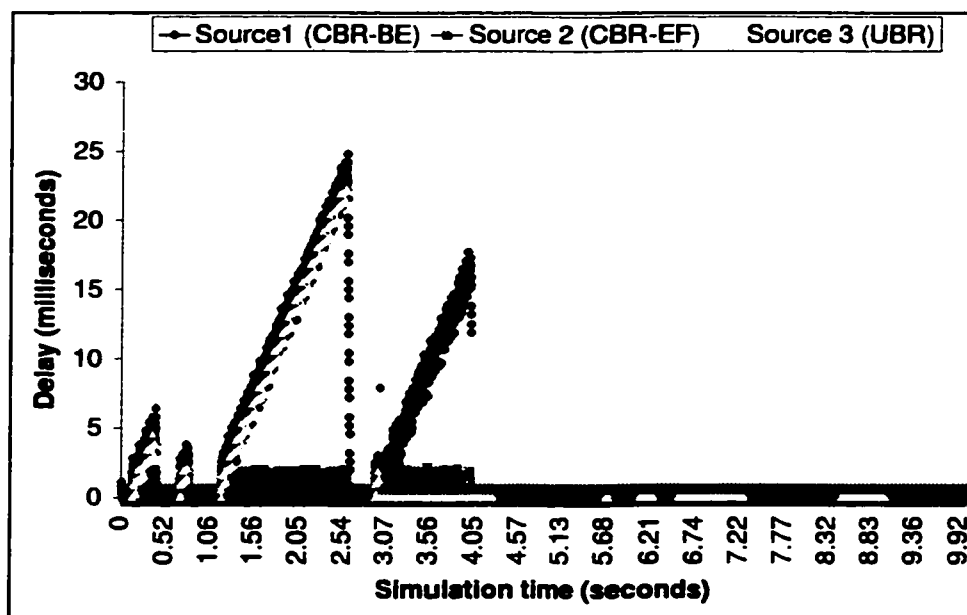


Figure 5.8: Queuing Delay for the Network Traffic.

is 3.2 milliseconds and the total transmission time is 2.72 milliseconds. We have also to consider the switching and queuing delays in two more hops.

Traffic of source 3 (UBR) end-to-end delay starts with a value of 7.1 milliseconds. This value increases with time as a result of the increase in the queuing delay. Traffic of source 3 (UBR) end-to-end delay reaches its maximum value of 29.3 milliseconds during its third session. During the link failure period, traffic of source 3 (UBR) end-to-end delay goes up to 23 milliseconds (refer to Figure 5.8). After the link restoration, traffic of source 3 (UBR) end-to-end delay drops to its minimum value of 7.0 milliseconds and remains constant until the end of the simulation (see the end-to-end delay graph). This is because traffic of source 3 (UBR) traverses to its destination using the alternative LSP which is not shared by the traffic of any other source. Hence, packets of source 3 (UBR) suffer no queuing delay as can be seen in the traffic of source 3 (UBR) queuing delay graph (see the value between 4.08 - 8.93 seconds).

As the case with the traffic of the two CBR sources, the variation in the end-to-end delay is due to the variation in the queuing delay. Packets of source 3 (UBR) are queued at router 2 in the first three seconds of the simulation and at router 6 during the link failure and afterwards. For example, consider the value at 3.942 seconds in the queuing delay graph which is 15.9 milliseconds. This value would result in an end-to-end delay of 22.9 milliseconds which is exactly the value at the corresponding point in the end-to-end delay graph. It is worth mentioning that the end-to-end for the traffic of source 3 (UBR) using the default and the alternative LSP is the same.

To confirm the simulation results and to justify values of queuing delay measured in

this experiment, we have measured the queue length at routers 2 and 6 for both types of traffic. Figure 5.9 shows the queue length of the traffic of source 2 (CBR-EF) at router 2 and at router 6 which is measured during the link failure and the traffic is rerouted through it. The queue length at router 2 is always between 0 and 1. That is why traffic of

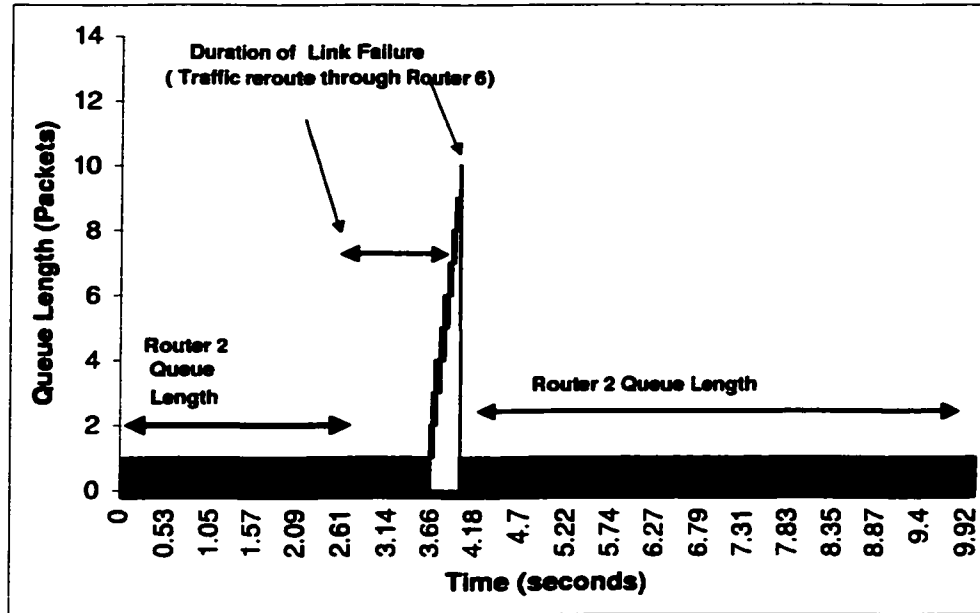


Figure 5.9: Queue Length for Traffic of Source 2 (CBR-EF) at Routers 2 and 6.

source 2 (CBR-EF) does not suffer high queuing delay and hence end-to-end delay. Traffic of source 2 (CBR-EF) queue length at router 6 is between 0 and 1 for the first half second of the link failure. This indicates that the increase in the traffic of source 2 (CBR-EF) end-to-end delay, during the link failure, is mainly due to the cost of the alternative LSP. Considering the queue length at 3.6 to 4.0 seconds, traffic of source 2 (CBR-EF) queue length goes up to 9 packets. Based on that it is logical to have increased queuing delay and by induction end-to-end delay, but that is not the case considering graphs of Figure 5.7. The reason behind that is source 2 (CBR-EF) packets drop takes place at this period.

Since the dropping mechanism adopted in the bf NS simulator is drop tail it is natural that these packets get dropped and their measures do not appear in the graphs. Next to that queue length graph shows the regular form of 0 and 1. This again leads to low queuing delay and end-to-end delay.

The queue length of the BE traffic which is generated by source 1 (CBR-BE) and source 3 (UBR), at routers 2 and 6 is shown in Figure 5.10 and Figure 5.11. Graphs plotted in these two figures justifies the increased queuing delay of the traffic of the two BE traffic sources. The increase in the queuing delay of the traffic of source 2 (CBR-BE) and source 3 (UBR), is in accordance with corresponding values of the queue length. The maximum queue length for the BE traffic is at the third second and that is where traffic of both sources have their maximum queuing delay and end-to-end delay. It is worth mentioning that the increase in traffic of source 2 (CBR-BE) end-to-end delay is partially due to the queuing delay and partially due to the extra cost resulting from using the alternative LSP. Queue length at router 2 after the fourth second is due to the traffic of source 2 (CBR-BE) since traffic of source 3 (UBR) is no more flowing through that router. At the same time, the queue length at router 6 is of the traffic of source 3 (UBR) traffic of source 1 (CBR-BE) after the link restoration returns to its default LSP. In both cases the queue length is at minimum and the queuing delay and end-to-end delay measures at this period agree with the queue length measures.

From this experiment, we conclude that MPLS in the core routers has almost zero switching delay. The other observation, a flow in MPLS network having two LSPs with equivalent cost would choose the *tunneled* of them although it could be more congested.

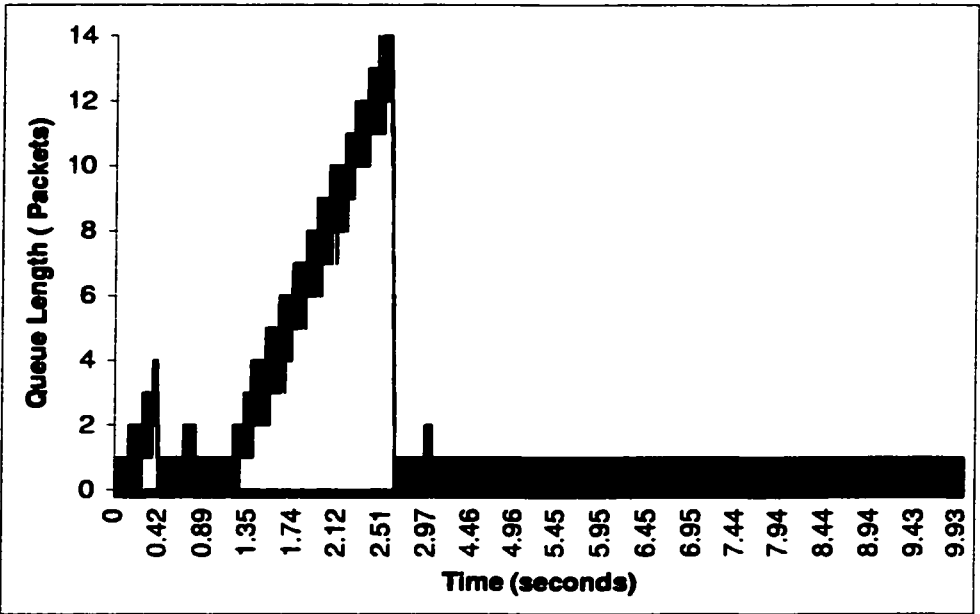


Figure 5.10: Queue Length for the BE Traffic at Routers 2.

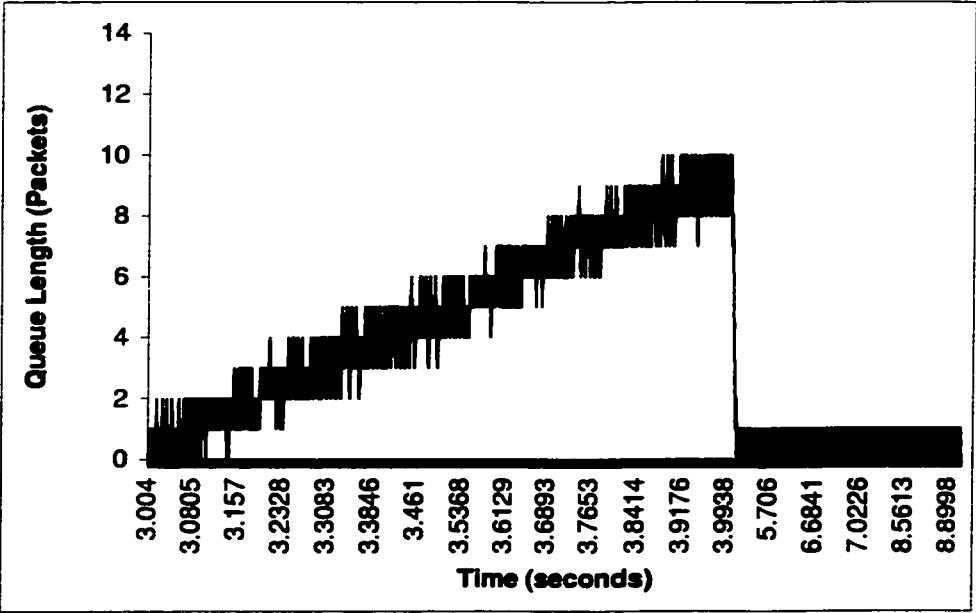


Figure 5.11: Queue Length for the BE Traffic at Router 6.

This is observed in the behavior of the traffic of source 3 (UBR) before the link failure. Later, as a result of changes in the network dynamics, if a flow finds a less congested route, it shifts to that route. This is observed in the behavior of the traffic of source 3 (UBR) after link restoration. When the network restores operation after the link failure resources are better utilized and the network performance improves.

5.2.3 Results for 1.7 Mbps Core Links Network

To examine the network performance under high competition for resources, we have reduced the core links capacity to 1.7 Mbps. This increases the transmission and end-to-end delays as seen in Table 5.2, and also the queuing delay.

Figures 5.12 and 5.13 show simulation results for the end-to-end delay and queuing delay of the traffic of the network sources. Simulation results agree with the mathematically calculated values. Recognize that traffic of source 1 (CBR-BE) end-to-end delay starts with a value of 2.5 milliseconds and increases with the increase in the queuing delay. The effect of source 3 (UBR) sessions on the traffic of source 1 (CBR-BE) measures is very clear. For example, traffic of source 1 (CBR-BE) end-to-end delay increases at 0.1567 seconds where source 3 (UBR) starts its first session. At 0.3951 seconds, source 3 (UBR) goes off and hence traffic of source 1 (CBR-BE) delay decreases. Its delay starts increasing again at 0.6956 seconds and that is when source 3 (UBR) starts its second session. At 3.0 seconds, queue length at router 2 decreases since source 3 (UBR) is no more sending packets. As a result, traffic of source 1 (CBR-BE) queuing delay decreases and its end-to-end delay decreases till it reaches steady value of around 11 milliseconds (3.0-3.5 seconds).

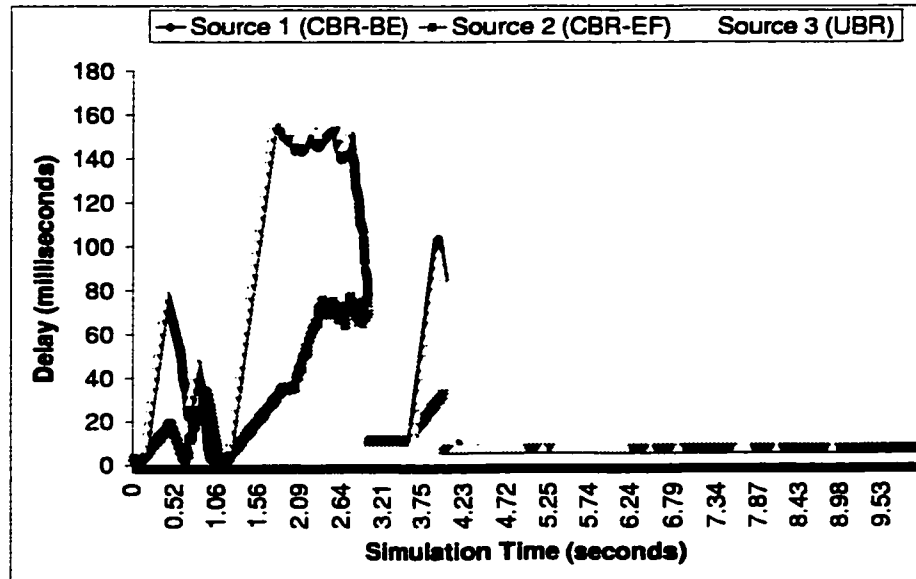


Figure 5.12: End-to-end Delay for the Network traffic (1.7 Mbps Links.)

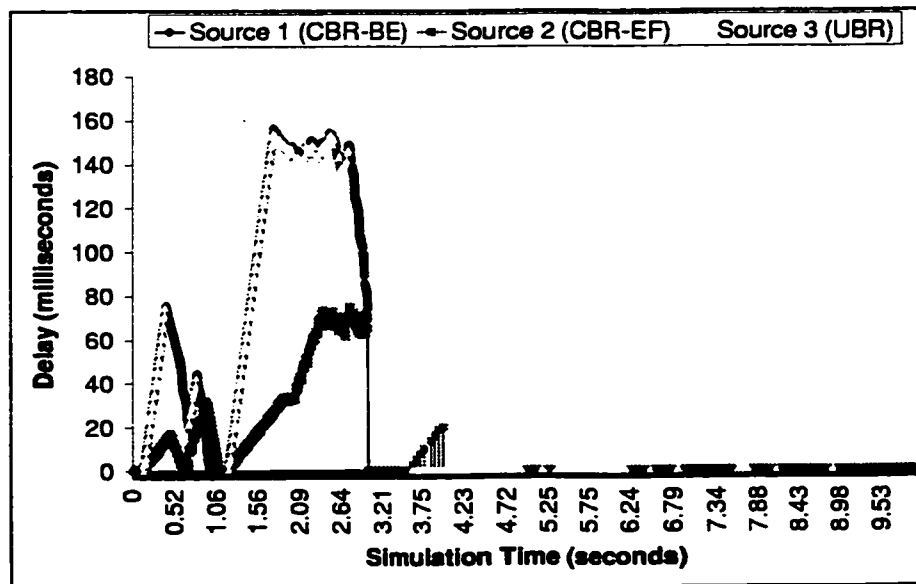


Figure 5.13: Queuing Delay for the Network Traffic (1.7 Mbps Links.)

At 3.5 seconds, source 3 (UBR) starts its first session during the link failure and hence, traffic of source 1 (CBR-BE) delay increases. After the link restoration, traffic of source 1 (CBR-BE) has a stable end-to-end delay of 2.6 milliseconds since traffic of source 3 (UBR) is no longer sharing traffic of source 1 (CBR-BE) LSP.

From the simulation results, traffic of source 2 (CBR-EF) end-to-end delay starts by a value of 2.5 milliseconds for the first arriving packet. source 3 (UBR) sessions have less effect on the traffic of source 2 (CBR-EF) end-to-end delay than that source 3 sessions have on traffic of source 1 (CBR-BE) performance. This can be concluded by observing the effect of source 3 sessions on traffic of both sources delays.

Traffic of source 3 (UBR) suffers the highest end-to-end delay among the traffic of the three sources. This is mainly due to the traffic LSP cost. Considering the queuing delay, traffic of source 3 (UBR) has almost the same queuing delay of the traffic of source 1 (CBR-BE) during the time they share routers and compete for resource.

From the simulation results, source 3 (UBR) first packet has an end-to-end delay of 7.7 milliseconds. The end-to-end delay of its first arriving packet during the link failure is 8.0 milliseconds. This is at 3.6165 seconds of the end-to-end delay graph where the UBR source starts its first session during the link failure. Next to that traffic of source 3 (UBR) queuing delay increases and hence its end-to-end delay. After the link restoration, traffic of source 3 (UBR) end-to-end delay has its minimum value which is 7.7 milliseconds. This is due to the very low queuing delay at router 6 (see the end-to-end delay and queuing delay graphs).

After the link restoration, traffic of source 3 (UBR) continue to use the alternative

LSP. This leads to a recognizable improvement in the whole network performance.

Figure 5.14 shows packet loss of the three sources. Packet drop takes place at router 2 in the timings shown in the graphs. The network starts dropping packets of source 1

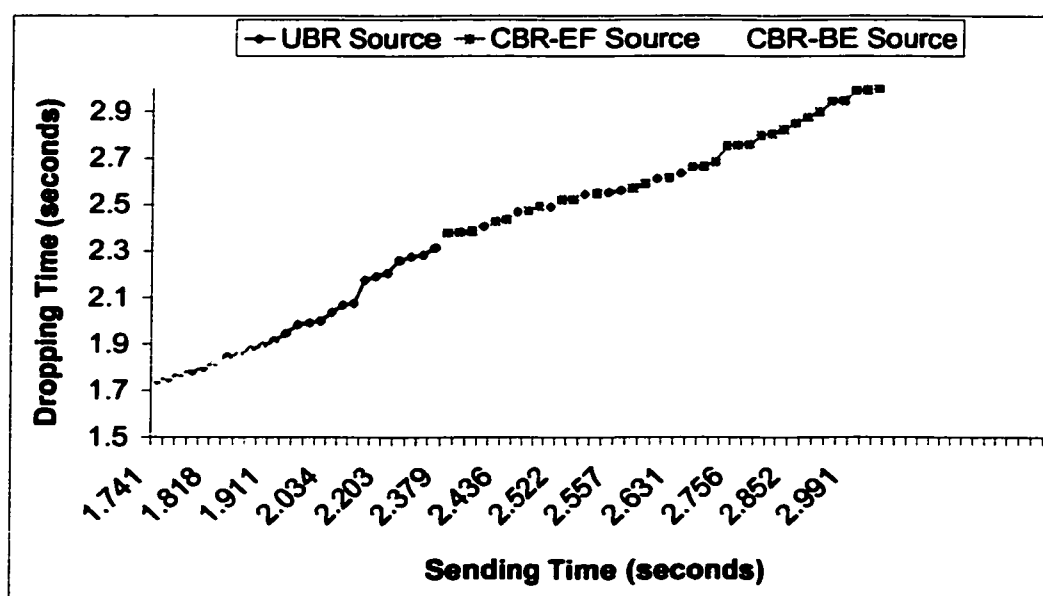


Figure 5.14: Traffic Sources Packet Loss at router 2 (1.7 Mbps Links.)

(CBR-BE) and source 3 (UBR) at 1.7 seconds when source 3 (UBR) starts its first session. This is because both sources produce best effort traffic and this results in high competition for the BE service resources. In other words, BE-service queue gets accumulated and the network starts dropping the BE-service traffic. source 2 (CBR-EF) first packet dropped was at 2.3786 seconds. This means, traffic of source 2 (CBR-EF) is dropped when there is very high need for dropping more packets at router 2. From this result, we conclude that the MPLS network provides fast service to packets marked to receive expedite forwarding. At the same time traffic of EF sources has the least packet loss count.

In this experiment, the queue length for both types of traffic EF and BE is measured. This is done to examine and justify the network behavior. It could also be considered as a conformance test for the measures collected. Figure 5.15 shows the EF and BE traffic queue length at router 6. The EF traffic is from source 2 (CBR-EF) and the BE traffic is from source 1 (CBR-BE) and source 3 (UBR). Traffic of source 2 (CBR-EF) queue length between 3.0 and 3.6 seconds, is between 0 and 1. The queuing delay and end-to-end delay of the traffic of this source agree with these queue length values. It is worth noting, the increase in the end-to-end delay during this period is partially due to the alternative LSP delay. Traffic of source 2 (CBR-EF) queue length increases at 3.6 up to 4.0 seconds. That is when traffic of source 3 (UBR) starts sharing the resources with the traffic of source 1 CBR and source 2 (CBR-EF) the same LSP. The resources sharing increases traffic of source 2 (CBR-EF) queuing end-to-end delay (See traffic of source 2 (CBR-EF) graphs). At 3.0-3.6 seconds, source 3 (UBR) was off which means only traffic of source 1 (CBR-BE) is present at router 6 BE service queue. That is why we see in the figure a queue length of 1 at maximum. Considering traffic of source 1 (CBR-BE) end-to-end delay, it decreases although the traffic is now traversing a longer LSP. This is due to the absence of the traffic of source 3 (UBR) which reduces the BE service queuing delay. Towards the end of the fourth second, source 3 (UBR) goes on and as a result the BE traffic queue length increases. The effect of this is clear on the traffic of source 1 (CBR-BE) queuing and end-to-end delay. Traffic of source 3 (UBR) has slight effect on the traffic of source 2 (CBR-EF) not as the case with traffic of source 1 (CBR-BE). Notice that the queue length values present in the figure next to the fourth second are of source 3 (UBR) traffic only.

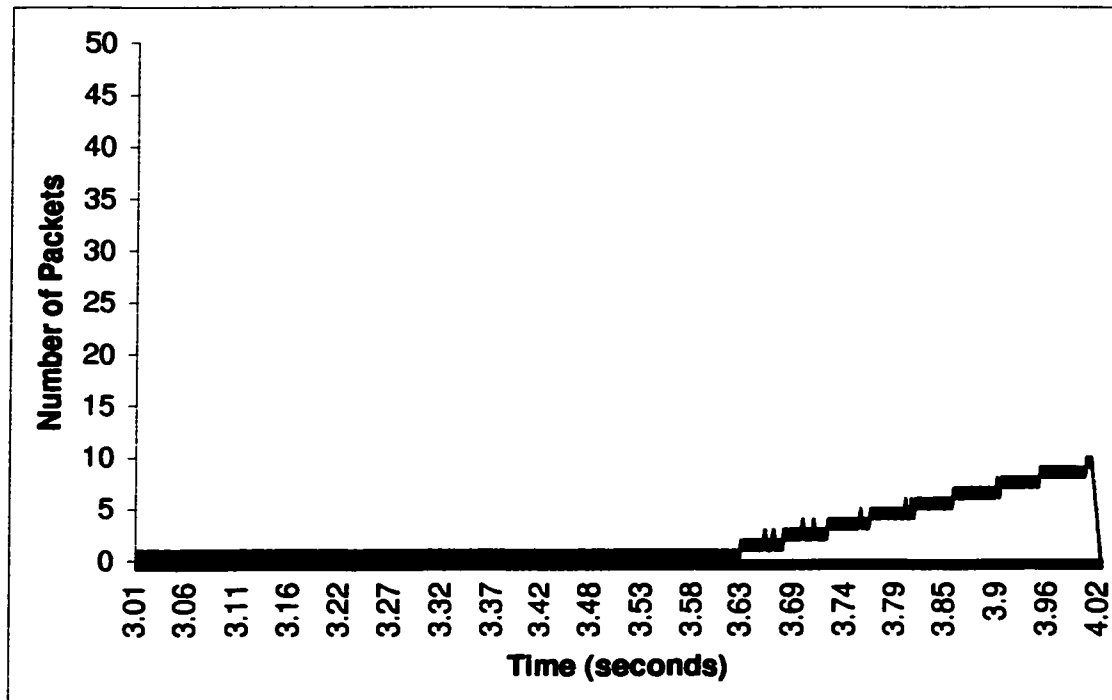


Figure 5.15: Queue Length for Traffic of Source 1 (CBR-BE) at Router 6.)

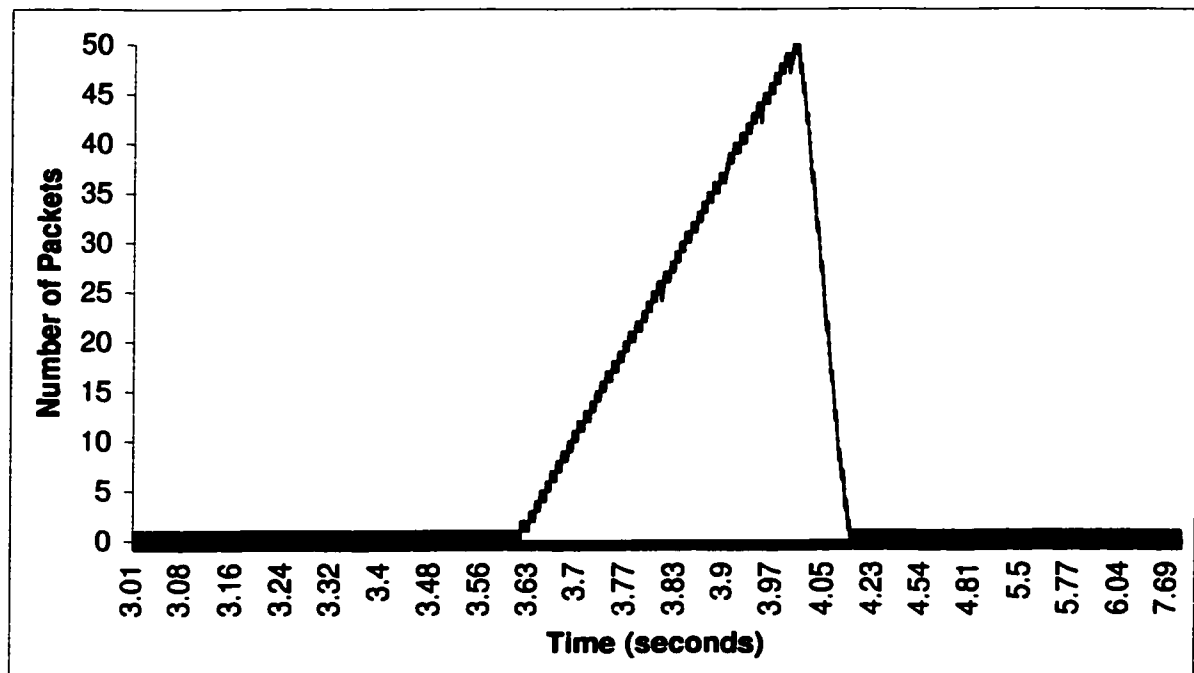


Figure 5.16: Queue Length for BE Traffic at Router 6.)

5.2.4 Results for CBR-EF Sources Network

To study the effect of having more traffic of the expedite forwarding type on the network performance, we have made both CBR sources to generate EF traffic.

Figures 5.17 and 5.18 show the end-to-end delay and queuing delay experienced by the network traffic. Traffic of both (CBR-EF) sources is affected the same way by source 3 (UBR) active sessions. As seen in the graphs, when source 3 (UBR) goes off, both sources end-to-end delay drops to its least value. During source 3 (UBR) active sessions, resources available for the EF service are fairly shared by the two CBR sources and hence their delays increases in the same ratio.

From the queuing delay graphs, we find that traffic of both CBR-EF sources has the minimum queuing delay after the link restoration. Considering the corresponding values of the end-to-end delay, we find that traffic of both sources has the minimum end-to-end delay in this period.

During source 3 (UBR) active sessions, its delay is slightly affected and it departs from its minimum value of 7.0 milliseconds to not more than 0.8 milliseconds. At the corresponding times, traffic of the CBR-EF sources end-to-end delay increases by an average value of 4.0 milliseconds. Traffic of source 3 (UBR) has the least queuing delay. This is because traffic of source 3 (UBR) is the only traffic to receive best effort service. The result is that it has the lower end-to-end delay which means it has been served better than the EF traffic. For example, at 0.595, 2.875, and 3.83 seconds, we find that traffic of source 3 (UBR) suffers the least end-to-end delay.

Here, we recall that one of the problems in the Intserv approach is that the network

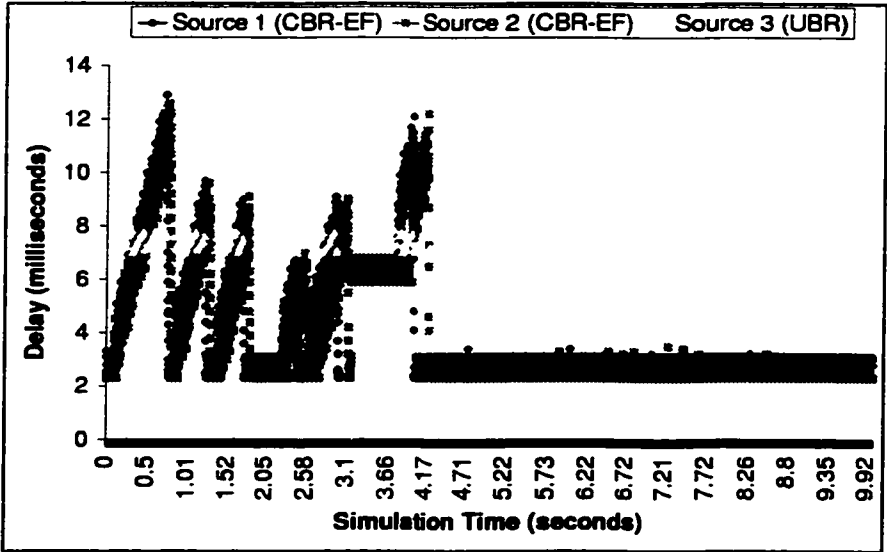


Figure 5.17: End-to-end Delay for the Network Traffic.

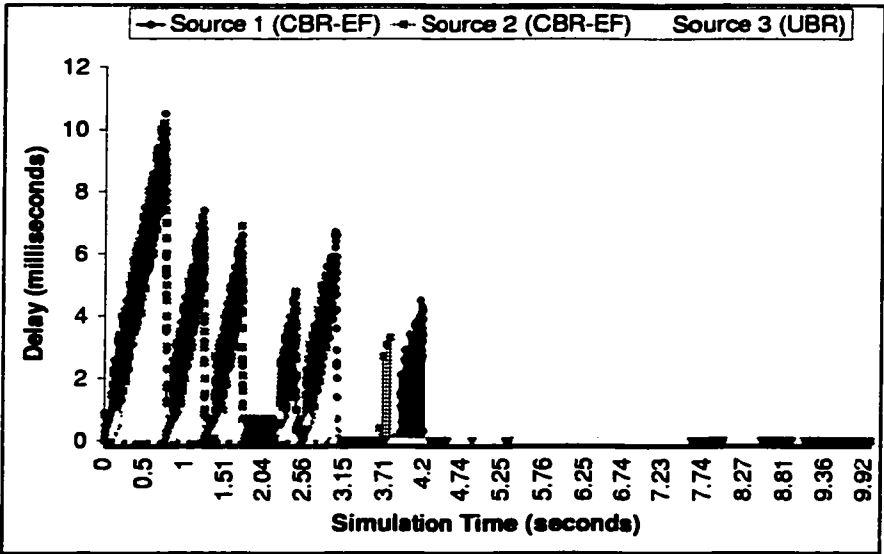


Figure 5.18: Queuing Delay for the Network Traffic.

may end up providing high quality of service to a traffic class and nothing for another class. In other words, all service to EF traffic and nothing for BE traffic. Our results in this experiment has shown that MPLS does not suffer from the same problem.

The linear relationship between the queuing delay and the end-to-end delay is confirmed once more by the graphs of the traffic of the network measures. This can be seen by reading the queuing delay values with the corresponding end-to-end delay values of any of the three sources. Recognize that after the link restoration and the departure of the traffic of source 3 (UBR) from its default LSP, traffic of source 3 (UBR) has an end-to-end delay of 7.0 to 7.1 milliseconds. Reading traffic of source 3 (UBR) queuing delay, we find that it has almost zero queuing delay.

As a conclusion out of this experiment it has been found that MPLS even with Diffserv is not scalable unless appropriate aggregation scheme is adopted. This is true since both EF traffic sources got higher delay than the BE traffic source which should have not happened if perfect aggregation is employed.

5.3 Link Failure in Diffserv MPLS with Multiple LSPs

We extend the network of Section 6.1 to study the network behavior in the presence of alternative LSPs. We work out with two alternative LSPs and then with multiple alternative LSPs. We mean to examine the way Diffserv MPLS network pick an alternative LSP in case of link failure.

5.3.1 Two alternative LSPs Simulation and Results

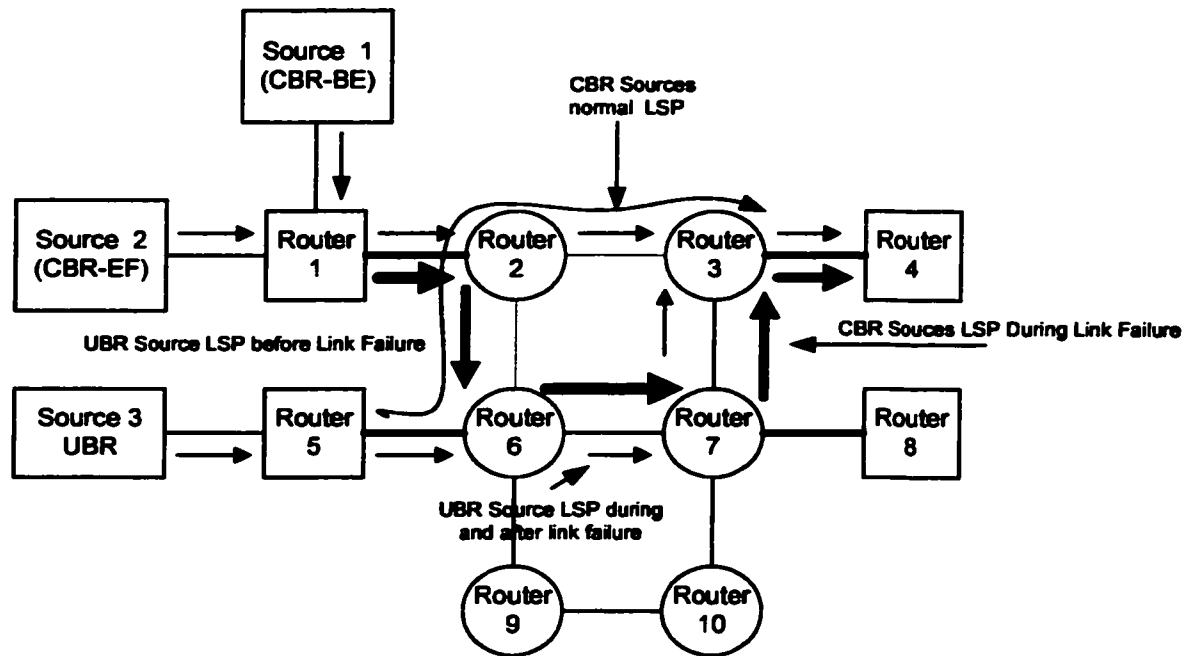


Figure 5.19: Link Failure with Two alternative LSPs.

Figure 5.19 shows the simulation network for this set of experiments. It is built of four edge LSRs and six core LSRs. Two of the edge LSRs (router 1 and 5) are source routers and the other two LSRs (router 4 and 8) are destination LSRs. The edge links are of 10 Mbps capacity and 0.1 milliseconds propagation delay each while the core links are of 2 Mbps capacity and 1.0 milliseconds propagation delay each.

The core LSRs are connected as shown so that during the link failure the traffic of the three sources going through router 6 has two alternative LSPs:

1. LSP1 which contains routers 6, 7, and 3.
2. LSP2 which contains routers 6, 9, 10, 7, and 3.

link is between router 2 and 9 and the other link is between router 10 and 3. The addition of these two links results in five alternative LSPs that can be used to reroute traffic during the link failure. The five alternative LSPs are:

1. LSP1 which contains routers 6, 7, and 3
2. LSP2 which contains routers 9, 10, and 3.
3. LSP3 which contains routers 6, 9, 10, and 3.
4. LSP4 which contains routers 9, 10, 7, and 3.
5. LSP5 which contains routers 6, 9, 10, 7, and 3.

In this simulation, we have made all the core links to have the same capacity and propagation delay. Here, LSP1 and LSP2 have the same cost because each of them contains the same number of links and routers. The same is true for LSP3 and LSP4. LSP5 has the highest cost since it contains the largest number of links and routers among the five LSPs.

During the link failure the Diffserv MPLS network rerouted the traffic of the two CBR sources using LSP2. This results in lower end-to-end delay for the traffic of the three sources. That is because in this case, no queuing delay has occurred for the traffic of the three sources since the traffic of the UBR source is flowing through router 6 and 7. As a result, the end-to-end delay of the two CBR sources increases to values equally to the sum of transmission delays and the propagation delays of the alternative LSP only. The traffic of source 3 (UBR) end-to-end delay remains at its minimum value.

Chapter 6

Conclusions and Future work

6.1 Summary

Our experimental work measures for the end-to-end delay have shown that MPLS in its basic form (Vanilla) reduces packet delay and hence average delay. This is concluded by comparing simulation results of a Vanilla MPLS network with those of an IP network. The fact that MPLS has lower end-to-end delay is justified by the following: processing in the MPLS network core routers is only a matter of label swapping and then forwarding which consumes less time than the processing of IP header performed in conventional IP routing. Though Vanilla MPLS reduces end-to-end delay, neither service guarantees have been provided nor differentiation in flows treatment have been observed.

Diff-serv MPLS simulation results have shown that it performs better than Vanilla MPLS and guarantees different treatment for applications based on type of traffic. This is true provided that sources or users data remained within the contracted or agreed upon

level. In Diffserv MPLS, traffic of expedite forwarding (EF) class suffers less end-to-end delay compared with traffic of best effort (BE) class.

From the simulation results, EF traffic end-to-end delay is not significantly affected by the increase in the volume of the BE traffic. If the volume of EF traffic goes beyond the agreed upon level, the situation changes and delay of the EF traffic becomes more than that of BE traffic. This is because EF traffic queue expands and competition for EF resources takes place. If the volume of BE traffic at the same time is low it suffers less end-to-end delay that may not include any queuing delay. In this case, BE traffic have no packet loss while EF traffic suffers packet loss. This shows the importance of keeping the volume of traffic on the agreed upon level. This also necessitate a method to monitor traffic sources behavior and to confirm that they remain within the allowed limits.

Among the components that constitute the end-to-end delay, the queuing delay is the most contributing component in most of the cases. Therefore the queuing delay of different data streams is measured. Collected results have shown that MPLS has negligible switching delay (almost zero) because as observed adding the constant transmission and propagation delays to the queuing delay produces almost the same measured end-to-end delay.

Varying a network resources by increasing and decreasing link capacities have clear effect on the network performance. The end-to-end delay increases and decreases linearly with the network link capacity. Increasing links capacities reduces the end-to-end delay. On the contrary, reducing link capacities where there will be competition for resources, increases the queuing delay and hence the end-to-end delay. Measures for varied link

capacity experiments confirmed the relationship between queuing delay and the end-to-end delay specially at the bottleneck nodes. In addition measures taken and analysis made proved the linear relationship between link capacity and the end-to-end delay.

MPLS enabled network reroutes traffic in case of link failure very quickly and directs traffic to an alternative route. Traffic returns to its default route if the link is restored and the default route which is restored is the best. Both types of traffic EF and BE end-to-end delay increases in case of link failure of either of the two. Although this is true, EF traffic suffers less delay and has better handling than the BE traffic which makes us recommend using EF in networks that have high probability of link failure.

The increase in the EF traffic amount in Diffserv MPLS networks beyond the share of the network resources reserved for EF service results in higher delay for the EF traffic than that of the BE traffic. This means that MPLS is not scalable unless appropriate aggregation scheme is adopted.

It is also observed that when a flow has two LSPs with equal delay and one of them is tunneled, MPLS would choose to send the flow through the tunneled one although it might be more congested than the other route. This because MPLS has very strong tunneling mechanism and hence the network relies on that when selecting LSPs. Later, as a result of changes in the network dynamics, if the flow finds a less congested route, it shifts to that route. MPLS networks have shown efficiency in rerouting traffic to the more suitable path among a number of alternatives in case of link failure. The selected path in this case serves the flow and improves over all network performance.

6.2 Conclusions

The results of our work can be listed in the following:

- MPLS enabled network reduces per packet and average end-to-end delay.
- MPLS in its basic form (Vanilla) does not give preference according to traffic type.
- Diff-Serv MPLS provides differentiated service and treats applications according to their traffic type.
- MPLS improves significantly the overall network performance and makes better utilization of resources.
- Traffic marked to receive EF service in MPLS networks gains a distinct treatment and benefits in all senses.
- Diff-Serv MPLS guarantees same level of service under all changes in network conditions, dynamic or static.
- Diff-Serv MPLS preserves a recognized fairness and does not provide all services on account of less service to lower priority traffic.
- Diff-Serv MPLS provides traffic engineering capabilities, e.g, reroute and explicit LSP enforcement.
- MPLS needs mechanisms to monitor traffic sources behavior.

6.3 Future Work

Aspects that can be considered for future research are:

- Study of other workload mixes.
- Address the issue of flow aggregation in MPLS networks to serve scalability.
- Integration of Admission Control.
- Study of performance and traffic mapping in IP and MPLS hybrid networks.

Bibliography

- [1] K. Nichols, V. Jacobson, and L. Zhang. A two-bit differentiated services architecture for the internet. *Network Working Group RFC: 2638*. Available at: <http://community.rozen.com/developers/idos/rfc/rfc2638.html>, July 1999.
- [2] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall, Englewood Cliffs, third international edition, 1996.
- [3] William Stallings. *High-Speed Networks TCP/IP and ATM Design Principles*. Collin Macmillan, Newyork: Macmillan, London, 1996.
- [4] John T. Moy. *OSPF Anatomy of an Internet Routing Protocol*. Addison Wesley Longman, Inc., first international edition, January 1998.
- [5] D. Waitzman, C. Partridge, and S. Deering. Distance vector multicast routing protocol. *Network Working Group RFC: 1075*, pages 1–24, November 1988.
- [6] Chuck Semeria and Frank Fuller. 3com's strategy for delivering differentiated service levels. *Technical report at www.3com.com/technology/tech_net/white_papers/500652.html*, 1997.
- [7] R. Braden, D. Clark, and S. Shenker. Integrated services in the internet architecture: an overview. *Network Working Group RFC: 1633*, pages 1–33, June 1994.
- [8] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers. *Network Working Group RFC: 2474 at www.ietf.org/rfc/rfc2474.txt*, pages 1–20, December 1998.
- [9] Eric C., Arun Viswanathan, and Ross Callon. Multiprotocol label switching architecture. *RFCs: draft-ietf-mpsls-arch-05.txt*, pages 1–62, October 1999.
- [10] Chuck Semeria. Enhanced routing in the new public network. *White Paper a tutorial at www.juniper.net*, 2000.
- [11] Leonard Kleinrock. On the modeling and analysis of computer networks. *Proceedings of the IEEE Conference on Computer Networks*, 81, Issue:8:1179–1191, August 1993.
- [12] T. E. Tedijano and R. O. Onvural. Nbbs path selection framework. *IBM Systems Journal*, 34, No. 4, 1996.

- [13] Jin Liu. An improved vectorial labeling scheme for qos-guaranteed routing algorithm of ATM networks. *Proceedings of the International Conference on communication technology*, October 1998-Beijing China.
- [14] Antonious F. Atlasis, Evangelos D. Baltazis, George I. Stassinopoulos, and Iakvos S. Venieris. A linear-based trunk reservation routing algorithm for ATM networks. *International Journal of Communication Systems*, 12:125–141, 1999.
- [15] Yoshio F. Turner and Yuval Tamir. Connection-based adaptive routing using dynamic virtual circuits. *A paper at www.cs.ucla.edu/yoshio/pdc98.pdf*, 1999.
- [16] J. M. Jaffe. Algorithm for finding path with multiple constraints. *Computer Networks Journal*, 14:95–116, 1998.
- [17] Shigang Chen and Klara Nahrstedt. On finding multi-constraints paths. *A tutorial at www.cora.jpcc.com*, 1998.
- [18] S. Chen K. Nahrstedt. Distributed qos routing with imprecise state information. *Proceedings of the IEEE Seventh International Conference on Computer, Communications and Networks, Lafayette, LA*, October 1998.
- [19] F. Xiang, L. Junzhou, W. Jieyi, and G. Guanqun. Qos routing based on genetic algorithm. *Computer Communication Journal*, 22:1392–1399, 1999.
- [20] N. F. Huang, C. S. Wu, and Y. J. Wu. A performance study of multicast routing algorithms for ATM networks. *Computer Networks ISDN Systems*, 27:101–116, 1994.
- [21] Antonious F. Atlasis. An adaptive routing algorithm for ATM networks using learning automaton. *Proceedings of the IEEE Conference on Networks*, pages 2669–2673, September 1998.
- [22] K. R. Krishnan and R. H. Cardwell. Routing and virtual-path design in ATM networks. *Proceedings of the IEEE Conference on Global Communications*, pages 765–769, 1994.
- [23] Devika Subramanian, Peter Druschel, and Johnny Chen. Ants and reinforcement learning: A case study in routing in dynamic networks. *www.citeseer.nj.nec.com/did/48041*, 1998.
- [24] Tim Harrison and Carey Williamson. Some routing problems on broadband ISDN. *Proceedings of the IEEE Conference on Computer Networks*, pages 191–201, 1996.
- [25] T. Ballaradie, P. Francis, and J. Crowcroft. Core-based trees (cbt): An architecture for scalable inter-domain multicast routing. *Proceedings of ACM SIGCOMM, 99, San Francisco, California*, pages 85–95, September 1993.
- [26] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource reservation protocol (RSVP) version 1 functional specification. *Network Working Group RFC: 2205*, pages 1–111, September 1997.

- [27] J. Wroclawski. Specification of the controlled-load network element service. *Network Working Group RFC: 2211*, pages 1–18, September 1997.
- [28] S. Shenker, C. Partridge, and R. Guerin. Specification of guaranteed quality of service. *Network Working Group RFC: 2212*, pages 1–20, September 1997.
- [29] Markus Alexander Wischy. Area analysis: Integrated and differentiated services. *A Tutorial at <http://studweb.studserv.uni-stuttgart.intserv>*, May 7th, 1998.
- [30] Scott W. Brim, Brian E. Carpenter, and Francois Le Faucheur. Per hop behavior identification codes. *Network Working Group RFC: 2836*, pages 1–7, May 2000.
- [31] Steven Blake, David L. Black, Mark A. Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss. An architecture for differentiated services. *Network Working Group RFC: 2475 at www.ietf.org/rfc/rfc2475.txt*, pages 1–36, December 1998.
- [32] Mary Petrosky. Diffserv: Something old, something new. *www.performancecomputing.com/columns/packets/9903.html*, March 1999.
- [33] D. Clark and J. Wroclawski. An approach to service allocation in the internet, work in progress. *A talk by D. Clark in the Int-Serv Working Group at Munich IETF*, August, 1997.
- [34] V. Jacobson. Differentiated services architecture. *A talk in the Int-Serv Working Group at Munich IETF*, August, 1997.
- [35] Peter Ashwood and Bilal N. Jamoussi. Mpls tutorial. *www.nanog.org/mtg-9905/ppt/mps/sld001.htm*, 2000.
- [36] Yakov Rekhter, Bruce Davie, Dave Katz, Eric Rosen, and George Swallow. Tag switching architecture. *an Internet-Draft at www.cisco.com/warp/public/732/tag/tagsw.ov.htm*, October 2000.
- [37] Thomas M. Chen and Tae H. Oh. Reliable service in MPLS. *IEEE Communications Magazine*, pages 58–61, December 1999.
- [38] Anoop Ghanwani, Bilal Jamousi, Don Fedyk, Peter Ashwood, Li Li, Nancy Feldman, and T. J. Watson. Traffic engineering standards in ip networks using MPLS. *IEEE Communications Magazine*, pages 49–53, December 1999.
- [39] James R. Leu, Robert Rennison, and Steve Vogelsang. Label aggregation technique for LDP. *Network Working Group Internet Draft at [ftp://laurelnetworks.com/pub/mps/draft-leu-mpls-ldp-label-aggregation-00.txt](http://laurelnetworks.com/pub/mps/draft-leu-mpls-ldp-label-aggregation-00.txt)*, July 2000.
- [40] Van Jacobson. Differentiated services for the internet. *First Internet2 Joint Application/Engineering QoS Workshop*, May 1998.

- [41] Francois Le Faucheur, Liwen Wu, Bruce Davie, Shahram Davari, Pasi Vaananen, Ram Krishnan, Pierrick Cheval, and Juha Heinanen. Mpls support of differentiated services. *RFC mpls-diff-ext-04.txt*, March 2000.
- [42] Juniper Networks Group. Junos ldp protocol implementation. *technical publications at: www.juniper.net/techpubs/software/junos40*, 2000.
- [43] Gerald R. Ash, Muckai K. Girish, Eric W. Gray, Bilel Jamoussi, and Gregory Wright. Applicability statement for cr-ldp. *draft-ietf-mpls-crldp-applic-01.txt*, pages 1–5, July 2000.
- [44] Bruce Davie, Yakov Rekhter, Eric Rosen, Arun Viswanathan, Vijay Srinivasan, and Steven Blake. Use of label switching with rsvp. *RFCs: draft-ietf-mpls-arch-00.txt*, pages 1–12, March 1998.
- [45] Daniel Awduche, Der-Hwa Gan, Tony Li, George Swallow, and Vijay Srinivasan. Extensions to rsvp for traffic engineering. *draft-swallow-mpls-rsvp-trafeng-00.txt*, September 1999.
- [46] Xipeng Xiao, Alan Hannan, brook Baily, and Lionel M. Ni. Traffic engineering with mpls in the internet. *Proceedings of the IEEE*, March 2000.
- [47] The ns manual (formerly ns notes and documentation). In Kevin Fall and Kannan Varadhan, editors, *The VINT Project: A collaboratoin between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC*, pages 1–313. www.isi.edu/nsnam/ns, 2001.
- [48] Gaeil Ahn and Woojik Chun. Mpls network simulator. *at: www.raonet.com/mns*, 2000.
- [49] Sean Murphy. Diffserv additions to ns-2. *Dublin City University, www.teltec.dcu.ie/murphys/*, 2000.
- [50] Ashley Stephenson. Diffserv and MPLS: A quality choice. *Technical Tutorial 21 at www.data.com/issue/981121/quality.html*, November 1998.

Vitae

- **Muhammed Yassir Obeid.**
- **Born in Khartoum, Sudan.**
- **Received Bachelor of Engineering (B.E.) degree in Computer Engineering from NED University, Karachi, Pakistan.**
- **Worked as a teaching assistant in Sudan University of Science and Technology, Khartoum Sudan.**
- **Joined the Department of Computer Engineering at KFUPM as a Research Assistant in September 1998.**
- **Completed Master of Science (M.S.) in Computer Engineering at KFUPM in August 2001.**
- **Email: yassirmohd@yahoo.com**