

An MPLS-based Quality of Service Architecture for Heterogeneous Networks

Srihari Raghavan

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science

Dr. Scott F. Midkiff, Co-Chair
Dr. Srinidhi Varadarajan, Co-Chair
Dr. Sallie M. Henry

November 12, 2001
Blacksburg, Virginia

An MPLS-based Quality of Service Architecture for Heterogeneous Networks

Srihari Raghavan
Computer Science

(ABSTRACT)

This thesis proposes a multi-protocol label switching (MPLS)-based architecture to provide quality of service (QoS) for both internet service provider (ISP) networks and backbone Internet Protocol (IP) networks that are heterogeneous in nature. Heterogeneous networks are present due to the use of different link-layer mechanisms in the current Internet. Copper-based links, fiber-based links, and wireless links are some examples of different physical media that lead to different link-layer mechanisms. The proposed architecture uses generalized MPLS and other MPLS features to combat heterogeneity. The proposed architecture leverages the QoS capabilities of asynchronous transfer mode (ATM) and the scalability advantages of the IP differentiated services (DiffServ) architecture. This architecture is constructed in such a way that MPLS interacts with DiffServ in the backbone networks while performing ATM-like QoS enforcement in the periphery of the networks. The architecture supports traffic engineering through MPLS explicit paths. MPLS network management, bandwidth broker capabilities, and customizability is handled through domain specific MPLS management entities that use the Common Open Policy Service (COPS) protocol to interact with other MPLS entities like MPLS label switch routers and label edge routers.

The thesis provides a description of MPLS and QoS, followed by a discussion of the motivation for a new architecture. The MPLS-based architecture is then discussed and compared against similar architectures. To integrate the ATM and DiffServ QoS attributes into this architecture, MPLS signaling protocols are used. There are two common MPLS signaling protocols. They are Resource Reservation Protocol with traffic engineering extensions (RSVP-TE) and Constraint-Routed Label Distribution Protocol (CR-LDP). Both these protocols offer comparative MPLS features for constraint routed label switch path construction, maintenance, and termination. RSVP-TE uses UDP and IP, while CR-LDP uses TCP. This architecture proposes a multi-level domain of operation where CR-LDP operates in internet service provider (ISP) networks and RSVP-TE operates in backbone networks along with DiffServ. Qualitative analysis for this choice of domain of operation of the signaling protocols is then presented. Quantitative analysis through simulation demonstrates the advantages of combining DiffServ and MPLS in the backbone. The simulation setup compares the network performance in handling mixed ill-behaved and well-behaved traffic in the same link, with different levels of DiffServ and MPLS integration in the network. The simulation results demonstrate the advantages of integrating the QoS features of DiffServ, ATM functionality, and MPLS into a single architecture.

ACKNOWLEDGEMENTS	7
CHAPTER 1. INTRODUCTION	8
CHAPTER 2. BACKGROUND	9
2.1. MULTIPROTOCOL LABEL SWITCHING	9
2.1.1. <i>Introduction</i>	9
2.1.2. <i>Operation of MPLS</i>	9
2.1.2.1. Label Switch Router Forwarding Component.....	10
2.1.2.2. Label Switch Router Control Component	12
2.2. NETWORK QUALITY OF SERVICE	15
2.2.1. <i>Introduction</i>	15
2.2.2. <i>History and the Need for Quality of Service</i>	15
2.2.3. <i>Quality of Service Architecture</i>	15
2.2.3.1. Quality of Service Metrics	16
2.2.3.2. Service Level Agreements.....	16
2.2.3.3. Traffic Classification.....	17
2.2.3.4. Congestion Management.....	17
2.2.3.5. Congestion Avoidance	17
2.2.3.6. Policing and Marking.....	18
2.2.3.7. Shaping	18
2.2.4. <i>Quality of Service Signaling</i>	18
2.2.5. <i>Service Models</i>	18
2.2.5.1. Integrated Services	18
2.2.5.2. Differentiated Services.....	19
2.2.6. <i>Policy-based Routing</i>	19
2.3. <i>Summary</i>	19
CHAPTER 3. MPLS-BASED QOS ARCHITECTURE	20
3.1. BACKGROUND	20
3.2. MOTIVATION	20
3.3. DETAILS OF THE MPLS-BASED QOS ARCHITECTURE.....	21
3.3.1. <i>Architecture Details</i>	23
3.3.1.1. QoS Mapping and Intra-domain Signaling.....	26
3.3.1.2. Admission Testing.....	35
3.3.1.3. QoS Control	41
3.3.1.4. QoS Management.....	44
3.3.2. <i>Architecture Comparisons</i>	50
3.3.2.1. Summary of QoS Architectures.....	51
3.3.2.2. Qualitative Comparison of Architectures	52
3.4. SUMMARY	54
CHAPTER 4. QUALITATIVE ANALYSIS OF RSVP-TE AND CR-LDP	55
4.1. QUALITATIVE ANALYSIS	55
4.1.1. <i>LSP Setup</i>	55
4.1.1.1. LSP Security	57
4.1.1.2. Upper Layer Protocol	58
4.1.2. <i>ER-LSP and CR-LSP Maintenance</i>	58
4.1.2.1. Scalability.....	58
4.1.2.2. Availability.....	60
4.1.2.3. Failure Detection and Error Reporting	61
4.1.2.4. Fault Tolerant Traffic Engineering.....	62
4.1.2.5. Generalized MPLS Signaling	65
4.1.2.6. Traffic Control	66
4.1.2.7. Policy Control	67
4.1.2.8. Quality of Service.....	67
4.1.3. <i>LSP Teardown</i>	67
4.2. SUMMARY	68

CHAPTER 5. QUANTITATIVE ANALYSIS	70
5.1. INTRODUCTION TO NETWORK SIMULATOR NS-2	70
5.2. SIMULATION AIMS AND METHODOLOGY	71
5.3. SIMULATION DETAILS	73
5.4. SIMULATION RESULTS	77
5.4.1. <i>Results with MPLS and DiffServ Disabled</i>	77
5.4.2. <i>Results for MPLS Enabled and DiffServ Disabled</i>	81
5.4.3. <i>Results with MPLS Disabled and DiffServ Enabled</i>	85
5.4.4. <i>Results with MPLS and DiffServ Enabled with a Single ER-LSP</i>	88
5.4.5. <i>Results with MPLS, DiffServ Enabled and with Multiple ERLSPs</i>	91
5.5. SUMMARY	96
CHAPTER 6. CONCLUSION AND FUTURE WORK.....	97
6.1. SUMMARY	97
6.2. CONCLUSIONS	97
6.3. FUTURE WORK	98
REFERENCES	99
APPENDIX. GLOSSARY	106
VITA.....	107

Table of Figures

Figure 2.1. MPLS domain	10
Figure 2.2. MPLS shim header format	11
Figure 2.3. Ingress LSR or LER functionality.....	12
Figure 2.4. Downstream on-demand label assignment and LSP creation.....	14
Figure 2.5. Components of a QoS architecture.....	16
Figure 3.1. MPLS-based QoS architecture.....	23
Figure 3.2. QoS mechanisms.....	24
Figure 3.3. Signaling architecture.....	32
Figure 3.4. Inter-MME signaling.....	39
Figure 4.1. Reference network.....	55
Figure 4.2. Soft-state refresh performance	59
Figure 5.1. “Dumbbell” topology used in the experiments [MNS].....	72
Figure 5.2. Performance with no UDP traffic (no MPLS, no DiffServ).....	78
Figure 5.3. Performance with medium level of UDP traffic (no MPLS, no DiffServ).....	78
Figure 5.4. Performance with high level of UDP traffic (no MPLS, no DiffServ).....	79
Figure 5.5. Performance with very high level of UDP traffic (no MPLS, no DiffServ).....	79
Figure 5.6. Performance with huge level of UDP traffic (no MPLS, no DiffServ).....	79
Figure 5.7. Total TCP throughput at the sink for different levels of UDP traffic (no MPLS, no DiffServ).....	81
Figure 5.8. Performance with no UDP traffic (MPLS, no DiffServ).....	82
Figure 5.9. Performance with medium level of UDP traffic (MPLS, no DiffServ).....	82
Figure 5.10. Performance with high level of UDP traffic (MPLS, no DiffServ).....	83
Figure 5.11. Performance with very high level of UDP traffic (MPLS, no DiffServ).....	83
Figure 5.12. Performance with huge level of UDP traffic (MPLS, no DiffServ).....	83
Figure 5.13. Total TCP throughput at sink for different levels of UDP traffic (MPLS, no DiffServ).....	84
Figure 5.14. Total TCP throughput at sink for different levels of UDP traffic (DiffServ, no MPLS).....	87
Figure 5.15. Performance with no UDP traffic (MPLS, DiffServ, single LSP).....	88
Figure 5.16. Performance with medium level of UDP traffic (MPLS, DiffServ, single LSP).....	89
Figure 5.17. Performance with high level of UDP traffic (MPLS, DiffServ, single LSP).....	89
Figure 5.18. Performance with very high level of UDP traffic (MPLS, DiffServ, single LSP).....	89
Figure 5.19. Performance with huge level of UDP traffic (MPLS, DiffServ, single LSP).....	90
Figure 5.20. Total TCP throughput at sink for different levels of UDP traffic.....	91
Figure 5.21. Performance with no UDP traffic (MPLS, DiffServ, per-flow LSP).....	92
Figure 5.22. Performance with medium level of UDP traffic (MPLS, DiffServ, multiple LSPs).....	93
Figure 5.23. Performance with high level of UDP traffic (MPLS, DiffServ, multiple LSPs).....	93
Figure 5.24. Performance with very high level of UDP traffic (MPLS, DiffServ, multiple LSPs).....	93
Figure 5.25. Performance with huge level of UDP traffic (MPLS, DiffServ, multiple LSPs).....	94
Figure 5.26. Total TCP bandwidth at the sink (MPLS, DiffServ, per-flow LSP).....	95

List of Tables

Table 2.1. Conventional Routing Architecture	12
Table 2.2. Label Switching Architecture	12
Table 3.1. Differences Between E-LSPs and L-LSPs [Bruc2000]	29
Table 3.2. ATM Service Mapping [Bile2000]	32
Table 3.3. ATM-Diffserv Mapping [Chuc2000]	33
Table 3.4. Comparison of Scheduling Disciplines [Hui1991]	42
Table 3.5. Comparison of QoS Provisioning	53
Table 3.6. Comparison of QoS Control	53
Table 3.7. Comparison of QoS Management	54
Table 4.1. LSP Setup [Data2000]	57
Table 4.3. Traffic Engineering Extensions	62
Table 4.4. Working Path Identification	65
Table 4.5. Comparison of the RSVP-TE and CR-LDP Signaling Protocols	68
Table 5.1. Criteria for Selecting an Evaluation Technique	74
Table 5.1. Statistics for Throughput (no MPLS, no Diffserv)	80
Table 5.2. Statistics for Throughput (MPLS, no Diffserv)	84
Table 5.3. Diffserv Conditioner Statistics (Diffserv, no MPLS)	86
Table 5.4. Statistics for Throughput (Diffserv, no MPLS)	86
Table 5.4. Diffserv Conditioner Statistics (MPLS, Diffserv, single LSP)	90
Table 5.5. Statistics for Throughput (MPLS, Diffserv, single LSP)	90
Table 5.6. Diffserv Conditioner Statistics (MPLS, Diffserv, per-flow LSP)	92
Table 5.7. Statistics for Throughput (MPLS, Diffserv, per-flow LSP)	94
Table 5.8. Traffic Characterization (MPLS, no Diffserv)	95
Table 5.9. Traffic Characterization (MPLS, Diffserv, single LSP)	96
Table 5.10. Traffic Characterization (MPLS, Diffserv, multiple LSPs)	96

Acknowledgements

I would like to thank Dr. Scott F. Midkiff, my advisor for his guidance and encouragement throughout my research. This work would not have been possible but for his support. I also would like to thank Dr. Srinidhi Varadarajan and Dr. Sallie Henry, my committee members for their helpful suggestions. I would like to thank Dr. Joseph Tront of Electrical Engineering department for allowing me to use his computer resources for this research.

Chapter 1. Introduction

Traditional quality of service (QoS) architectures are generic and all encompassing and do not deal with the peculiarities of specific link-layer mechanisms. These architectures do not take into account the differences in QoS requirements at the internet service provider (ISP)-level access networks and backbone networks. These differences in QoS requirements arise due to differences in the volume of traffic handled at ISP and backbone networks. This resulted in generic QoS architectures that are not customized to the requirements.

These traditional generic QoS architectures are either very strict in their QoS enforcement, like ATM-based architectures, or lenient in their enforcement, like DiffServ-based architectures. These types of architectures present problems because strict enforcement leads to poor scalability due to high state information storage requirements. Lenient enforcement allows ill-behaved flows to enter the core of the network and cause network resource over-utilization and loss of revenue to ISPs, among other such issues.

This motivates the need for a single, new QoS architecture to handle heterogeneity in networks, that offers flexibility in its handling of different volumes of traffic at different parts of the network, and is customizable. In addition to this, the architecture should leverage the benefits found in current QoS architectures.

To achieve these goals, a multi-protocol label switching (MPLS)-based QoS architecture is proposed in this thesis. This architecture leverages the benefits of ATM and DiffServ-based architectures and has management elements that can be used to customize the architecture for a particular domain. MPLS provides heterogeneity since it can work with different link-layer mechanisms. This architecture also provides both strict and lenient QoS enforcement at different parts of the network, thus being scalable and fulfilling the requirements at the same time. MPLS, through its signaling protocols, acts as glue in this architecture. MPLS signaling protocols, such as RSVP-TE and CR-LDP, are the key behind the customizability and flexibility of this architecture.

Deliverables from this research include details of the QoS architecture, results from simulation experiments done to demonstrate the claims in this architecture, and the conclusions that can be drawn from this architecture and the analysis.

This thesis is divided into six chapters. Chapter 2 gives background information, dealing with MPLS and QoS. Chapter 3 provides the details of the proposed architecture and compares the architecture with other well-known architectures. Chapter 4 provides a qualitative analysis of the two main signaling protocols used in this architecture. Chapter 5 provides simulation results that demonstrate the effectiveness of the MPLS and DiffServ combination in the backbone networks. Chapter 6 presents conclusions that are drawn from the qualitative analysis of the architecture and quantitative simulation results. This is followed by a brief description of the opportunities for further research.

Chapter 2. Background

This chapter provides a basis for understanding the theory behind Multiprotocol Label Switching (MPLS) and Quality of Service (QoS). In particular, it gives an overview of MPLS technology, its architecture, operational details, advantages, and applications. This is followed by an introduction to QoS that includes components of a generic QoS architecture and methods to implement and measure QoS.

2.1. Multiprotocol Label Switching

2.1.1. Introduction

According to Law, “there are mainly two kinds of computer networks that comprise the Internet, Internet Protocol (IP)-based networks and Asynchronous Transfer Mode (ATM)-based networks” [Law2000]. Performance and traffic management capabilities are the key advantages of ATM-based networks, while scalability and flexibility are the key advantages of IP-based networks. Neither type of network offers the full advantages of the other type of network. The Internet Engineering Task Force (IETF) standardized the Multiprotocol Label Switching (MPLS) [Eric2000] architecture to merge the strengths of these two types of networks into one standards-based alternative.

MPLS introduces a new connection-oriented forwarding paradigm, based on fixed-length labels. This fixed-length switching concept is similar, but not the same as followed in ATM and frame relay networks [Nata2000]. MPLS is currently applied to IP-based networks. The rest of this thesis focuses on IP-based networks and how MPLS can work with such networks. According to the seven-layer OSI reference model, MPLS resides somewhere between the data link layer and the network layer. MPLS introduces the concept of connection-oriented mechanisms in IP-based networks that are basically connectionless. It also adds new methods for traffic engineering and traffic management in these networks. The circuit-switching or virtual circuit model, such as that used in ATM, possess advantages like bandwidth reservation, performance management, and traffic management. MPLS provides IP networks with such advantages of the circuit-switching model in addition to the scalability and flexibility merits that are already present in IP-based networks.

2.1.2. Operation of MPLS

The basis of MPLS operation is the classification and identification of IP packets with a short, fixed-length, and locally significant identifier called a label and forwarding the packets to a switch or router that is modified to operate with such labels. The modified routers and switches use only these labels to switch or forward the packets through the network and do not use the network layer addresses.

The routers, which assign such labels to the packets, are called Label Edge Routers (LERs) and the modified routers and switches that use these labels to forward traffic are

called Label Switch Routers (LSRs). A Label Switched Path (LSP) is the particular path that a packet or flow traverses through the network based on the labels assigned to that packet or flow. The MPLS domain is a portion of a network that contains devices that understand MPLS.

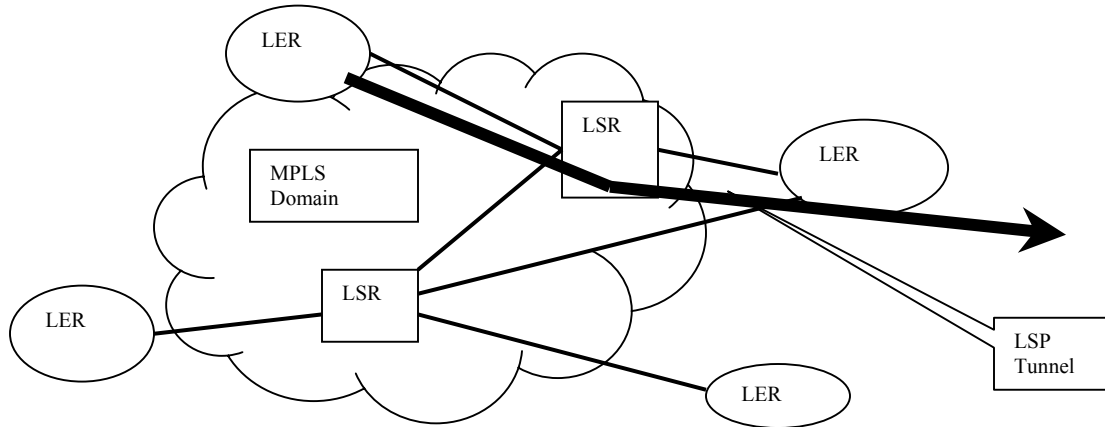


Figure 2.1. MPLS domain.

Traditional routing at the network layer can be divided into two basic functions, control and forwarding. The forwarding component is responsible for the actual forwarding of the packets from an input port to an output port across a switch or a router. This forwarding action uses both the forwarding table in the router and the information carried in the packet. The control component, consisting of one or more routing protocols and algorithms, is responsible for the construction and maintenance of the forwarding table.

Conventional store-and-forward routing is based on network reachability information. As a packet traverses the network, each router uses the network layer header in the packet to get all the necessary forwarding information. This information is used as an index into a routing table that specifies the packet's next hop. This process is repeated at each router in the path through the network. Thus, the optimal forwarding of a packet is determined again and again by the intermediate routers. To get maximum forwarding performance, only the destination address is studied by the intermediate routers and other information like IP precedence and VPN membership information are usually not considered [Cisco2000a]. MPLS achieves efficiency and speed, by using the concept of "labels" to eliminate this oft repeated process. Additionally, it achieves other benefits, since labels can summarize information not limited to just the destination reachability information. The MPLS architecture also contains the control and forwarding component, but here, these two are completely separated so that each component can be independently developed and modified.

2.1.2.1. Label Switch Router Forwarding Component

The algorithm used by the label switching forwarding component to make a forwarding decision is based on the forwarding table and a label that is carried with the packet. Thus, the key idea of MPLS is to include a label in each packet. A label is a fixed-length

structure that is attached to every packet that comes through the LER. According to Bruce, “the label does not directly encode any information from the network layer header like source or destination addresses” [Bruc2000]. The label summarizes essential information about the packet. This might include destination, precedence, VPN membership, QoS information from the Resource Reservation Protocol (RSVP), and even the entire route for the packet, as chosen by the ingress LER based on administrative policies [Cisco2000a].

According to Bruce, “the LSR forwarding table consists of a sequence of entries, where each entry contains an incoming label and one or more subentries. Each subentry consists of an outgoing label, outgoing interface, and the next hop address. More than one subentry corresponds to multicast forwarding” [Bruc2000]. An LSR can maintain only one forwarding table for the router or one forwarding table per router interface. For link layer protocols that cannot carry a label within their packets, an MPLS “shim” header, as shown in Figure 2.2, can be used.

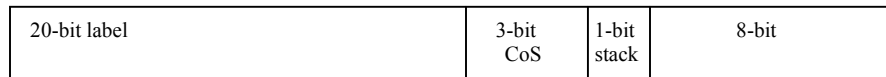


Figure 2.2. MPLS shim header format.

The 32-bit MPLS header has the following fields. The label field (20 bits) carries the actual value of the MPLS label. The class of service (CoS) field can affect the queuing and discard algorithms applied to the labeled packet along the network. The one bit stack field is used to identify the end of stack when multiple labels are used. Finally, the time-to-live (TTL) field provides conventional IP time-to-live functionality.

The algorithm used by the forwarding component is called the label swapping algorithm. When an intermediate LSR gets a packet, “the router extracts the label from the packet and, for each subentry of the matched entry in the forwarding table, the router replaces the label in the packet with the outgoing label (mentioned in the subentry) and sends it to the next hop specified by the subentry”[Bruc2000]. MPLS provides a highly flexible mapping of IP flows to LSPs based on the concept of forward equivalence classes (FECs). FECs are “subsets of packets that are sent to the same next hop, even if the packets within the subset differ in their network layer header information”[Bruc2000]. In MPLS, the assignment of a particular packet to a FEC is done just once, as the packet enters the MPLS domain [Eric2000]. The label represents this classification. This ensures that once a packet is assigned to a FEC, represented by a label, subsequent routers do no further header analysis. Many IP flows can be mapped to the same FEC that can be further merged by the intermediate LSRs.

Thus, the forwarding component uses a single forwarding algorithm in contrast to conventional network layer routing. Tables 2.1 and 2.2 illustrate the differences [Bruc2000].

Table 2.1. Conventional Routing Architecture

Routing Function	Unicast routing	Unicast routing with types of service	Multicast routing
Forwarding algorithm	Longest match on destination address	Longest match on destination address and exact match on type of service	Longest match on source address and exact match on source address, destination address, and incoming interface

Table 2.2. Label Switching Architecture

Routing Function	Unicast Routing	Unicast routing with type of service	Multicast routing
Forwarding algorithm	Common forwarding (label swapping)		

2.1.2.2. Label Switch Router Control Component

The LSR control component has methods and procedures to distribute routing information among LSRs and to construct and maintain forwarding tables. The routing information distribution can be done by existing routing protocols like open shortest path first (OSPF) and the routing information protocol (RIP). The construction of forwarding table needs label creation and distribution of the label associations to other LSRs, table creation at each intermediate router, label switched path creation, label insertion/table lookup, and data transfer using the created label switched path [Web2001].

The MPLS architecture [Eric2000] suggests many ways to create and distribute label-FEC associations to other LSRs. Depending on the aims of the MPLS deployment being considered, these methods vary in their use. For example, labels are requested by ingress or intermediate LSRs and are determined by the downstream or egress LSRs. This is followed by upstream label distribution. This process is called downstream on demand label assignment and distribution. This is useful in setting up explicitly routed label switched paths (ER-LSP). Figure 2.3 shows the ingress LSR functionality [Chuc2000].

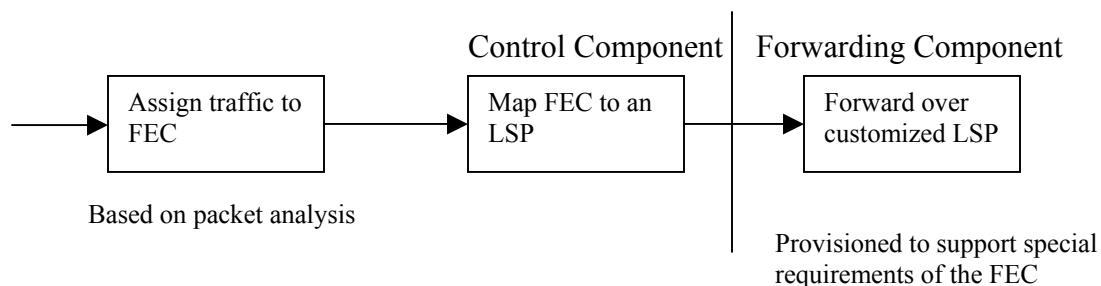


Figure 2.3. Ingress LSR or LER functionality.

Label switched paths can be classified into hop-by-hop routed LSPs and explicitly routed LSPs. In hop-by-hop routed LSPs, each LSR determines the next interface to route the LSP based on the network layer routing topology database. This method of setting up a hop-by-hop routed LSP is called LSP setup by independent control. In explicitly routed LSPs, the complete route for the LSP is specified in the setup messages itself and the route is installed in the intermediate nodes using MPLS signaling protocols. All the nodes along the ER-LSP follow the route specification that is carried by the setup message and will use the route as a basis for label request, label mapping, and data transfer. This method of setting up an explicitly routed label switched path is called LSP setup by ordered control. The ordered control method gives network service providers more control over how to map the network traffic to the underlying network topology. MPLS can also support the concept of label stacking with an unlimited number of labels in the stack. Label stacking can be used for VPN support and MPLS-based fault-tolerance [Eric2001]. The downstream assigned on-demand label distribution method is considered here to explain the five stages of a LSP creation. The signaling protocol used is called the label distribution protocol (LDP). LDP is a reliable signaling protocol that uses TCP. Apart from label distribution, it is also used to negotiate traffic-related characteristics and capabilities of the participating MPLS LSRs. The LSP creation proceeds as follows.

The ingress LSR, also called the label edge router (LER) may have an unlabeled packet traversing through it. In this case, it will use the traditional routing table to find the next hop and will issue a label request towards the next hop. This behavior is called the on-demand mode. The label request message propagates through the network and each intermediate router will receive a label from its downstream router. This is done by the label distribution protocol (LDP). If the path should take into account certain constraints like available bandwidth, QoS guarantees, and administrative policies, then the constraint routed label distribution protocol (CR-LDP) or the Resource Reservation Protocol with traffic engineering extensions (RSVP-TE) signaling protocols can be used. Each LSR then creates entries in its local label information base (LIB). The contents of this table specify the label-FEC associations and each entry in this table contains a mapping between the input port and input label to the output port and output label. These entries are updated whenever renegotiation of label bindings occurs. The ingress LSR then pushes the obtained label on top of the IP packet and forwards the packet to the next hop. Each subsequent LSR examines *only* the label in the received packet, replaces it with the outgoing label present in the label information base (LIB) and forwards the packet to the network through the specified port. This phase uses the incoming label map (ILM) and next-hop label forwarding entry (NHLFE) modules in the MPLS architecture. If the incoming packet is not labeled, the FEC-to-NHLFE (FTN) map module is used. When the packet reaches the egress LSR, the label is stripped from the packet and is delivered using the traditional network layer routing module. If the egress LSR is not capable of handling MPLS traffic, the penultimate hop popping method is used. In this method, the LSR whose next hop is the egress LSR, will handle the label stripping process instead of the egress LSR [Eric2000].

Figure 2.4 illustrates the steps above using an example. Assume that LER B is directly connected to the 30.0.x.x network (i.e., the network with IP network identifier 30.0) and that LER A is connected to the 30.4.x.x network. In this configuration, LER D directly connects to the 30.1.x.x network. The routing tables in Figure 2.4 show only the related entries. Label assignment and distribution is done to make sure that there is a common understanding among the intermediate routers about the intent of the labels.

In the example, the LER B assigns labels 35 and 48 to networks 30.0.x.x and 30.1.x.x, respectively. These labels are placed in the incoming label column of the routing and forwarding table. This is to ensure that the labels distributed are the ones that any LSR that is in direct contact with B will apply to the packets that are sent to B. For example, LSR C should apply label 48 to all the packets, destined for the 30.1.x.x network and which uses B as the next-hop.

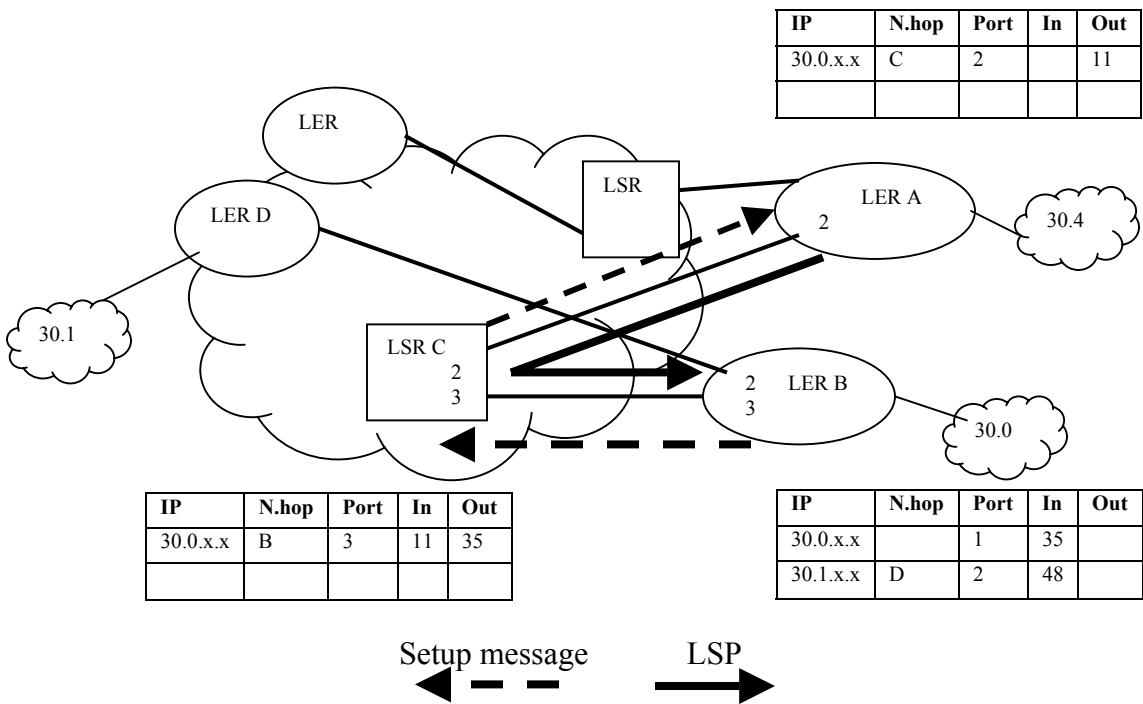


Figure 2.4. Downstream on-demand label assignment and LSP creation.

LSR C then gets the labels that were forwarded to it by B and uses them to configure its label table. For example, it places label 35 as the outgoing label for packets that are destined for the 30.0.x.x network. LSR C then assigns label 11 for packets destined to 30.0.x.x and sends it to its neighbors. LER A then gets the label bindings and places 11 in the outgoing label entry for the 30.0.x.x network. Now, packets meant for the 30.0.x.x network from LER A will traverse the network with a label of 11 through port 2 to LSR C. LSR C will replace the label in the packet with label 35 and send the packet out through port 3 toward the next hop, LER B. LER B, the egress LSR, then strips the label from the IP packet and performs traditional Layer 3 (IP) routing on the packet.

2.2. Network Quality of Service

2.2.1. Introduction

Quality of service, with respect to computer networks is a set of service requirements to be met by the network while transporting some network traffic flow. Here, the flow represents a packet stream from source to destination that can be unicast or multi-cast, with associated quality of service requirements [RFC2386]. Network QoS is a measurable level of service delivered to network users, characterized by quality of service parameters. Service providers provide this service to the customers for a cost. According to an article from *Cisco Systems*, “edge routers perform packet or flow classification, admission control, and configuration management while the backbone routers perform congestion management and congestion avoidance”[Cisco1999].

2.2.2. History and the Need for Quality of Service

In IP-based networks, the only service model traditionally provided was best-effort service. This service does not support reliability. It performs little error control and no retransmission on error. This type of service requires higher-layer protocols, like the Transmission Control Protocol (TCP) to provide reliability and error-control. ATM-based networks came into prominence due to their ability to provide hard QoS guarantees that IP-based networks could not offer. Traditionally, without network quality of service, the solution to congestion problems in the network was to over-provision bandwidth. This is not always feasible for Internet service providers (ISPs) and other business due to the cost and effort involved and is certainly inefficient from an economic perspective. New Internet services like voice over IP (VoIP) and multimedia services, such as streaming and web-casting, require QoS guarantees for their operation.

2.2.3. Quality of Service Architecture

The following functions comprise a generic network quality of service architecture.

- Methods to request and receive levels of service through service level agreements (SLAs). An SLA is the service level request format, consisting of quality of service (QoS) parameters like bandwidth, jitter, and latency. This agreement is a formal service contract between the customers and the service providers.
- Signaling, buffer allocation, and management to allow a network to grant the requested service level. RSVP is an example of a signaling protocol that could co-ordinate reservation of resources.
- Control of applications that deviate from their set levels, called policing and marking. This is generally preceded by traffic profile determination.
- Methods to arrange for the traffic flow to traverse portions of the network in a manner that would guarantee contracted service levels using QoS-based routing.

- Congestion avoidance, congestion management, queuing, and scheduling methods to prevent network conditions from causing detrimental effects to the service level.
- Methods to allow administrative control policies to take effect through policy-based networking.
- Accounting and billing for the service.

Figure 2.5 shows the components of a reference QoS architecture [Armi2000]. These components are explained in detail below.

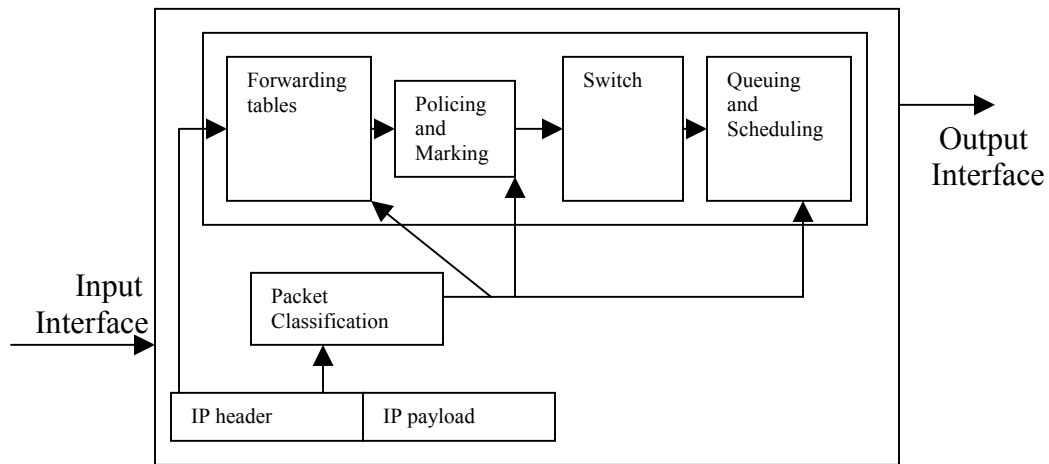


Figure 2.5. Components of a QoS architecture.

2.2.3.1. Quality of Service Metrics

To implement a service level in a network for a customer, service requirements have to be expressed in some measurable QoS metrics. According to Chen, “there are three types of metrics, additive, multiplicative, and concave” [Chen99]. QoS metrics may also differ based on the type of networks. While delay, reliability and cost are some of the parameters for IP-based networks [Chen99], cell loss ratio, cell delay variation, and maximum cell transfer delay are examples of QoS parameters for ATM networks [Chen99]. For wireless networks, jitter, bandwidth, noise, and fading are some of the deciding QoS metrics [Chen99]. Metrics also differ based on the layer being considered. For example, frame error rate is a link-layer metric while fading and the resulting bit error rate (BER) are physical layer metrics.

2.2.3.2. Service Level Agreements

Service level agreements in general, include aspects of network provision and details of acceptable service level. An SLA is a contract between the network user and the service provider. Negotiation between the customer and the provider are the key behind successful SLAs. Committed information rate, maximum round trip latency, total downtime, percentage of successfully delivered frames, and minimum time to full

restoration are some sample parameters that can be found in an SLA [Para2001]. ISPs need SLAs so that they can configure their network to handle the incoming traffic. The customers need SLAs so that their applications can receive expected levels of service. There are two types of SLAs, static SLAs and dynamic SLAs [Xipe1999].

2.2.3.3. Traffic Classification

Traffic classification is used by network elements to classify the traffic flowing through them into different priority levels or service classes. This will guarantee service levels to users and their applications based on their SLAs. After classification, other traffic handling policies like policing and marking can be applied to the flow. Classification is also done to enforce any specified administrative policy to a particular flow or traffic. Traffic classification can be done by utilizing link layer, network layer, transport, or application, or other upper layer headers and is meant to achieve a certain granularity of control. Classification can also be done using the packet's payload. In practice, higher granularity means longer delay at the routers due to the delay in packet processing.

2.2.3.4. Congestion Management

According to an article from *Cisco Systems*, “congestion management allows the network elements to control congestion by determining the order in which packets are transmitted out of an interface based on priorities or service levels assigned to those packets” [Cisco2000]. It involves queue creation, assignment of packets to queues and scheduling of the packets in the queue. Congestion management is not a preventive mechanism, but a reactive mechanism that is called for only when congestion conditions develop in the network. Queue management is central to this mechanism and some examples of queuing policies are first-in-first-out (FIFO) queuing, weighted fair queuing (WFQ), custom queuing, and priority queuing [Cisco2000].

2.2.3.5. Congestion Avoidance

Congestion avoidance is done to avoid congestion from building up in networks. This is generally realized using packet drops. Tail Drop and Random Early Detection (RED) [Sall1993] are some examples of techniques used for congestion avoidance. The RED algorithm utilizes the congestion reaction features of TCP and is highly suited to TCP/IP networks. TCP responds to the traffic drop by slowing the transmission rate. To utilize this feature, the RED algorithm drops packets randomly even before congestion develops. Thus, RED lowers traffic in the network caused by that source. There are many variations of RED like Weighted RED (WRED) and RED In/Out (RIO) [Armi2000]. The thresholds beyond which RED will start dropping packets are configured by the service provider and can be used to favor traffic associated with premium service levels.

2.2.3.6. Policing and Marking

Policing and marking are related to traffic regulation. Each traffic class has a certain limit on how fast the packets may arrive or a limit on the number of acceptable packets within a certain time interval. Policing and marking are related actions taken by a router when the packets are determined to be out-of-profile from that stipulated in the service level agreement. Policing acts in a way to drop the out-of-profile packets while, marking just identifies and marks the out-of-profile packets. These marked packets are handled appropriately in the queuing and scheduling stage [Armi2000]. Such marked packets are the first ones to be dropped during periods of congestion. Both of these components share a metering module that determines whether the packets are in profile or out of profile.

2.2.3.7. Shaping

Traffic shaping by the network elements is achieved by modifying the traffic characteristics to conform to the contracted traffic descriptors in the SLA. Some algorithms for traffic shaping include reverse leaky bucket and spacing. The main idea is not to determine conformance, but to schedule packet departures so that the flow conforms to traffic specifications. Non-destructive ways of handling congestion include explicit congestion notification (ECN) mechanisms like backward ECN (BECN) and forward ECN (FECN). Here the out-of-profile packets are not dropped but the source of the flow experiencing congestion is notified of the congestion and rate of flow can be decreased.

2.2.4. Quality of Service Signaling

Quality of service signaling is a method used by an end station or a network element to notify its neighbors and request special handling of certain traffic. It is useful for coordinating traffic handling techniques like shaping, policing, and marking. Signaling is important to configure uniform, successful end-to-end QoS service handling across the network. True end-to-end QoS requires that all the network elements, like switches, routers, and firewalls appropriately support QoS. The coordination of these components is done by QoS signaling [Cisco2000].

2.2.5. Service Models

The IETF has defined different IP QoS and CoS architectures. The most prominent among them are the integrated services QoS architecture and the differentiated services QoS architecture.

2.2.5.1. Integrated Services

The integrated service model (IntServ) [RFC 1633] is based on a goal to augment the best-effort service model traditionally provided by connection-less IP networks. There are two types of services defined under the integrated services model. The IntServ

guaranteed service (GS) [RFC2212] was formulated to support real-time applications that have stringent bandwidth and latency requirements. The IntServ controlled load service (CL) [RFC2211] was formulated to support traditional applications, whose main criterion is to achieve a certain level of performance under any network condition. IntServ is an architecture requiring per-flow traffic handling at every hop along an application's end-to-end path and explicit signaling of each flow's requirements using a signaling protocol like RSVP. IntServ suffers from lack of scalability due to the scalability problems with the standard RSVP signaling protocol [Armi2000]. As a result, the differentiated services architecture was proposed.

2.2.5.2. Differentiated Services

Differentiated Services (DiffServ) is based on an architecture [RFC 2475] that pushes complex decision making to the edges. This results in less processing load on core routers and, thus, faster operation, due to less signal state processing and storage. According to this architecture, a differentiated services code point (DSCP) is carried in every packet. This is carried in the old IP type of service (ToS) field. Classification, rate shaping, and policing are done at the edge routers and packets are mapped onto service levels. Per-hop queuing and scheduling behaviors, or simply per-hop behaviors (PHBs), are defined through which a number of edge-to-edge services might be built. The expedited forwarding (EF) PHB [RFC 2598] requests every router along the path to always service EF packets at least as fast as the rate at which EF packets arrive. This entails rate shaping and expedited packet servicing at the routers. As a consequence EF is used to achieve low-loss, low-latency, and low-jitter edge-to-edge services. The assured forwarding (AF) PHB [RFC 2597] supports more flexible and dynamic sharing of network resources by supporting soft bandwidth and loss guarantees appropriate for bursty traffic. According to Francis, "at each transit node, the DSCP is used to select the per-hop-behavior (PHB) that determines the scheduling treatment and drop probability of each packet" [Fran2001].

2.2.6. Policy-based Routing

Policy-based routing (PBR) goes beyond traditional routing protocols and implements packet forwarding and routing according to a service provider or enterprise's requirements and needs. Using policy-based routing, network service providers can implement policies that selectively cause packets from different customers to take different paths.

2.3. Summary

This chapter discussed the theory behind MPLS and network quality of service. The next chapter discusses motivations for a new architecture based on MPLS and the subsequent chapters discuss the architecture itself followed by a qualitative and quantitative analysis of the two main MPLS signaling protocols used in the architecture.

Chapter 3. MPLS-based QoS Architecture

This chapter details the MPLS-based QoS architecture proposed in this research. A reference heterogeneous network is outlined for the purpose of explaining the motivation behind the proposed architecture. The architecture is described based on the reference framework proposed in Cristina, *et al.*, [Cris1998]. This is followed by an analysis of how this architecture compares to other well-known QoS architectures.

3.1. Background

This chapter describes the MPLS-based QoS architecture with a focus on how MPLS-signaling helps achieve conformance to the requested levels of service. MPLS signaling achieves this through interaction with policy-servers, traffic engineering, and network management components.

This chapter does not provide a detailed description of the algorithms used for policing, shaping, metering and queuing, description of technical contents of the SLAs and SLSs, billing and accounting functions, authentication and security procedures of the QoS architecture, QoS measurement, and QoS specific network management protocols. According to Natalie [Nata2000] and Armitage [Armi2000], the most commonly found algorithms for QoS control purposes are leaky bucket, virtual scheduling, dual leaky bucket, dual virtual scheduling, reverse leaky bucket, and spacing algorithms. The technical details of the SLAs, SLSs, and billing, accounting specifications can be obtained from the commercial vendors or from the Qbone architecture documentation [Qbone2001a]. QoS measurement before and after QoS provisioning in the networks is left for further study. The QoS measurement architecture is expanded, as necessary from the Qbone measurement architecture [Qbone2001a].

3.2. Motivation

Consider the heterogeneous network of Figure 3.1 consisting of an LMDS-based fixed wireless access network, with an ATM-based fiber backbone connecting to the Internet. In the near future, it is expected that these networks is IP-based and that the ATM-based components in such networks is gradually phased out. To satisfy the QoS and traffic engineering needs of IP-based networks, in the absence of ATM-based overlay components, MPLS is considered here as an alternative.

The LMDS-based access network is based on the existing fully operational LMDS network at Virginia Tech. The system is described as follows [Lmds2001][VT2001].

The system is a point-to-multipoint system with a central hub servicing remote locations around the hub. The hub is located on Slusher Tower, a twelve-story dormitory building on campus. There are two remote locations served at the time of this writing. The hub connects the remote sites to the campus network using both fiber and ethernet. A remote site serves an off-campus administrative office and a student television-learning studio. The second remote location links a local

service provider to campus to provide student access to the campus network. The current backbone network used at Virginia Tech, to connect to the Internet, is ATM-based.

Future backbones can be all-optical, such as the optical backbone used by CA*net 3 [Canet2001].

The issues and expectations from such an MPLS-based alternative for the reference network are as follows.

- Interoperability between the copper-based, wireless, and optical domains.
- Quality of service guarantees for different types of applications with different source and sink domains.
- Intra-area, inter-area, intra-Autonomous System (intra-AS), and inter-AS traffic engineering along with management of network resources.
- Network resource provisioning in different domains.
- Security of data transfer from source to destination.
- Loop handling and prevention in paths that go through different domains.
- Scalability of the network design to efficiently handle future multimedia traffic.
- Congestion and flow control mechanisms to aid QoS guarantees.
- Legacy network support.
- Multicast support to aid applications such as videoconferencing and distance learning.
- Virtual private networks to enable distributed sites for a single connection to be able to communicate securely through the Internet.
- Fault-tolerant schemes for robust handling of important traffic.
- Network and administrative management for heterogeneous networks along with accounting and billing with customized solutions to cater to the needs of different network customers.

The following section discusses how the proposed MPLS-based QoS architecture could resolve the issues and meet expectations for the system.

3.3. Details of the MPLS-based QoS Architecture

In Figure 3.1, the access network (AN) can be wired or wireless. Wired networks could be either copper-based or optical. Wireless networks use electromagnetic waves in free space. The wireless networks could use either radio frequency (RF) or infrared (IR). The wireless networks could be inherently fixed wireless or mobility-enabled wireless networks. This thesis focuses on wireless networks such as LMDS that are fixed wireless networks with RF waves as the means of achieving wireless communication. The access network could also be a copper-based network like Ethernet, gigabit Ethernet, or token ring. An optical-based access network is also possible. For this architecture, MPLS is proposed as the single IP-based control plane.

The MPLS-based IP control plane is meant to solve the problems arising out of the heterogeneity in the networks. End-to-end path setup consists of two phases, construction of paths inside a network area, i.e., intra-area and the construction of paths that connect network areas together, i.e., inter-area. To help construct these paths, the MPLS management entity (MME) is introduced. The MME consists of a traffic engineering server and bandwidth broker elements. The traffic engineering server (TES) is a per-area server used in setup, policy, and administrative management of inter-area and intra-area MPLS LSPs. Depending on the size of the area, a hierarchy of TESs can be considered. These servers have direct or indirect links with the LERs and the LSRs of a particular area. The possible configurations of TES and LSR interaction are:

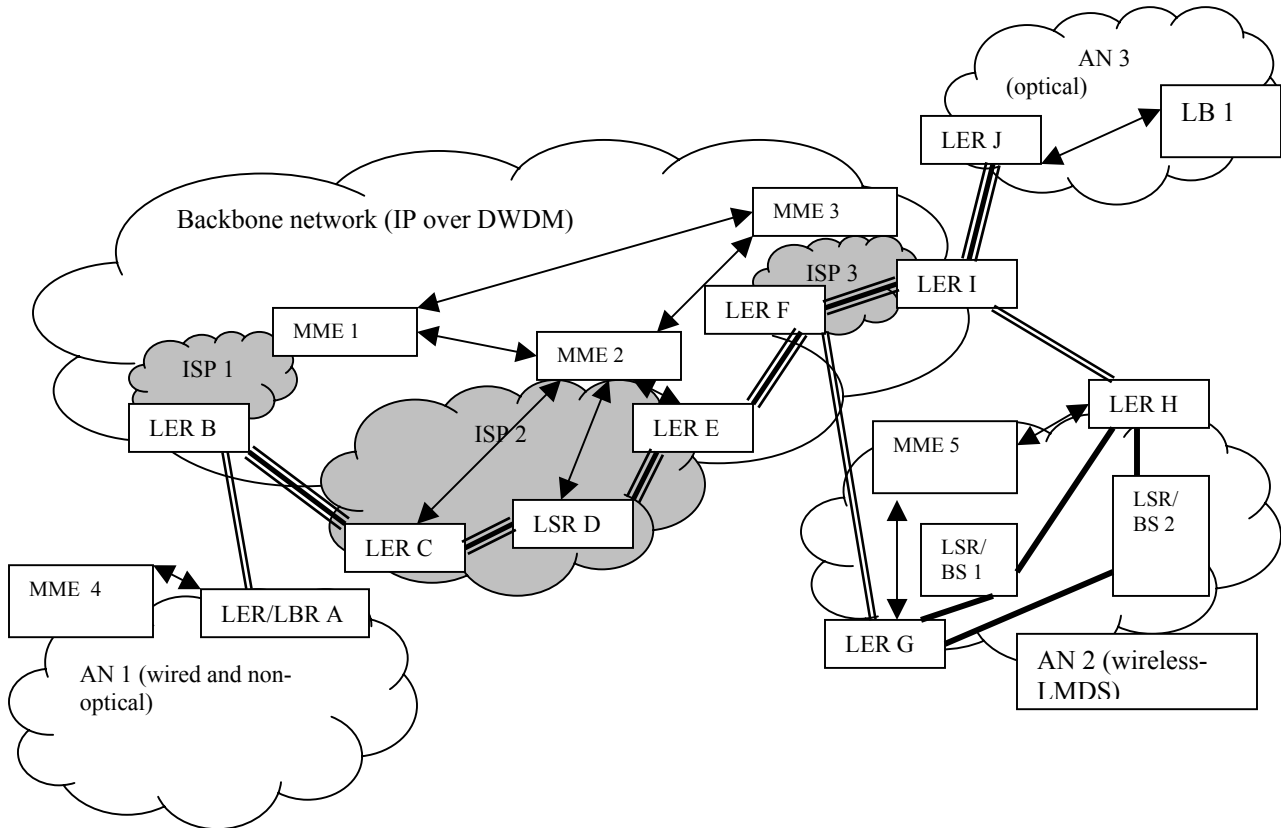
- TES or hierarchy of TESs connecting to all the LERs and LSRs in the domain, or
- TES or hierarchy of TESs connecting to only the LERs of the domain.

For simplicity, this architecture connects all the LERs and LSRs directly to the TESs. The intra-domain policy request, and decision exchanges between the LERs and the TESs are based on the common open policy service (COPS) protocol [RFC2748] and its variation, the COPS-PR protocol [Cops2001]. The interaction between the LERs and LSRs uses the MPLS signaling protocols based on RSVP-TE and CR-LDP. Inter-area LSP setup is based on the simple inter-domain bandwidth broker signaling (SIBBS) protocol followed in the Qbone architecture [Qbone2001b]. The inter-AS and/or inter-domain LSP setup is also based on the variation of the SIBBS protocol.

The bandwidth broker component of the TES is meant for admission control and network QoS resource management. This bandwidth broker could be built based on an architecture that is domain-specific or domain non-specific. In this architecture, a domain-specific architecture is preferred. The bandwidth brokers must be domain-specific to take into consideration, the peculiarities and differences between the different network domains of a heterogeneous network. For example, resource provisioning for optical networks and that for wireless networks need two different approaches that could not be satisfied by a single architecture.

In this QoS architecture, a multi-level QoS mapping with an ATM-like QoS mapping in the periphery or edge of the ISP networks and a DiffServ-like QoS mapping in the backbone or core of the networks is envisioned. The basic idea behind this two-level architecture is to enforce strict entry criteria at the periphery and allow only conformant flows inside the backbone networks. This gives rise to easily manageable, flexible, and highly scalable flows in the backbone networks.

Thus, the important features of this architecture include the two-level QoS mapping, special handling of the heterogeneous network through an MPLS-based control plane, inter-area LSP setup, traffic-engineering enhanced network management elements, and efficient resolution of the issues and expectations as outlined in the previous section. The details are provided below.



LER	Label Edge Router	BB	Bandwidth Broker
AN	Access Network	MME	MPLS Management Entity (BB+TES)
LSR	Label Switch Router	LBR	Label Border Router
LB	Lambda Broker	BS	Base Station
≡≡≡	MPLS tunnel between ISPs	≡≡≡	Traffic between ANs and ISPs
—	Traffic inside a domain	↔	control messages

Figure 3.1. MPLS-based QoS architecture.

3.3.1. Architecture Details

Cristina [Cris1998] discusses a generalized QoS framework for distributed multimedia applications operating over multimedia networks with QoS guarantees. According to Cristina, “the framework discusses QoS principles that govern the construction of the QoS framework, QoS specification that govern application-level QoS requirements, and QoS mechanisms that realize the desired application end-to-end QoS behavior” [Cris1998]. This paper uses this framework to compare the contemporary QoS architectures and give a qualitative comparison. This generalized QoS framework is used to describe the architecture proposed in this thesis. The framework is also used to compare the proposed architecture against different QoS architectures.

Figure 3.2 shows the different components of the QoS mechanisms formulated for the framework.

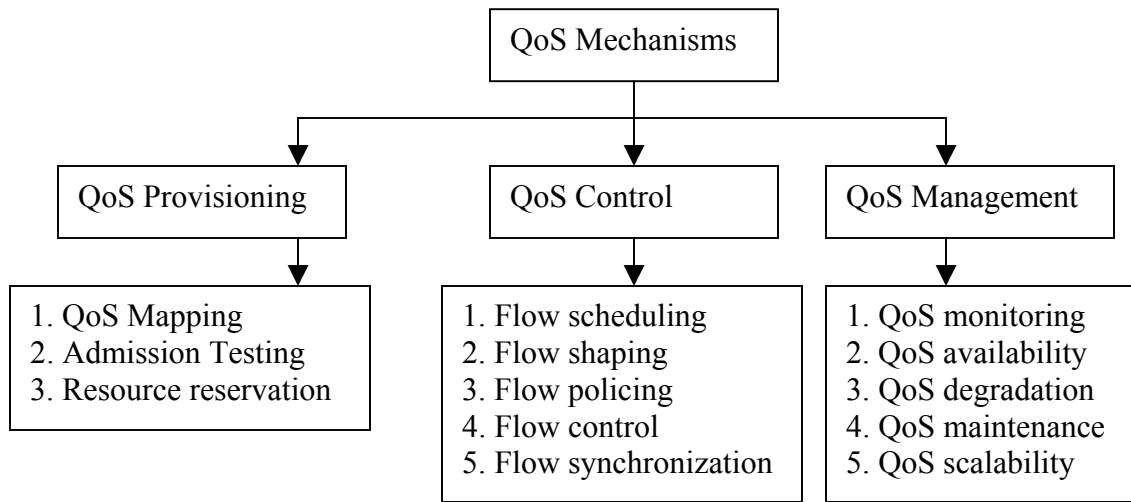


Figure 3.2. QoS mechanisms.

According to Cristina, the different QoS mechanisms include “QoS provision that deals with flow establishment and end-to-end QoS re-negotiation phases, QoS control that deals with the control behavior like policing and shaping at the router elements, and QoS management that deals with management issues like re-negotiation and actions on QoS degradation” [Cris1998]. The QoS provision mechanisms include the QoS mapping, QoS admission testing, and the resource reservation modules.

The QoS mapping module “performs the translation functions between QoS representations at different system levels, to relieve the user of thinking in terms of lower level specification”[Cris1998]. This means that a user can specify QoS in terms of abstract parameters like average and peak bandwidth, jitter, loss, and delay requirements and an entity will translate them into service classes for resource allocation purposes. For example, if the parameter values require high bandwidth and low delay with real time requirements, it can be mapped to the rt-VBR class of the ATM service class definition.

The QoS admission testing module “is responsible for checking availability of resources against the requirements and decide on whether to allow the new request or not” [Cris1998]. Since an end-to-end QoS guarantee needs global resource availability, the local resources are reserved on a successful admission control decision, but they are committed only if the end-to-end admission control test is successful.

The QoS signaling and resource reservation module “arrange for the allocation of suitable end-system and network resources according to the user QoS specification” [Cris1998]. The QoS admission module needs the services of this module to confirm whether or not the global end-to-end admission control test was successful. The QoS mapping module must take into consideration the capabilities of the signaling protocols before mapping the user-specified QoS specifications to network QoS. In this architecture, MPLS signaling protocols RSVP-TE and CR-LDP are both capable of resource reservation. The resource reservation protocol interacts with QoS-based routing for initial path setup and

for QoS mapping and admission control at each local resource handler to setup an end-to-end path with guaranteed bandwidth.

The QoS control component “operates on a time scale at or close to media-transfer speeds. This component provides real-time traffic control based on the requested QoS levels from the QoS provisioning phase” [Cris1998]. This component is not considered in detail in the architecture proposed in this thesis, but the applicability of certain well-known QoS control algorithms is discussed. This component includes the following modules [Cris1998].

- The flow scheduling module “manages the forwarding of flows in an integrated manner in both the end-system and network.” Flows are generally independently scheduled at end systems while being aggregated and jointly scheduled inside the network.
- The flow shaping module “regulates flows based on the user-requested QoS levels. It has been mathematically proven that the combination of traffic shaping at the edge of the network and scheduling in the network can provide hard performance guarantees.”
- The flow policing module “is appropriate when user traffic crosses administrative boundaries and needs to be closely monitored. Policing is used to observe whether the QoS contracted by a provider is being maintained or not.”
- The flow control module includes open-loop and closed-loop control methods that are directly applicable to non-adaptive and adaptive flows, respectively. The former needs deterministic guarantees while the latter can adapt to fluctuations in the available resources.
- The flow synchronization module “is required to control the event ordering and precise timings of multimedia interactions.”

The QoS management component is “to maintain the agreed levels of QoS” [Cris1998]. This component includes the following modules [Cris1998].

- The QoS monitoring module “is meant to track the ongoing QoS levels being provisioned and being used.” This gives an idea to the managing entity about the performance optimizations to be realized.
- The QoS availability module allows the application to specify monitoring and feedback of certain QoS parameters to be informed of the delivered performance.
- The QoS maintenance module “allows for the comparison of the monitored QoS against the expected performance and exerts optimizations to match the level of service.” This also takes care of link and node failures and handles fault tolerance and recovery mechanisms. To maintain the provided QoS, traffic engineering measures may need to be applied.
- The QoS scalability module “includes measures to enhance the scalability of the architecture. Multicast applications need scalability to satisfy the receiver-centric bandwidth requirements.” This module includes mechanisms like QoS filtering, aggregation, and hierarchical QoS structure.

3.3.1.1. QoS Mapping and Intra-domain Signaling

As introduced in Section 3.3, a two level QoS mapping is considered in this architecture with CR-LDP as the signaling protocol in the periphery of the network and RSVP-TE as the signaling protocol in the backbone networks. The choice of the signaling protocols is justified in the following section.

3.3.1.1.1. Motivation

There are significantly different QoS requirements at the boundary of the networks and at the backbone or core of the current Internet [RFC2990]. Thus, a single QoS architecture based on ATM, DiffServ [QBone2001a], or IntServ [RFC2990] does not satisfy the current QoS requirements [RFC2990]. Hence, there is a need for an architecture based on multiple levels of QoS mappings.

The reasons for stipulating strict entry criteria in the periphery of the networks are to allow only conformant flows, to police, shape and buffer non-conformant flows to make them conformant, and to protect a flow from being deprived of its resources. This mapping is also intended to ensure that the conformant flows allowed inside the backbone networks can be easily mapped onto coarser QoS mappings in the backbone.

The motivations behind coarse QoS mapping in the network backbone include:

- scalability to ensure that the potential large number of flows are supported without performance degradation,
- ease of management of conforming flows,
- limits to the number of QoS-mapped pre-configured MPLS LSPs,
- support for a reliable and fault-tolerant MPLS-LSP strategy due to fewer QoS classes, and
- reduction of per-flow state maintenance at the intermediate routers leading to better scalability and performance.

Depending on the deployment scenarios of the signaling protocols there could be five signaling architectures available for study.

1. CR-LDP as the one and only signaling protocol in both the periphery and the backbone network.
2. RSVP-TE as the one and only signaling protocol in both the periphery and the backbone network.
3. CR-LDP as the signaling protocol in the backbone network and RSVP-TE as the signaling protocol in the network periphery.
4. RSVP-TE as the signaling protocol in the backbone network and CR-LDP as the signaling protocol in the periphery.
5. Disjoint signaling domains of RSVP-TE and CR-LDP with an integrating mechanism for end-to-end QoS guarantee.

Options 1 and 2 above are probably preferable since a single protocol eases management, they provide a flexible single architecture, implementation is easier, deployment of variations is easier, and there are no inter-operability issues. Unfortunately, the current market trends with a set of router vendors using RSVP-TE and another set of router vendors using CR-LDP [Tele2000], makes options 1 and 2 not viable. Options 4 and 5 seem to rise up as potential solutions. There are many disadvantages with option 5. Options 3 and 4 are controlled variations of option 5. This architecture is based on the premise that option 4 is the best possible solution. The choice of option 4 is justified in the following sections.

RSVP has been recognized to be an important reservation protocol because of its characteristics and market support from major backbone router vendors. However, RSVP has also been recognized to have certain scalability problems due to the overhead associated with the soft-state approach, the potential presence of a large number of flows that will add an undue overhead to the soft-state approach, and the absence of reliability that can be a problem if the tear or confirmation messages are lost in the network.

3.3.1.1.2. RSVP-TE for DiffServ QoS Mapping

RSVP [RFC2205] is used to service resource reservation requests. These requests dictate the level of resources like bandwidth and buffer space that must be reserved along with scheduling behaviors to be installed at the intermediate routers for flows that require end-to-end QoS commitment [Paul1997].

A flow descriptor consisting of flowspec and filterspec forms a RSVP reservation request path message. The flowspec describes the required QoS and is meant for a router's scheduler component. The flowspec [RFC2205] includes service class and two sets of numeric parameters. The numeric parameters are Rspec, or reserve specification, that defines the required QoS and Tspec, or traffic specification, that describes the data flow. The filterspec defines the set of data packets that should receive a specific QoS treatment and is used by a router's classifier component. Flows that do not match all the classifiers are handled as best effort traffic. The Tspec parameters include peak rate of flow in bytes per second, bucket depth in bytes, token bucket rate in bytes per second, minimum policed unit in bytes, and maximum datagram size in bytes [Paul1997]. The Rspec parameters include bandwidth in bytes per second, and slack term in milliseconds.

RSVP uses a combination of destination address, transport layer protocol type, and destination port number to identify an RSVP session. It is important to understand that RSVP is not a routing protocol, but just a reservation protocol that needs a routing protocol underneath it to identify the path to be taken by the RSVP messages. The primary roles of the RSVP path message are to install reverse routing state in each router along the path and to provide receivers with information about the characteristics of the sender traffic and end-to-end path for resource reservation. The primary role of the RSVP resv message is to carry reservation requests to the intermediate routers along the reverse path.

The RSVP adspec object is an optional object that the sender may include in the RSVP path message to convey the characteristics of the end-to-end communications path. This information gives the receivers an idea of the level of the required reservation to achieve end-to-end QoS. If the sender sends an adspec along with the RSVP path message, it is called one pass with advertising (OPWA) and it is called one pass (OP) without the adspec. In this architecture, OPWA is assumed.

In summary, RSVP includes the following features [RFC2205].

- RSVP can make reservations for both unicast and many-to-many multicast flows.
- RSVP makes reservations only for unidirectional flows but RSVP-TE has been extended to perform bi-directional path setup.
- RSVP is receiver oriented.
- RSVP maintains soft-state in routers and hosts, providing support for dynamic membership changes and adaptation to routing changes.
- RSVP transports and maintains traffic control and policy control parameters.
- RSVP has several reservation styles to fit a variety of applications.
- RSVP provides transparent operation through non RSVP-capable routers.
- RSVP supports both IPv4 and IPv6.

RSVP-TE is an extension to RSVP to establish label-switched paths (LSPs) in MPLS [Dani2000]. RSVP-TE in its present version can be used to instantiate explicitly-routed LSPs with or without resource reservations, smooth rerouting of LSPs, preemption, and loop detection. Presently, RSVP-TE can be used only to setup unicast LSPs, but not multicast LSPs. The set of packets that are assigned the same label effectively defines the RSVP flow. RSVP-TE currently supports establishing LSP tunnels with or without QoS requirements, dynamic rerouting of an established LSP tunnel, observation of the actual route traversed by the established tunnel, identification and diagnosis of LSP tunnels, preemption or replacement of an established LSP tunnel for administrative reasons, downstream-on-demand label allocation, distribution, and binding, and support of controlled-load IntServ service and class-of-service IntServ service.

MPLS with DiffServ mapping in the network core is mutually beneficial to both MPLS and DiffServ components. MPLS provides DiffServ with path protection and restoration while DiffServ acts as a CoS architecture for MPLS. This is important since MPLS does not possess any inherent CoS or QoS mapping on top of it [Bruc2000]. MPLS is path-oriented and it can potentially provide faster and more predictable protection and restoration capabilities in the face of topology changes compared to conventional hop-by-hop routed IP systems [Fran2001]. MPLS with DiffServ can give network designers the flexibility to provide differential treatment to certain QoS classes that need path-protection.

This type of two-level mapping is scalable since RSVP, with its scalability problems, is used only to set up and maintain a potentially smaller number of MPLS LSPs, while the traffic through the LSPs is actually managed by the scalable DiffServ CoS architecture.

To ensure DiffServ functionality over MPLS networks, the DSCP-marked packets must get the corresponding QoS treatment at each LSR. This is important since LSRs do not see the IP header and DSCPs are encoded in the type of service (TOS) field of the IPv4 header. It is also necessary to decide whether the DiffServ over MPLS network needs to use less than or more than eight per hop behaviors (PHBs) since this will affect the way MPLS carries the DiffServ packets.

The MPLS shim header has a three-bit field called the Exp field that is meant for experimental use. Francis suggests a way to use this field for supporting fewer than eight PHBs [Fran2001]. The limitation of eight PHBs is due to the difference in sizes of the DSCP (6 bits) and Exp (3 bits) fields. Thus, though the DiffServ standards allow for up to 64 different DSCPs, MPLS can support only eight different DSCPs using the Exp field. This particular way of using the LSP with Exp bits representing the DSCPs is called the Exp-inferred-PSC LSP (E-LSP). Just as a conventional DiffServ router maintains a mapping between the DSCP values and the PHBs it supports, an LSR must maintain a mapping from Exp values to PHBs. For this E-LSP setup, no additional signaling is needed. According to Bruce, “the label tells an intermediate LSR where to forward a packet and the Exp bits tell it what PHB to treat the packet with”[Bruc2000]. Whenever there is a need to provide more than eight PHBs or to use DiffServ with MPLS for links that do not support shim headers like ATM, E-LSP is not used. Francis’ solution is to use the label field itself as the information carrier about different PHBs [Fran2001]. This requires the label distribution protocols to be enhanced to support this extended feature. This type of LSP that uses the labels to support DiffServ functionality is called the label-only-inferred-PSC LSPs (L-LSPs). With L-LSPs, the Exp bits or the congestion loss priority ATM CLP bit is still needed to convey the drop preference. Table 3.1 details the differences between E-LSPs and L-LSPs.

Table 3.1. Differences Between E-LSPs and L-LSPs [Bruc2000]

E-LSPs	L-LSPs
PHB is determined from Exp bits	PHB is determined from label or from label and Exp/CLP bits
No additional signaling is required	PHP or PSC is signaled at LSP setup
Exp-PHB mapping is configured	Label – PHB mapping is signaled. Exp/CLP – PHB mapping is well known (used only for AF)
Shim header is required	Shim or link layer header may be used
Up to eight PHBs per LSP	One PHB per LSP except for AF and PSC per LSP for AF
With bandwidth reservation, the bandwidth is shared by set of transported PSCs	With bandwidth reservation, the bandwidth is per-PSC

The label forwarding with DiffServ LSRs consists of four stages [Fran2001]:

- incoming PHB determination,
- outgoing PHB determination with optional traffic conditioning,
- label forwarding, and
- encoding of DiffServ information into encapsulation layers like Exp and CLP.

A labeled packet can arrive at a DiffServ ingress router with multiple levels of label stack entries. The decision to choose that label stack entry for incoming PHB determination stage depends on the DiffServ tunneling mode. The tunneling modes identified are the pipe model, short pipe model, and uniform models [Fran2001]. The pipe model is mandatory while the short pipe model, a variant of the pipe model, and the uniform model are optional. For the purpose of label forwarding in DiffServ over MPLS purposes, a DiffServ context for a label must be maintained in the incoming label map (ILM) for each incoming label [Eric2000]. The DiffServ context is comprised of the following [Fran2001] information.

- LSP type (E-LSP or L-LSP).
- Supported PHBs.
- Encapsulation to PHB mapping for an incoming label.
- Set of PHB to encapsulation mappings for an outgoing label.

3.3.1.1.3. CR-LDP for QoS Mapping

LDP [RFC3036] and CR-LDP [Bile2000] specify the label distribution protocol and the constraint-based label distribution protocol, respectively. MPLS LSRs set up LSPs between them to agree on the meaning of the labels, used to forward traffic between and through them. LDP is the protocol that allows one LSR to inform another LSR of the label bindings it made. CR-LDP is used to distribute labels, set up, and maintains LSPs along explicitly routed or constraint-routed paths [RFC3036] [Bile2000]. According to [RFC3036], “LDP associates a FEC with each LSP and the LSPs are extended through the network as each LSR splices incoming labels for a FEC to the outgoing label assigned to the next hop for the given FEC.”

There are four categories of LDP messages [RFC3036].

- Discovery messages, used to announce and maintain the presence of an LSR in the network and to enable LSP peers to set up connections for label distribution.
- Session messages, used to establish, maintain, and terminate sessions between LDP peers.
- Advertisement messages, used to create, change, and delete label mappings for FECs.
- Notification messages used to provide administrative and error signaling.

The discovery messages are carried in UDP packets with the “all routers in this subnet” multicast address. This message is sent periodically. When an LSR learns the address of another LSR, it establishes a TCP connection. An indirect discovery mechanism is also possible when the LSR generates a unicast discovery message to a UDP port of a specific IP address. The session, advertisement, and notification messages use TCP for reliability

and in-order delivery. All LDP messages have a common structure that uses a type-length-value (TLV) encoding scheme. The value of a TLV may itself encode other TLVs. Thus, functionality can be added to LDP in the future. CR-LDP is one such functionality extension. According to Bilel, “constraint-based routing offers the opportunity to extend the information used to setup paths beyond that is available for the routing protocol.” CR-LDP extends LDP by adding the following functionality [Bile2000].

- Strict and loose explicit routing, encoded as a series of ER-hops contained in a constraint-based Route TLV.
- Traffic parameters like peak rate, committed rate, and service granularity can be specified in the Traffic Parameters TLV.
- Route pinning feature, that prevents the LSP from changing its path even when a better path becomes available at some hop along the loosely source routed portion of the LSP.
- If a route with sufficient resources cannot be found, existing paths can be rerouted to reallocate resources to the new path by a process called pre-emption. This is supported in CR-LDP by the concept of setup and holding priorities.
- Failure handling by re-routing, is supported here through different mechanisms.
- LSPID, to identify the LSP that had been set up.
- Resource class feature that specifies the class of resources that can be explicitly included or excluded from the CR-LSP path. For example, if the CR-LSP path has a choice of satellite or terrestrial path, the resource class feature can be used to include or exclude the high delay satellite path.

The Traffic Parameters TLV along with LSPID TLV must be specified in a CR-LDP label request message for the QoS mapping procedures to work. The Traffic Parameters TLV contains fields for flags, frequency, weight, peak data rate (PDR), peak burst size (PBS), committed data rate (CDR), committed burst size (CBS), and excess burst size (EBS). The flags field has one bit for each of the traffic parameters. Each bit, if set represents a negotiable traffic parameter and, if reset, means that the concerned traffic parameter cannot be negotiated. The frequency field defines the granularity of the provided CDR. The weight field determines the CR-LSP’s relative share of the possible excess bandwidth above its committed rate. The values PDR and CDR are in units of bytes per second while PBS, CBS, and EBS are in units of bytes.

The CR-LDP label request message is first sent by the ingress LSR. This is processed by the intermediate LSRs for traffic parameter values. The flags and weight fields values are altered, if needed. The intermediate LSRs also reserve the requested resources in its domain. The CR-LDP label mapping message is generated by the egress LSR on receiving the label request message. If the flags allow, the traffic parameters and weight must be sent back in the label mapping message and any LSR should adjust the resources it reserved if the mapping message contains differing requirements. Table 3.2 shows how CR-LDP traffic parameters can be used to provide ATM-like mapping and the associated traffic conditioning algorithms to be applied. In this way, ATM-like traffic management can be realized using CR-LDP. This capability allows strict control over the traffic that

can be tagged as conformant traffic and, hence, is well suited for the periphery of the networks.

Table 3.2. ATM Service Mapping [Bile2000]

Service Examples	PDR	PBS	CDR	CBS	EBS	Service Frequency	Conditioning Action
Best Effort	Infinite	Infinite	Infinite	Infinite	0	Unspecified	-
ATM-CBR	PCR	CDVT	=PCR	=CDVT	0	Very frequent	Drop if > PCR
ATM-rt-VBR	PCR	CDVT	SCR	MBS	0	Frequent	Drop > PCR Mark > SCR,MBS
ATM nrt-VBR	PCR	CDVT	SCR	MBS	0	Unspecified	Drop > PCR Mark > SCR,MBS
ATM-UBR	PCR	CDVT	-	-	0	Unspecified	Drop>PCR
ATM-GFR.1	PCR	CDVT	MCR	MBS	0	Unspecified	Drop > PCR
Delay Sensitive	User	User	=PDR	=PBS	0	Frequent	Drop > PDR
Throughput Sensitive	User	User	User	User	0	Unspecified	Drop > PDR, PBS Mark > CDR,CBS

The most important issue with such a multi-level QoS mapping is conversion of signaling protocol messages between component domains. This issue is addressed in this architecture through MPLS-based and non-MPLS based solutions like MPLS LSP hierarchy, MPLS LSP aggregation, MPLS LSP tunneling, inter-MME interaction, adding functionality to edge routers to handle both RSVP-TE and CR-LDP messages, and conversion of the representation of QoS parameters between the signaling protocols. The proposed signaling architecture is as shown in Figure 3.3.

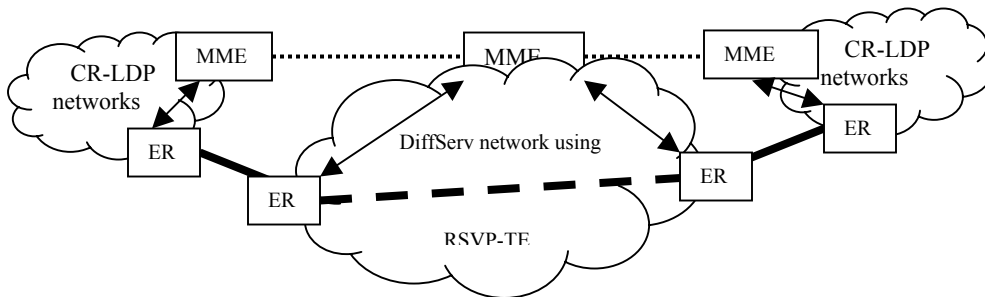


Figure 3.3. Signaling architecture.

The signaling protocol message conversion can be done either through an explicit interaction between the RSVP-TE and CR-LDP signaling protocols or through an implicit interaction between them. The type of application for which MPLS is used determines the proper choice between these options. For the purposes of QoS

applications and traffic engineering, the implicit option seems to be a better choice. The implicit option ensures that the CR-LDP-based networks at the periphery do not know the presence of RSVP-TE in the core of the networks. For the periphery, the RSVP-TE core is hidden and represents only an extremely reliable backbone link.

Through LSP tunneling, the LSPs established between the LERs of the RSVP-TE core network carry CR-LDP messages without going through any protocol conversion at the LERs [RFC3031]. The RSVP-TE core network gets abstracted as a reliable link in the end-to-end path created by CR-LDP, with the MMEs acting as the bandwidth broker and traffic engineering server. The reliability of the core is important since the CR-LDP network should not know about the existence of RSVP-TE network. If a link or node did fail in the core network, the LER or the MME of the core backbone network needs to generate a CR-LDP specific error message back to the source. The failure needs to be handled either by MPLS-based solutions like rerouting, fault-tolerance, and path-protection inside the RSVP-TE core itself or by link-level fault tolerance, e.g., using SONET rerouting and a dual-ring topology.

The MMEs of different domains interact with one another to provision bandwidth across the RSVP-TE capable core and can generate a CR-LDP error message. The RSVP-TE based network core in this architecture is generally fully meshed with pre-constructed CR-LSPs [Chuc2000]. The MMEs and LERs of the RSVP-TE core maintain a mapping table to convert between the QoS mapping in the periphery and in the network core. Table 3.3 shows how the mapping could be done.

Table 3.3. ATM-Diffserv Mapping [Chuc2000]

ATM Service Category	Diff-Serv PHB
CBR	EF
rt-VBR	AF priority 1
nrt-VBR	AF priority 2,CS
	AF priority 3
ABR, GFR, UBR, W-UBR	AF priority 4

In this architecture, the fine-grained ATM-like QoS traffic is tunneled through an LSP built through the core. These LSPs are built with a coarse-grained QoS based on DiffServ. The MMEs take care of assigning these LSPs to carry the fine-grained QoS traffic using the conversion table. It is important to understand that the QoS levels of the traffic through the core are not changed. The network packets with differing QoS levels are just tunneled through with no changes to their headers or settings by intermediate LSRs in the RSVP-TE core. If the QoS parameter values of the traffic through the core are explicitly changed, the problem of restoring the original parameter values in the traffic at the exit domains is difficult to solve. This is alleviated in this architecture through the use of LSP tunneling and LSP aggregation. Use of explicitly routed LSP tunnels [RFC3031] is considered in this architecture. This tunnel construction could also be setup by constraint-based routing.

Using the explicitly routed LSP tunnel or constraint-routed LSP tunnel, the CR-LDP signaling and data traffic packets can be carried through the backbone networks without the intermediate core routers knowing the type of traffic being carried. The path setup in this architecture proceeds as described in the following.

The end-to-end bandwidth guaranteed LSPs for traffic engineering and quality of service purposes start and end at the ISP networks where the customer traffic first merges with traffic from other customers. The maximum diameter of such an end-to-end LSP network that represents the span of the ISP and backbone networks must be chosen by a tradeoff between the two following options.

- The LSPs should start as close as possible to the customer's network to avoid the service differentiation problems described by Pravin [Prav1999], thus aiming for LSP networks with larger diameter.
- The span of the LSP network should be small to enable it to scale to the size of Internet.

Traffic engineering is generally done in the core of the network where network conditions are less dynamic and the network topology is fully meshed. Spanning out from the core, the ISP networks in this architecture form MPLS LSPs that connect to the fully meshed core like a hub and spoke design. The hub represents the network core that is always connected. The spokes represent the dynamically established LSPs in the ISP networks.

A host in a CR-LDP periphery network trying to set up an end-end MPLS path through a RSVP-TE core network should start with identifying and possibly marking the traffic. This can also be done with the help of the domain's MME and leaf edge router. The host then contacts its domain MME. This MME contacts the ISP's MME or the LER with MME functionality and conveys the resource requirements. This LER then tries to set up a CR-LDP path with resources as requested after checking for conformance to SLA agreements and local resources availability.

When a CR-LDP message tries to enter a RSVP-TE based core network, the MME or the LER/MME will look up the traffic engineering database to reserve the required resources and forward the signaling traffic as tunneled high priority traffic through the RSVP-TE core. The tunneling of CR-LDP signaling traffic is handled as follows.

1. The backbone MPLS networks is set up for DiffServ QoS mapping using either E-LSPs or L-LSPs with bandwidth guarantees. The signaling traffic should be forwarded with higher priority than data traffic.
2. The MME corresponding to the backbone ingress should be capable of understanding CR-LDP messages. It also needs to know to generate CR-LDP notification messages to the source in case of unavoidable errors in the RSVP-TE networks. The MME, after recognizing it as a CR-LDP label request signaling message, should push a label that takes it to the egress LSR that can deliver it towards the destination of the signaling packet. The Exp bits of this tunnel packet are set such that it receives a high priority treatment at all intermediate routers. The DiffServ tunneling mode should be such that the tunneled packet is not

- opened or altered. All the PHB scheduling and drop preferences are based on the outermost label stack entry.
3. The destination network's MME will decide whether or not to allow the traffic and sends a CR-LDP label mapping message or a notification message back to the source. This signaling message is let through the RSVP-TE core in a similar way as the CR-LDP request messages. But, the ingress LSR/MME of the RSVP-TE core that let the label request message through will note the details regarding the traffic that is going to go through this LSP. This will help setup the right DiffServ mapping modes at the ingress LSR of the RSVP-TE core networks for future data traffic. Thus, a CR-LDP mapping message with CBR-like traffic mapping will have its label mapping noted and an associated EF DiffServ forwarding class is reserved.
 4. When the conformant traffic arrives at the ingress LSR of the RSVP-TE core, the database is consulted and the DSCP meant for that traffic is mapped to the Exp bits and/or label value and the traffic is tunneled through the network after pushing a label onto the already labeled traffic. The PHB meant for that type of traffic is applied to the traffic and is switched through the MPLS RSVP-TE core network using the concept of LSP tunneling. Penultimate hop popping [RFC 3031] is done at the penultimate LSR of the tunnel to ensure that the traffic is given proper treatment again at the egress of the RSVP-TE networks and then on to the CR-LDP island.

The main drawback of the implicit option is that the ends of the traffic need to be CR-LDP islands. But, this is conformant with the design of ATM-like QoS mapping in the periphery with DiffServ-like QoS mapping in the backbone of the networks. This drawback can be eliminated by the explicit option. The explicit option provides methods of interoperation between the two protocols [Greg2000]. This research discusses RSVP-TE in the periphery of the networks and cites CR-LDP as the preferred signaling protocol in the backbone. There is an explicit interaction between the two protocols in the form of a specialized LSR called inter-working LSR (IWLSR) that can "speak" both CR-LDP and RSVP-TE. The IWLSR converts messages between RSVP-TE and its CR-LDP equivalents. The complexity of implementation and conversion between these two protocols is the main drawback behind this method. Simplicity, scalability, and ease of conversion are the main advantages of the implicit conversion option.

3.3.1.2. Admission Testing

The admission testing module is responsible for comparing QoS resource requests to the available resources in the domain and deciding whether or not to allow a new request. In this architecture, this function is handled by the MPLS management element (MME) that consists of the domain-specific bandwidth broker (DBB) and traffic engineering server (TES) modules. To achieve scalability, the MME should follow the style of the current DNS implementation [RFC 1034] with a distributed database implemented as a hierarchy of many MME servers. A centralized scheme is not followed due to single point of failure, need for high-volume traffic handling, and maintenance problems.

The domain-specific bandwidth broker component is active in the end user networks. This component maintains up-to-date information about the resources available in the domain. The traffic engineering server component offloads the MPLS LER from the following functions, traditionally done by the MPLS LER.

- Traffic engineering functions.
- Constraint-based route calculation functions.
- Path management functions like rerouting, fault-tolerance, and path protection.
- Traffic management functions.
- Path selection, path optimization, and policy-based MPLS path setup.
- Maintenance functions.

This component is also active in the backbone networks, where traffic engineering is better suited and needed. The MPLS routing and traffic engineering server (RATES) [Petr2000] is an example implementation of a traffic engineering server.

According to Petri, *et al.*, “in MPLS networks with bandwidth-guaranteed LSPs, shortest path routing with fixed link metrics can cause LSP setup requests and in certain circumstances traffic to go through the LSP be rejected, even though these requests may have been admissible under a different routing scheme” [Petr2000]. Better network infrastructure utilization while maintaining QoS guarantees is the primary objective of traffic engineering. MPLS can control the path the traffic takes from ingress node to egress and, hence, is suitable for traffic engineering and, hence, better admission control. Traffic engineering solves the problems of congestion caused by uneven traffic mapping onto the physical network. In short, traffic engineering (TE) is a method to balance the network traffic load on the various links, routers, and switches in the network so that the traffic load on these elements do not reach extreme levels. Traffic engineering helps meet the following network requirements [Chuc2000].

- Routing primary paths around known bottlenecks or points of congestion in the network.
- Providing precise control over how traffic is rerouted when the primary path has failures.
- Providing more efficient use of available aggregate bandwidth and long-haul fiber.
- Enhancing the traffic-oriented performance characteristics of the network by minimizing packet loss, minimizing prolonged periods of congestion, and maximizing throughput.
- Enhancing statistically bounded performance characteristics of the network like loss ratio, and delay variation, to help serve multimedia traffic and other delay variation sensitive services in a better way.

MPLS achieves these traffic engineering requirements by supporting the following functions [Chuc2000] [RFC 2702].

- Establish, activate, deactivate, modify, reroute, and teardown of LSPs.
- Configure a strict or loose explicit route for an LSP.
- Support configurations for explicitly including or excluding a set of resources from the physical route of an LSP.
- Prioritize LSP.

- Preempt LSPs.
- Create QoS-enabled paths.
- Achieve per-LSP accounting and statistics control for fine grained traffic management.
- Create backup-LSPs to provide graceful degradation in the event of router or link failures.

In the proposed architecture, the main QoS and traffic engineering metrics considered are bandwidth requirements for an LSP in the form of available bandwidth, reserved bandwidth, effective bandwidth, and administrative policies. Delay, loss, and other metrics that could be present in an SLA can be converted to an effective bandwidth requirement for LSPs. According to Chuck, “a generic MPLS-based traffic engineering solution contains four components, forwarding, distribution, path selection, and signaling” [Chuc2000]. The forwarding component takes care of packet forwarding. MPLS operates as the forwarding component in this architecture with label swapping algorithm as the packet forwarding method. The distribution component takes care of information dissemination about links, their resource information, and their current state to the MMEs. The path selection component uses constrained shortest path first (CSPF) and calculates the shortest path taking into consideration parameters or constraints like capacity, interior gateway protocol (IGP) metrics, matching resource classes, and administrative policies. The output of the CSPF execution is a list of paths with each path defined by a list of intermediate LSRs. The signaling component coordinates communication between the intermediate LSRs identified by the CSPF output and reserves resources depending on the requirements. The intra-domain signaling component for traffic engineering purposes in this architecture is RSVP-TE.

The MME uses the COPS/COPS-PR protocol to communicate with the LER. Thus, in this design, MME acts as a policy decision point (PDP) and the LER as the policy enforcement point (PEP). A traffic engineering solution needs detailed knowledge about the network topology, dynamic network load information, and link attributes. The information distribution component thus needs IGP extensions to carry link attributes in addition to normal link state advertisements. The standard flooding algorithm used by the link-state IGP ensures that the link attributes are distributed to all routers including the MME. The MME is configured so that it acts as an IGP peer and can receive IGP link state updates. Each MME maintains the network link attributes and topology information in a specialized traffic engineering database (TED). This TED is used exclusively for calculating explicit paths for the LSPs.

The path selection component composed of CSPF and TED is used by the MME to calculate the paths for its own set of LSPs across the routing domain. Input to the CSPF algorithm includes the link-state information from the TED, link attributes like reserved bandwidth, and available bandwidth, and administrative attributes like resource class attributes, and policy decisions. The MME will then communicate, through COPS, to the required ingress LER for setting up the LSP. The ingress LER uses the RSVP-TE signaling component to physically set up the path with bandwidth reservations. The ingress LSR then gets back to the MME with the set up results. The domain’s MME

maintains a list of LSPs through the domain and also acts as an admission controller for new setup requests. The MME also acts as a path selection agent to route the traffic, when a labeled packet arrives to be routed along any of the available LSPs. The LSP selection by the MME should be such that, future bandwidth and LSP selection requests are handled in an efficient manner with the least rejection rate.

In present architectures [Chuc2000], each ingress LER maintains the TED and does the CSPF calculation in a distributed way. In the proposed architecture this function is still distributed, but is done by a network of specialized controllers in the form of MMEs. With the network of specialized controllers there is a clean separation between control, i.e., MME and data transfer, i.e., LER functionality. Policy management, traffic engineering management, and network management becomes simpler with fewer control devices to manage. Policy decisions can be quickly and easily changed and/or updated affecting a few servers rather than a larger number of LERs. To make this type of MME architecture scalable and avoid a single point of failure, it is made similar to the DNS hierarchy [RFC 1035]. Using this hierarchical architecture:

- no MME has information about all the LSPs,
- each domain or ISP has a local MME that handles the traffic engineering and /or bandwidth broker requirements for that domain,
- each host knows which bandwidth broker to contact for path setup or resource reservations,
- each MME is configured with its upper level MME, and
- the ISP level MMEs can contact a backbone level MME.

A fully-meshed backbone is constructed by initially defining the LSP bandwidth reservation to be zero. Traffic flows through the LSPs for a certain duration and utilization statistics are obtained. Administrators can then set up the required bandwidth to these LSPs and let the MMEs find the best path between LERs [Eric2001a]. MPLS path protection can be added. The path protection can be either 1:1, 1: n , or m : n . In 1:1 protection, a backup LSP protects each primary MPLS path. This backup LSP is used in case in case of primary LSP failure. In 1: n protection, a single backup LSP protects many (n) primary LSPs. In m : n protection, m backup LSPs protect n primary LSPs with $m < n$ and $m > 1$. The IGP is adjusted to advertise these protection LSPs. Depending on the priority of traffic that is carried in an LSP, that particular LSP could be protected by 1:1, 1: n , or m : n protection mechanisms with 1:1 protection as the highest protection level. The DiffServ code points and PHB mappings are adjusted to make sure that each traffic class is assigned to LSPs based on the class's QoS requirements. An offline CBR and simulation utility can be used to set up these LSPs with bandwidth reservations and other policy decisions. Based on the utilization levels of LSPs and simulation results from offline CBR utilities, optimizations like creation of backup LSPs, bandwidth adjustments, and teardown of LSPs are carried out. Sufficient care should be taken to avoid route oscillations and route flapping in the process [Petr2000].

The inter-MME signaling protocol in this architecture is based on the SIBBS protocol [Eric2001b]. The SIBBS protocol is adapted to suit the requirements of this architecture. The bandwidth broker component of the MME can receive a resource allocation request

(RAR) from either a client in its domain or from another bandwidth broker. The bandwidth broker responds with a resource allocation answer (RAA). The RAR may cause changes to leaf router configurations and ISP-level router configurations to set up traffic conditioning algorithms. Additional RARs may also be generated towards other bandwidth brokers. The RARs may contain space and time co-ordinates of the service, kind of service, and characteristics of the traffic. An inter-domain reservation depends not only on the customer network's SLA, but also on the SLA and SLS between domains. Thus, the reservation can be carried out only if the SLAs and policy requirements of different domains are compatible and can fulfill the user's needs and requirements. According to the Qbone architecture, "every bandwidth broker must track SLSs between its domain and peering domains, set of reservations that have been accepted, and availability of all resources that can be reserved"[Qbone2001b]. In this architecture, the bandwidth broker directly tracks the SLSs. The bandwidth broker also acts as the PDP in this environment with policy settings stored in a database within it. The policy enforcement points (PEPs) are the leaf routers and the domain edge routers. The actual resource use is tracked by the routers and can be monitored by the bandwidth broker on demand.

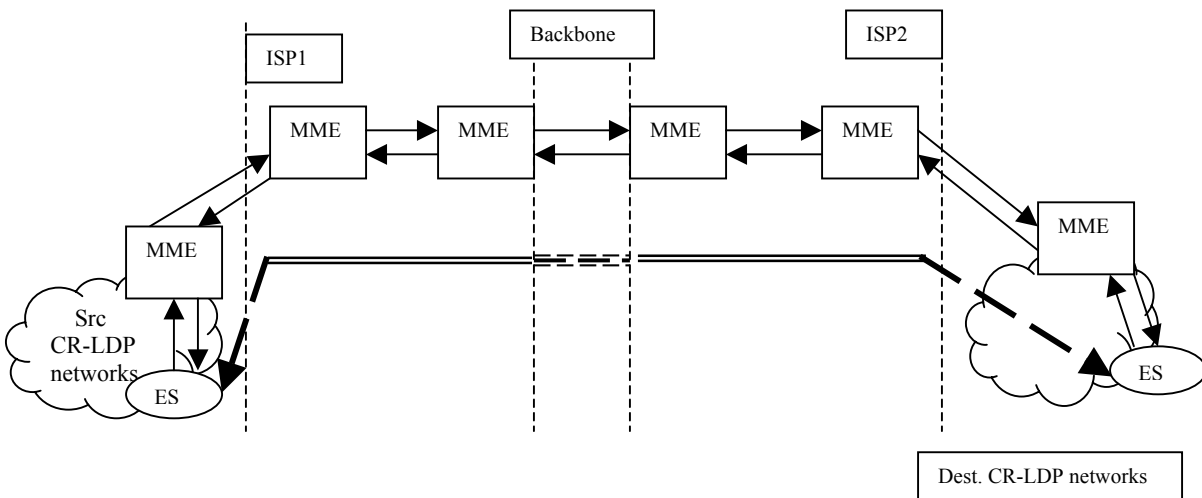


Figure 3.4. Inter-MME signaling.

The MME in the origin domain receives a RAR request from the end system (ES). Note that in this origin domain the BB component of the MME is active while the TES component is inactive. According to its own configuration, the MME can choose to aggregate this request with others. The source domain MME may carry out the following functions [Qbone2001b].

- Authentication to verify whether the ES is from its domain.
- A check if sufficient resources are available in the source domain for supporting such a transaction.
- Identification of the egress router to which the flow may be assigned and the leaf router that can reach ES directly.
- A check of whether the flow fits in to the SLS between itself and the ISP.
- A check of whether the flow may be accepted under the administrative policies.

If the flow can be allowed, the MME authenticates with the domain ISP's MME (IMME) and includes the domain identification along with the flow request specifications. If the MME in the source domain rejects the request for some reasons, it sends a resource allocation answer (RAA) message to the end system with the reason code. It should be noted that the end-to-end MPLS LSP starts at the ISP networks by default. Figure 3.4 shows the inter-MME signaling architecture.

The ISP-level MME first authenticates the request from the source domain's MME. The MME then determines the egress router and outgoing interface from inter-domain tables and with help from exterior gateway protocols (EGPs). It also checks whether the conformance with the SLA/SLS between the source domain and ISP. The MME then checks whether the requested resources fall within the SLS connecting to a successor domain en route to the destination. The policy database is then checked to determine whether the flow can be accepted or not. If all the checks pass, the ISP-level MME then requests the TES component of the MME to set up the LSP for the data transfer.

The TES component first checks whether an LSP is already present that can take up the flow, taking into consideration the QoS requirements, administrative policies, and other user-configured parameters. The TES then sets up packet classification, and traffic conditioning policies at the ingress and egress routers using the COPS protocol. The TES also adjusts the available resources of the selected LSP. The TES then records the LSPID that has been chosen and updates policy related measurement information and adjusts use statistics and free resources. The TES then replies back to the customer's MME with an RAA to confirm the reservations. If an LSP that can take up the flow is not present, the TES server looks up the domain's resources to support the flow and if it can, it forwards the RAR to the next MME in the route. It must be noted that LSPs are not set up by the TES, but only a reservation is made. The LSP setup, in this case, starts only when it gets a positive RAA from other MMEs down the network.

All the transit ISP domains carry out the same type of checks and reservations. In the fully meshed backbone, the backbone MME generally does not do any new path setup. The backbone's MME decides whether it can carry such traffic after careful authentication. It then forwards the RAR to the next ISP domain that has been identified to be in the path to the destination after making modifications to the RAR.

The MME in the destination domain also makes the relevant checks and reservations and sends it to the end system for an RAA. If the destination end system is not specified and only a destination prefix is mentioned, an RAA from the destination domain MME is expected. A positive RAA would traverse back to the source domain through the same path over which the RAR traversed. In the transit domains, the positive RAA should result in resource confirmation. For an LSP that already exists and can support the flow, the per-LSP statistics are updated and the effective bandwidth resources are updated at the MME. The LSPID can be mapped for the flow. If a pre-existing LSP is not available, the ISP domain's MME tries to set up a new intra-domain LSP.

The MME, in trying to set up a new LSP, first carries out optimization calculations so that the resources are efficiently used. In RATES [Petr2000], the management component carries out a minimum interference algorithm that identifies critical links first. These critical links are those paths that need to be avoided to efficiently process future requests. The actual path for the LSP is then constructed using the CSPF algorithm with the constraints of bandwidth and other policy decisions. This explicit path is then notified to the suitable ingress router. The ingress router then uses its signaling component like CR-LDP to set up a path. On the receipt of the label mapping message, the path setup is complete for this domain and this is notified back to the MME using COPS. The ingress and egress routers in the domain are also updated by the MME through COPS for traffic conditioning, and other policy decisions. The MME then modifies the RAA to reflect the positive reservation of resources and forwards it to the preceding MME.

The procedure described above to confirm resources through the setup of LSPs is done at every transit domain. In the backbone domain, the MME, on noting a positive RAA, will configure its ingress and egress routers for PHB conditioning, DSCP marking, and other DiffServ related functions. The source end system receives the RAA and can start data flow.

Note that the data transfer through the backbone is through tunneling, with packets enclosed in a DiffServ header marked with the corresponding DSCPs. This ensures that the backbone is scalable since there is no per-flow state that needs to be maintained in the backbone core LSRs. Per-LSP state alone is maintained in the backbone LSRs. The data transfer through the rest of the system follows the LSP set up through the ISP networks and per-flow state is maintained in the ISP networks.

3.3.1.3. QoS Control

According to Cristina, “the QoS control mechanisms perform real-time traffic control of flows based on requested levels of QoS established during the QoS provisioning phase” [Cris1998]. In this architecture, the QoS control measures are strict at the periphery of the networks while the measures are softer in the DiffServ-based backbone networks. The fundamental QoS control mechanisms are described in the following sections. The QoS control mechanisms for this architecture are based on existing schemes.

3.3.1.3.1. Flow Scheduling

The proposed architecture does not deal with the end system flow scheduling mechanisms. There are several queue service disciplines in use today that could be used for flow scheduling purposes. Hui compares the different rate-based scheduling disciplines that can be implemented at the output queues of switches [Hui1991]. These disciplines include virtual clock, fair queuing, delay-earliest-due-date, jitter-earliest-due-date, stop-and-go, and hierarchical round robin mechanisms. They are compared in the context of connection-oriented packet switching networks. The emphasis had been on

examining their mechanisms to provide delay, delay jitter, and bandwidth guarantees. The results are tabulated in Table 3.4.

Table 3.4. Comparison of Scheduling Disciplines [Hui1991]

	VCL	FQ	D-EDD	SG	HRR	J-EDD
Work conserving	Yes	Yes	Yes	No	No	No
Throughput guarantee	Yes	Yes	Yes	Yes	Yes	Yes
Delay guarantee	No	No	Yes	Yes	Yes	Yes
Jitter guarantee	No	No	No	Yes	No	Yes
Constant buffer	No	No	No	Yes	Yes	Yes
Protection	Yes	Yes	Yes	No	Yes	yes

In Table 3.4, VCL represents the virtual clock algorithm, FQ represents fair queuing, D-EDD represents delay-earliest-due-date, SG is stop and go, HRR is hierarchical round robin, and J-EDD is jitter-earliest-due-date algorithm. The scheduling disciplines that best fit this architecture’s needs are those that have bandwidth, jitter, and delay guarantees for different levels of QoS requirements.

Disciplines like weighted fair queuing (WFQ), weighted round robin (WRR), class-based queuing (CBQ), and priority queuing (PQ) have been found to be effective and essential for DiffServ networks [Cisco1999] [RFC 2474]. These disciplines are considered for the MPLS-based DiffServ backbone in this architecture. The traffic flows in the architecture are scheduled independently in the end-systems. The flows are then aggregated and scheduled as an integrated flow in the ISP networks.

3.3.1.3.2. Flow Policing, Marking and Shaping

The first step in flow conditioning is to determine whether or not packets from a traffic flow are out of profile. ATM traffic management examples are discussed here. These are relevant because the QoS mapping followed in this architecture needs strict QoS control mechanisms at the periphery of the networks and ATM traffic management methods are directly applicable. In ATM, the most stringent compliance algorithm is the generic cell rate algorithm (GCRA) [Nata2000] [Luiz2001]. If packets are not conforming there is a choice to either mark them or to drop them. In this architecture, DiffServ marking is generally done to change the DSCP value to one that corresponds to lower QoS levels.

Traffic shaping in this architecture is mandatory. The objective of traffic shaping is to create a conforming flow from a non-conforming flow. According to Luiz, “it may also be necessary for a public network to reshape the traffic, even if the traffic is shaped by the source” [Luiz2001]. Traffic shaping may change the data flow characteristics by intentionally delaying some packets using queuing and scheduling strategies. Well-known algorithms that are used for traffic conformance testing are the leaky bucket algorithm and virtual scheduling algorithms for CBR conformance testing. The dual leaky bucket and dual virtual scheduling algorithms are generally meant for VBR-traffic

conformance testing. Token bucket algorithms in place of leaky bucket algorithms are also used to add more flexibility when faced with bursty traffic [Nata2000].

3.3.1.3.3. Flow Control and Congestion Control

Flow control is usually done in the transport layer and in some cases, in the data link layer. According to Srinivasan, “the flow control mechanisms can be classified as open-loop, closed-loop, and hybrid mechanisms” [Srin2001]. In the open-loop method, the source describes its desired flow rate and the network decides whether to accept the call and the sender transmits at that rate. The call setup phase involves the network prescribing parameters followed by the user choosing the parameter values and then the network admits or denies call. The data transmission starts with user sending within parameter range, the network policing the users, and also uses scheduling mechanisms. In the closed-loop method, the source has to adjust its rate based on feedback from the network or the receiver. The common schemes are on-off, stop-and-wait, static window, DECbit, NETBLT, PacketPair, and the ATM Forum’s EERC [Srin2001]. In the hybrid method, the source asks for a minimum rate but can send more if available.

This MPLS-based architecture aids transport layer flow control and congestion control mechanisms by being capable of supporting explicit congestion notification (ECN). The explicit congestion notification (ECN) concept is a softer alternative to packet drop mechanisms like random early detection (RED). In RED, intermediate routers drop packets, when they decide that they are entering a period of congestion. In ECN, the routers notify the source that there is congestion on the link and thus help the sources reduce their input traffic when needed. This is done through marking the packets selected by RED by setting a congestion experienced bit in the IP header of packets from ECN-capable transport connections. Two bits in the IP header form the ECN field. The first bit is referred to as the ECN-capable transport bit (ECT). This ECT bit is set by the source and indicates whether the transport connection can support ECN [Mark2000]. The second bit is the congestion experienced (CE) bit. When the destination host receives a packet with CE bit set, it sets a bit in the TCP header in a packet back to the source. The sender can react by decreasing its congestion window.

If an end-to-end path traverses an MPLS tunnel, the CE bit of the IP header is not accessible to the LSRs for marking. To permit ECN to be used here, Krishnamurthy proposed to use one bit in the EXP bits of the shim header [Kkra1999]. In another MPLS-based method explained by Mark, the function of the EXP bit is not overloaded and does not carry information about ECT [Mark2000].

3.3.1.3.4. Flow Synchronization

According to Cristina, “flow synchronization is required to control the event ordering and precise timings of multimedia interactions. Event synchronization with or without user interaction, continuous synchronization for disparate sources and sinks, audio-video synchronization, and lip synchronization are some examples of flow synchronization” [Cris1998]. The ISP level MME can be configured to make related flows travel in the

same LSP, thereby guaranteeing flow-level synchronization. The network administrator can configure these statically or the LSPs can be configured dynamically by distributing labels such that the flows that need to be synchronized use the same LSP. The SLA and SLS can give parameters that can identify the flows to be synchronized.

3.3.1.4. QoS Management

QoS management in this architecture extends the features of MPLS as described below.

3.3.1.4.1. QoS Monitoring and QoS Availability

QoS monitoring allows the architecture to track the ongoing QoS levels achieved by the lower layers. According to Cristina, “QoS availability allows the application to specify the interval over which one or more QoS parameters can be monitored and the application informed” [Cris1998]. MPLS supports QoS monitoring and QoS availability through operation and maintenance (OAM) cells in MPLS LSPs, per-LSP statistics, link management protocol (LMP), and ECN notification support.

The OAM functionality in the MPLS user-plane includes different OAM cells such as connectivity verification cells, forward defect identifier (FDI) cells, backward defect identifier (BDI) cells, and performance cells [Neil2001]. The connectivity verification or CV OAM cells are sent periodically from LSP source to LSP sink. These CV cells can be used for defect detection related to misrouting of LSPs, link or node failure, and to trigger path protection switching. The forward defect identifier (FDI) and backward defect identifier (BDI) OAM packets carry the defect type and location to the near and far end, respectively. The BDI can be used by an LSP source to start or stop QoS aggregation. The performance OAM (P-OAM), which is still in specification stages, can be used to measure user-plane loss of packets and their aggregate octets. According to Neil, “through OAM functions, an operator can verify whether QoS guarantees given in SLAs, are in fact being met by the connection” [Neil2001].

Per-LSP statistics can be gathered easily in MPLS. These statistics can be used to characterize traffic, optimize performance, and plan capacity. These statistics can also be used as input to network analysis and planning tools to identify bottlenecks and trunk utilization [Chuc2001]. The per-LSP statistics can be used to monitor service utilization and end-to-end traffic demand matrix and by providing constraint-based routing protocols with fine-grained information between every ingress-egress pair. SNMP Management information base definitions for MPLS give more information about per-LSP parameters. Definitions of management information base (MIB) objects for MPLS and LDP have been defined [Joan2000].

3.3.1.4.2. QoS Degradation and QoS Maintenance

This architecture supports QoS maintenance resulting from QoS degradation by extending MPLS functionality. MPLS provides loop control and path protection mechanisms to give high quality service guarantees to flows. MPLS also provides

restoration or recovery in the form of rerouting using backup LSPs or dynamic LSPs. MPLS also supports crank-back schemes to aid MPLS signaling protocols and MPLS routing protocols by giving feedback on LSP setup and link failures.

Routing transients are network conditions where the routing information is constantly changing. These changes occur due to link failures or router failures or both. During such conditions, routing information stored at different routers may be inconsistent leading to temporary loops. Loops cause increased consumption of network resources and delay in network convergence. Methods for loop handling in MPLS include TTL-decrement, buffer allocation, path vector based determination, and colored-threads-based loop detection and prevention [Bruc2000]. Loops can lead to packet drops, as the TTL values on the IP packets will expire in due course. Packet drops due to loops depends on two factors, the time it takes for a router to detect the failure and the time it takes for it to distribute this information to the rest of the network. To reduce packet losses during transient conditions, traffic rerouting around the failed link or node should be followed. MPLS provides fast rerouting using constraint-based routing and protection LSPs. According to this method, MPLS constraint-based routing is used to construct a protection LSP around a link. When the link fails, the LSR attached to the failed link can use the MPLS label stacking facility to nest all the LSPs that used to go over the failed link onto the protection LSP. This type of local rerouting can lead to sub-optimal forwarding. But while the protection LSP is used, the information about this failure is distributed via OSPF to all the LSRs including the ingress LSR and the MME through the OAM cells. Once the MME knows about the failure, it can use constraint-based routing to compute a new route and then establish new label forwarding state along the freshly computed route [Bruc2000]. LSPs that carry high quality QoS traffic can be configured with such an end-to-end protection LSP with lower-level QoS traffic supported by localized protection.

The heterogeneous network includes wireless networks. Fixed wireless networks, as considered in this research, add link degradation in addition to link failures. The following MPLS-based architecture is proposed to overcome the limitations. This wireless architecture is based on a similar architecture for mobile wireless networks [Bija2000]. The proposed fixed wireless label switched architecture consists of base stations, modified LSRs that can do inter-base station handoff, fixed wireless network nodes (FWNN) on top of offices or buildings, and MPLS LSRs that are part of the wired MPLS network segments. The main features of this architecture in handling the wireless domain include the presence of more than one base station and FWNN pairs to reach downstream destination networks, little or no mobility of wireless nodes, inter-base station handoffs, and use of modified LSRs (Mod-LSRs) for inter-base station handoff. The presence of multiple base station and FWNN pairs to reach the same destination network is needed to ensure reach and to achieve strict QoS guarantees in the event of wireless link failures and link quality degradation. As this architecture assumes fixed wireless, little or no mobility of network nodes is assumed. The nodes are primarily for providing wireless network services to offices and buildings. The inter-base station handoffs are done to ensure that the downstream networks are not disconnected due to link quality and link failure problems in the wireless domain and not due to node

mobility. The handoffs are done inside the wired domain as in the case of mobile wireless networks.

Moreover, the multiple base station and FWNN pairs to reach the downstream destination networks should not fall under the same risk group (RGs). For example, if in a particular segment of a region, heavy rainfall degrades LMDS connectivity to a particular downstream destination network, the presence of other pairs of base stations and FWNNs in this segment of the region also is in the same risk group. The use of the other base station and FWNN pair should then be avoided. Thus, the multiple pairs of base stations and FWNNs should be sufficiently separated by distance or by some other dimension, as much as possible, that they not belong to the same risk group. This requirement can be relaxed in segments that carry only best effort traffic. For premium traffic classes, this requirement is mandatory in this architecture. Mod-LSRs extend the functionality of a common LSR with support for inter-base station handoff mechanisms. It should be noted that an LSP extends as far as these mod-LSRs and not to base stations. This is to ensure a clean separation of functionality for base stations and LSRs. A mod-LSR supports many base station and FFNN pairs in this architecture.

A routing area (RA) is defined which includes a mod-LSR controlling a set of base stations with each base station controlling many FFNNs. QoS degradation due to link failures and link quality degradation is handled in two ways, through base station level fault tolerance and mod-LSR level fault tolerance. The base station level fault tolerance consists of procedures to take care of temporary and transient link degradations by switching to other radio links or radio frequencies. This can be done based on the designs of the Havana or Insignia base station enhancements [Javi1998][Seou2000]. The mod-LSR level fault tolerance takes care of rerouting traffic between different base station and FFNN pairs of different RGs to ensure QoS maintenance.

The mod-LSR level fault tolerance is supported through polling and a trap message model between the LSR and the base station. This model is based on the Simple Network Management Protocol's (SNMP's) trap model [RFC1905]. According to this model, the mod-LSR polls the base stations in turn, with a pre-determined inter-poll duration, to determine the status of the wireless links. QoS degradation due to link failures or other reasons initiates rerouting. In case the base station notices a significant QoS degradation, it can notify the mod-LSR on its own using trap messages. The mod-LSR then initiates a rerouting according to QoS requirements. The rerouting and handoff scenarios are carried out by the mod-LSR as follows. In one case, the mod-LSR decides to reroute traffic over a different base station and FFNN pair to reach the same destination network. The mod-LSR routes the packets through the next feasible base station and FFNN pair to reach the destination. The feasibility is determined by the QoS requirements of the flow. In another case, if the mod-LSR notices that it cannot reroute packets to the destination network through any of the existing base station and FFNN pairs, it pushes the problem to the upstream LSR. The base station and the FFNN pairs might be sharing the same RG. The upstream LSR can either switch traffic through another pre-constructed LSP that can handle the flow or it can initiate the construction of an LSP. Depending on the QoS requirements of the flow, the policy of "make-before-

break” [Bija2000] is followed. While sending packets meant for the downstream destination networks, the mod-LSR must check the status of the path and select a path through a particular base station and FFWN pair that can handle the flow at that time. The mod-LSR can also send packets through multiple paths to ensure delivery. Duplicates and out of order packets are not handled in this architecture and are left for further study.

3.3.1.4.3. QoS Signaling Heterogeneity

The Internet transport architecture is moving toward a model of high-speed routers interconnected by intelligent optical core networks [Bala2000][Greg2000a]. With such heterogeneous networks, there can be devices in the network that do not process data based on the information carried in either packet or cell headers and that understand only timeslots, wavelengths, or physical ports. For the purposes of this architecture, the intelligent optical core network is assumed to be an IP over dense wavelength division multiplex (DWDM) network. Utilizing an MPLS-based IP-centric control plane within optical networks to support dynamic provisioning and restoration of paths established through the optical domain seems to be a possible solution for handling such heterogeneity. The optical domain needs special handling because the optical interconnects (OXC) and the optical network data plane cannot forward or recognize IP packets.

The assumed IP over DWDM network core architecture has the following properties [Bala2000].

- IP routers are attached to an optical core network, and connected to the peers at the other end of the network through dynamically established switched lightpaths.
- The optical core itself is incapable of processing IP packets.
- The interaction between the IP routers and the optical core is over a well-defined signaling and routing interface called the optical user-network interface (O-UNI).
- The optical core is subdivided into several optical subnetworks connected through a well-defined signaling and routing interface called the optical network-network interface (O-NNI).

The IP routers at the edge must establish lightpaths before communication at the IP layer can begin. Thus, the IP data plane over optical networks acts as an overlay network. The IP routers and OXC) have a peer model in the control-plane for dynamic discovery of IP endpoints attached to the optical core network [Bala2000]. The proposed architecture assumes that the IP control plane and the optical network control planes are similar and are both based on MPLS. But, service models based on IP over optical networks can decide the tightness or looseness of the coupling between the two control planes. According to Bala, “IP over optical service models can be peer, overlay, and inter-domain models” [Bala2000].

The peer model where the IP and optical networks are treated together as a single integrated network managed and traffic engineered in the same manner. GMPLS [Pete2001] is envisioned as the single signaling protocol in this case with the optical

network services being invoked implicitly as part of an end-to-end MPLS LSP. The overlay model assumes that the routing, signaling, and other control functions are entirely independent and the model is conceptually similar to the current IP over ATM overlay network model. The inter-domain model has separate routing instances in the IP and optical domains but information from one routing instance is passed on to the other routing instance and signaling messages to setup optical services takes place through explicit signaling at the UNI. The signaling has to take into account the special characteristics of the optical domain. GMPLS extension for RSVP-TE and CR-LDP is chosen to be the solution.

The proposed architecture uses the inter-domain model for IP over optical networks. This is done because the inter-domain model combines the best practices of the peer model and overlay models. This model is relatively easy to deploy and avoids the complex IP routing adjacency management over the optical network present in other models. GMPLS is described below followed by the details of IP over optical integration in the proposed architecture.

For interoperability among heterogeneous networks, MPLS signaling architecture has been extended to consider devices that cannot understand packet or cell headers [Pete2001]. According to Pete, the interfaces can be:

- “packet switch capable (PSC) interfaces that can recognize packet/cell boundaries and can forward data based on the content of the packet/cell header,
- time-division multiplex capable (TDM) interfaces that can recognize forwarding data only based on the data’s time slot in a repeating cycle such as SONET crossconnects,
- lambda switch capable (LSC) interfaces that are capable of forwarding data only based on the wavelength on which the data is received such as optical crossconnects (OXC), and
- fiber switch capable (FSC) interfaces that are capable of forwarding data based on a position of the data in the real world physical spaces such as OXCs with fiber level operations” [Pete2001].

Using the concept of nested LSPs [Kire2000], one can build a forwarding hierarchy with FSC interfaces at the top, followed by LSC interfaces, followed by TDM, and PSC interfaces [Pete2001]. MPLS signaling mainly meant for the PSC interfaces has been extended to include all four types of interfaces and is called the generalized MPLS signaling (GMPLS). GMPLS is the generalization of the MPLS control plane. GMPLS is used in this architecture to allow heterogeneity. GMPLS signaling differs from traditional MPLS (TMPLS) by supporting TDM, lambda, and fiber switching. In traditional MPLS there is a requirement that an LSP that carries IP has to start and end in a router, while in GMPLS it has been extended requiring an LSP to start and end on similar types of LSRs. GMPLS allows an LSP to carry a payload that is not limited to IP traffic. GMPLS modifies the structure of MPLS labels, label request procedures, the unidirectional nature of an LSP, modes of operation, and error reporting procedures. These modifications are done to cater to differing payloads in the LSPs, and to take into

account that the label spaces in certain non-PSC links are much smaller than those in PSC type links. GMPLS also supports the termination of an LSP on a specific egress port.

GMPLS labels are called generalized labels and are different than the traditional MPLS labels. The label does not have a hierarchy like the traditional MPLS labels and when multiple levels of labels are required, each LSP must be established separately. The label is variable length and the interpretation is left to the LSRs that use this label. The GMPLS label request carries an LSP encoding parameter representing the nature of the LSP that supports communication of characteristics required to support the LSP being requested. There are ways in GMPLS for upstream nodes to suggest labels for a downstream node. These methods permit the upstream node to start configuring its hardware before the downstream node communicates the proposed label. This pre-knowledge is generally suitable for optical LSRs that need time to adjust its hardware consisting of micro-mirrors. This feature is just a recommendation and the downstream nodes can reject or accept it.

GMPLS, introduces the need for bi-directional LSPs. In traditional MPLS the LSPs are essentially unidirectional. With bi-directional LSPs, both the downstream and upstream data paths are established using a single set of signaling messages. This method reduces the setup latency to one RTT plus processing time. This method also limits the control overhead to be the same as that for a unidirectional LSP. The bi-directional LSP setup is indicated by the presence of upstream label information in the signaling messages. Contention resolution mechanisms to resolve simultaneous port or resource requests by the bi-directional LSP setup have also been studied [Pete2001]. There are ways in GMPLS where the LSR ingress can specify the label to be used on a link leading to fine-grained LSP control. GMPLS also introduces mechanisms by which protection requirements can be requested for a particular LSP setup [Pete2001].

GMPLS is based on MPLS-TE and, hence, traffic engineering extensions to IGPs are assumed to be a basic requirement for using GMPLS. Both RSVP-TE and CR-LDP have been extended for GMPLS [Pete2001b] [Pete2001c]. GMPLS mandates a downstream-on-demand label distribution with ingress-initiated ordered control mode [Eric2000]. A liberal label retention mode is normally used with label allocation by either control-driven or data-driven models. Route selection for LSPs is normally by explicit routing using an online-CBR or an offline-CBR. A routing domain is made of GMPLS nodes. An IP router with SONET/SDH interface card is an example of a GMPLS node. GMPLS requires IGPs with the necessary extensions to support non-PSC type links. Non-PSC layers introduce several scalability concerns because hundreds of parallel physical links can now connect two nodes through the concept of wavelengths. It becomes impractical to associate an IP address to each end of the individual links and, hence, scalability is a concern. Link bundling, unnumbered links, and extensions to routing protocols are some of the enhancements proposed to address the scalability concern [Pete2001a]. In this architecture, there is no direct control interaction between clients and respective OXCs. According to Osama, “the core optical network can provide lightpath creation, deletion, modification, and status inquiry services to clients”[Osam2001a]. Lightpath creation allows a lightpath with specified attributes to be created between a pair of network ports

in OXCs. Lightpath deletion allows an existing lightpath to be deleted and resources reclaimed. Lightpath modification allows certain parameters of the lightpath to be modified. Lightpath status enquiry allows the status of a lightpath's parameters to be queried. RSVP and LDP have been extended to support the above services [John2000][Osam2001b] and are used here to meet the UNI signaling requirements. The optical domain's MME does necessary checks to ensure authenticity and availability. When all the downstream domains have agreed to support the data flow, it instructs the ingress OXC-LSR to proceed with the lightpath operations such as creation and deletion.

The peer model, on the contrary can use GMPLS throughout the end-to-end path without the help of the control entities or management entities in ISP or backbone networks. This model requires that a single routing protocol instance run over both the IP and optical domains. Though this eliminates the need of management entities in the proposed architecture, the peer model requires a tight integration between the IP clients and optical network that is found to be impractical in the near term [Bala2000] [Darr2000] [Greg2000b].

3.3.1.4.4. QoS Scalability

For the purposes of QoS and traffic engineering, flows in MPLS are called traffic trunks [RFC 2702]. A traffic trunk is defined as a collection of individual flows that share two common properties. The first one is that all flows are forwarded along the same path and the second is that they all share the same class of service. To provide scalability and QoS guarantees, MPLS constraint-based routing routes only traffic trunks and not single flows. By routing at the granularity of traffic trunks scalability is achieved. According to [RFC2702], the "basic attributes of traffic trunks are:

- traffic parameter attributes like peak rates, average rates, and burst size,
- generic path selection and management attributes like resource class affinity, adaptivity, and load distribution,
- priority attribute to define the relative importance,
- preemption attribute to determine whether the traffic trunk can preempt a specific traffic trunk, and
- resilience attribute that determines the behavior of a traffic trunk under fault conditions."

In this architecture MPLS traffic trunks are used as the basis of MPLS traffic engineering. To further improve the scalability there are LSP aggregation and label merging [Eric2000] and these are followed in the ISP and core networks.

3.3.2. Architecture Comparisons

This section starts with a brief introduction of well-known QoS architectures and then presents a qualitative comparison of the architecture proposed in this research with the other QoS architectures based on the framework established by Cristina [Cris1998].

3.3.2.1. Summary of QoS Architectures

This section summarizes the different QoS architectures that have been proposed in the literature.

3.3.2.1.1. Heidelberg QoS Model

According to Cristina, the HeiProject at IBM has developed a comprehensive QoS model to provide QoS guarantees in both end-systems and network, support guaranteed levels of service, provides QoS mapping and QoS management, heterogeneous QoS support, multicast support and QoS adaptivity [Cris1998].

3.3.2.1.2. XRM

The COMET group at Columbia University is developing an extended integrated reference model (XRM) as a modeling framework for control and management of multimedia communication networks. This model contains five distinct planes supporting network management, resource control, protocol modeling, and end-system abstractions.

3.3.2.1.3. OMEGA

According to Cristina, “the University of Pennsylvania’s OMEGA is meant to examine the interrelationship between application QoS requirements and the ability of local and global resource management to satisfy these demands. This architecture assumes a network subsystem that can provide delay and jitter bounds along with an operating system that is capable of providing run-time QoS guarantees” [Cris1998]. The essence of the OMEGA system is the resource reservation and end-to-end resource management.

3.3.2.1.4. Integrated Services Architecture

IntServ is a comprehensive IP-based QoS framework and architecture. IntServ can provide best effort, controlled delay, predicated delay, and guaranteed delay bounds. IntServ specifies QoS mechanisms for the network and not for the end-systems.

2.3.2.1.5. QoS-A

According to Cristina, “the Quality of Service Architecture (QoS-A) is a layered architecture of services and mechanisms for QoS management and control of continuous media flows in multiservice networks”[Cris1998]. This architecture incorporates flows, service contracts, and flow management.

3.3.2.1.6. OSI QoS Framework

This generic architecture emphasizes representation of QoS requirements, description of QoS characteristics, QoS categories, and QoS management functions.

3.3.2.1.7. Tenet Architecture

The Tenet group at the University of California, Berkeley created the Tenet architecture that contains a real-time channel administration protocol, real time internet protocol, and continuous media transport protocol. According to Cristina, “this architecture runs over a wide area ATM network and makes a distinction between deterministic and statistical guarantees for hard real-time and continuous media flows” [Cris1998].

3.3.2.1.8. TINA QoS Framework

According to Cristina, “the TINA QoS framework addresses the specification and realization of QoS support for telecommunications applications” [Cris1998]. QoS provision, QoS negotiation, QoS mapping, and QoS degradation are handled in this architecture.

3.3.2.1.9. MASI End-to-End Model

The CESAME project at Lab MASI, is developing an end-to-end QoS support architecture for multimedia communications. This architecture runs a generic QoS mapping for ATM-based networks [Cris1998].

3.3.2.1.10. End System QoS framework

According to Cristina, “the Washington University researchers have developed a QoS framework for providing QoS guarantees within the end-system for networked multimedia communications” [Cris1998]. The four main components are QoS specification, QoS mapping, QoS enforcement, and protocol implementation.

2.3.2.2. Qualitative Comparison of Architectures

Tables 3.5 through 3.7 compare the architectures with respect to QoS provision, QoS control, and QoS management. Table 3.5 compares QoS provisioning capabilities of the different architectures, Table 3.6 compares QoS control, and Table 3.7 compares QoS management functions. Information about existing schemes is from [Cris1998]. The features of the proposed architecture, denoted by “This Architecture,” are included for each comparison. The comparisons are done by taking into account whether or not a particular architecture supports the specified capability in both the end-system and in the network. This comparison is important because an end-to-end architecture should handle both network level QoS issues and end-system level QoS issues. A value of ‘E’ in the table cells indicates that the particular architecture handles end-system QoS capability in a detailed manner. A value of ‘N’ in the table cells indicates that the particular architecture handles network QoS capability in a detailed manner. If a particular architecture merely gives an overview of the end-system or network QoS capability, then a value of ‘(E)’ or ‘(N)’ is entered in the table. If an architecture does not discuss the end-system or network level QoS capability, a value of ‘-’ is entered.

Table 3.5. Comparison of QoS Provisioning

QoS Models	QoS Mapping	Flow Specification and Resource Allocation	Admission Control
XRM	E and N	E and N	(E)N
QoS-A	E and N	E,(N)	E and N
ISO	(E)(N)	E and N	E and N
Heidelberg	(E)N	E and N	E and N
TINA	(E)	(N)	N
IETF	E and N	-	E
Tenet	E and N	N	N
MASI	E(N)	E,(N)	E
OMEGA	E(N)	E,(N)	E, (N)
WashU	E	E	-
This Architecture	N,(E)	N	N,(E)
Legend			
-	Not addressed in this architecture		
E,N	Addressed in detail in this architecture for both the end-system and network		
(E),(N)	Not addressed in detail, but touched on, in this architecture		

Table 3.6. Comparison of QoS Control

QoS Models	E2E Flow Scheduling	Flow Shaping	Flow Control	QoS Filtering	Flow Synchronization
XRM	(E),N	-	N	-	-
QoS-A	E,(N)	E	(E)	(E),N	E
ISO	-	-	-	-	-
Heidelberg	E,(N)	(E)	(N)	N	-
TINA	-	-	-	-	(N)
IETF	-	-	-	-	-
Tenet	N	N	(E)	N	-
MASI	E	-	-	-	E
OMEGA	E,(N)	E	E	-	-
WashU	E	E	-	-	-
This architecture	(N)(E)	(N)(E)	(N)(E)	(N)(E)	-
Legend					
-	Not addressed in this architecture				
E,N	Addressed in detail in this architecture for both the end-system and network				
(E),(N)	Not addressed in detail, but touched on, in this architecture				

Table 3.7. Comparison of QoS Management

QoS Models	Monitoring/Alerts	QoS Maintenance
XRM	N	-
QoS-A	EAD	ENRS
ISO	EN	EN
Heidelberg	ED	ERS
TINA	(N)	-
IETF	EN	ENR
Tenet	ED	ERS
MASI	E	E
OMEGA	E	ER
WashU	-	ER
This architecture	(E)NAD	(E)NRS
Legend		
-	Not addressed in this architecture	
E,N	Addressed in detail in this architecture for both the end-system and network	
(E),(N)	Not addressed in detail, but touched on, in this architecture	
A	QoS availability addressed in detail	
D	QoS degradation addressed in detail	
S	QoS scalability addressed in detail	
R	QoS re-negotiation addressed in detail	

From Table 3.5, it is evident that the proposed architecture is essentially an architecture that concentrates on QoS mechanisms for the network but end-system QoS issues are also discussed. Unlike the architectures from Washington University and IETF, all the three QoS provisioning issues were discussed in the proposed architecture. While XRM and QoS-A architectures operate on an end-to-end basis with detailed end-system and network QoS provisioning, the proposed architecture focuses on network level QoS issues. All the architectures except the proposed architecture provide single-level QoS mapping. Table 3.6 compares the QoS control features. The QoS control features for this architecture are leveraged from existing research and, hence, are not discussed in depth. But, current end-system and network QoS control algorithms could be incorporated in the proposed architecture. QoS management in the proposed architecture is superior when compared to the other architectures due to MPLS management capabilities and the presence of MMEs. The proposed architecture also discusses issues like QoS scalability and renegotiation in detail as compared to architectures like MASI, XRM, and ISO.

3.4. Summary

This chapter discussed the proposed architecture in detail and provided comparisons to existing QoS architectures that fulfill similar requirements. The next chapter presents a qualitative analysis of the two main signaling protocols that are used in the proposed architecture.

Chapter 4. Qualitative Analysis of RSVP-TE and CR-LDP

This chapter presents a qualitative analysis of RSVP-TE and CR-LDP, the two main signaling protocols required for establishing intra-domain MPLS LSPs with bandwidth guarantees. The comparison takes into account the setup and maintenance of a bandwidth-guaranteed LSP in the proposed architecture and analyzes the differences between the two protocols.

The protocols are analyzed under the conditions of bandwidth-guaranteed LSP setup, bandwidth-guaranteed LSP maintenance and error handling, and LSP teardown.

4.1. Qualitative Analysis

Figure 4.1 shows the reference network for the following discussions. LSRs A, B, and C are ISP1 LSRs. LSRs A and C are the LERs for the ISP1 network. LSRs D, E, and F are the backbone LSRs. LSRs D and F are the LERs for the backbone. LSRs G, H, and I are the ISP2 LSRs. LSRs G and I are the LERs for the ISP2 network.

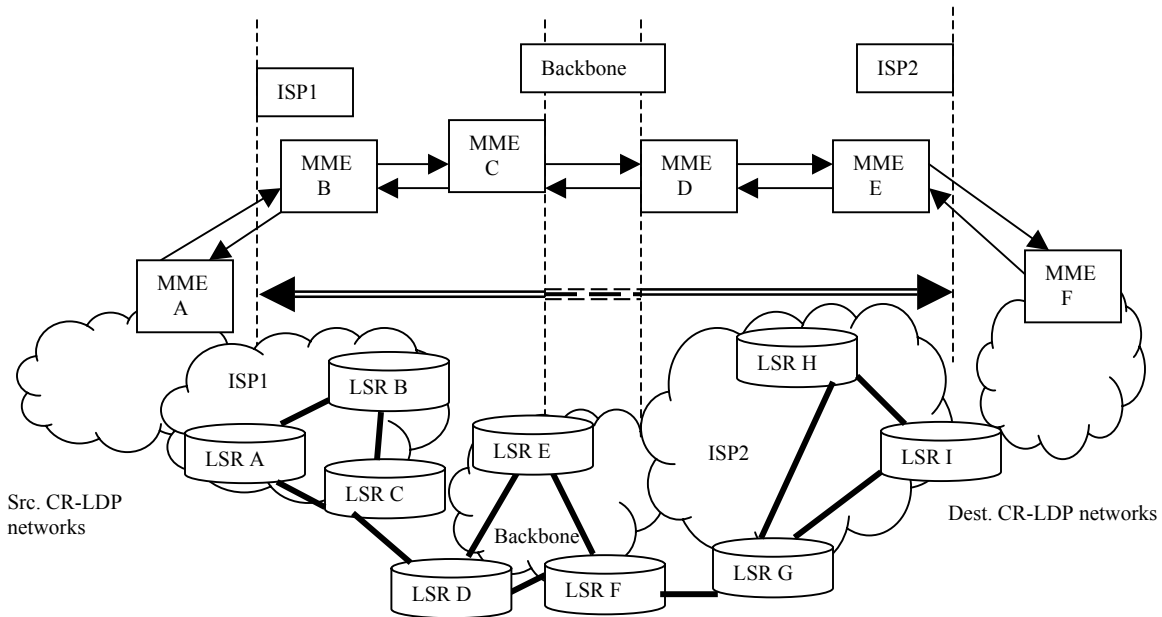


Figure 4.1. Reference network.

4.1.1. LSP Setup

Consider a scenario in which support for real-time applications is requested by the source CR-LDP network and the details are given to MME B by the customer domain MME A. Successful end-to-end reservations are done and now there is a need to commit resources. Consider the ISP1 domain. Assume that there are also administrative constraints involved in selection of the routes an LSP can take. MME B uses COPS to request the ingress LSR A to set up a bandwidth-guaranteed LSP to LSR C in the form of loose explicit route constraints. The loose constraints are as follows.

{specific node A}—{group 1}—{specific node B}—{group 2}—{specific node C}

The specific node requests for A, B, and C are strict constraints. The path between A and B through group 1 and the path between B and C through group 2 are loose explicit route constraints. MME A also sends peak data rate and burst data rates. This route could be obtained using a CSPF computation on a traffic engineering database (TED) maintained by the MME. The TED is built by traffic engineering extensions to IGPs.

CR-LDP handles the LSP setup as follows. A CR-LDP label request message generated by LER A will contain an ER-TLV containing the explicitly routed path [Bile2000]. In this case, the ER-TLV will contain four ER-Hop TLVs. The ER-Hop TLV that corresponds to the group 1 or 2 will contain a list of nodes that could potentially be used to satisfy the group constraints. A label request message can arrive at a node that is part of group 1, but is not a routing neighbor for the next hop in the explicit route. In this case, the LSR forwards the label request message with the ER-Hop TLVs intact. In the event that the LSR receives a label request message that it can directly send it to the next LSR mentioned in the explicit route, the LSR removes the topmost ER-TLV and forwards the rest to the next hop. When the label request message reaches the egress, i.e., LSRC, the egress sends a CR-LDP label mapping message back to the source along the same path traversed by the label request message. In this way, the label request and label mapping messages traverse the MPLS domain. To satisfy the bandwidth reservations for the real-time traffic, the PDR and PBS parameters of the traffic parameter TLV are specified and sent along with the label request message. The CDR and CBS parameters are set equal to those values of PDR and PBS. The MME also configures the QoS control modules in the ingress and egress LSRs for admission control. This configuration drops packets that exceed the PDR and PBS specifications. Negotiation flags for these parameters are generally set so that the intermediate LSRs and MMEs can increment or decrement the PDR and PBS values depending on the resources in a particular domain. The negotiated parameter values are sent back in the CR-LDP label mapping message to make the MMEs and LSRs adjust the domain resources accordingly. Failure in the CR-LSP setup is flagged via CR-LDP notification messages. The status TLVs in these messages specify events that led to the failure. Status codes for various failure conditions are defined in [Bile2000].

RSVP-TE handles the LSP setup as follows. The RSVP-TE path message can contain a LABEL_REQUEST object, EXPLICIT_ROUTE object, RECORD_ROUTE object, SESSION_ATTRIBUTE object, and CoS FLOWSPEC object. The path message containing the LABEL_REQUEST object indicates that a label binding is requested for this LSP [Chuc2000a]. The path message includes an EXPLICIT_ROUTE object with four sub-objects. The four sub-objects include two IPv4 sub-objects and two autonomous system sub-objects. This composition is to take into account the two specific nodes and two groups. The processing is similar to that done by CR-LDP. Each LSP using RSVP-TE must be established using a reservation style [Chuc2000a]. Among the shared-explicit, fixed filter, and wildcard filter reservation styles, the wildcard filter is not used with the EXPLICIT_ROUTE object. The path message utilizes SENDER_TSPEC and

POLICY_DATA objects to support the real-time flow specifications. The POLICY_DATA object can contain user credentials for issues like quotas, accounting, and user classes. If a LSR cannot process the request or cannot assign a label, an RSVP-TE PathErr message is sent back to the sender.

The difference in the LSP setup is the choice of transport protocol to forward the request and mapping messages. RSVP-TE uses connectionless IP or UDP. CR-LDP uses UDP for neighbor discovery and TCP sessions for label requests and mapping messages. The choice between UDP and TCP affects the scalability and reliability of the signaling protocols. Some issues regarding the choice of TCP and IP are shown in Table 4.1.

Table 4.1. LSP Setup [Data2000]

RSVP-TE	CR-LDP
On some platforms, raw IP accessibility is restricted.	TCP may be not available in some implementations.
RSVP requires that all received IP packets be delivered to the RSVP module without regard to destination addresses. This will require changes to IP stack.	No such changes are needed.

4.1.1.1. LSP Security

LSP security is needed to achieve authentication of valid users and prevention of unauthorized users from affecting valid user’s flow. According to Michael, “MPLS in its native form cannot provide protection against core mis-configurations and attacks that come from within the core, data encryption, integrity, and origin authentication, and customer network security” [Mich2001].

Hence, the security of a secure MPLS (SMPLS) LSP is of primary importance in an MPLS-based VPN. A basic requirement here is to prevent a packet destined for a host within a given VPN from reaching a host with the same address in another VPN. It has been found that an MPLS network can be secured to a level comparable to an ATM or Frame Relay-based network [Mich2001]. A secure MPLS domain of interpretation (DOI) for internet security association key management protocol (ISAKMP) is presented in [Tiss2001]. Parameters have been defined for internet key exchange protocol (IKE) to establish a security association on behalf of the secure MPLS DOI. According to Tissa, “these associations are used to secure phase II exchanges in the secure MPLS DOI. Payload encryption, authentication, and protection against various attacks such as connection hijacking, replay, and man-in-the-middle for MPLS networks have been protected against, using the above method” [Tiss2001]. SMPLS-Authentication Header (SMPLS-AH) and SMPLS-Encapsulating Security Payload (SMPLS-ESP) are two headers that are defined for secure MPLS. They are carried using shim header encapsulation [Tiss2001a]. LDP makes use of the TCP MD5 signature option to protect the LDP traffic between two adjacent LSRs. Security mechanisms for non-adjacent LSR

traffic setup are also available and protect the LDP traffic against the replay attacks and integrity protection attacks.

TCP is vulnerable to denial of service attacks and the performance degradation of a TCP session due to unauthorized access can affect CR-LDP signaling. Authentication and policy control are specified for RSVP [Data2000]. A disadvantage for RSVP-TE is that IP security protocol (IPSEC) cannot be used when there is a need of intermediate LSRs to access RSVP-TE messages like the path message. CR-LDP can utilize IPSEC as part of the normal processing itself [Data2000]. However, IKE message exchange for RSVP-TE has been defined [Tiss2001a]. This exchange includes the introduction of a new transport message type that will not be processed by any intermediate nodes and a new RSVP object called Secure_MPLS_Message.

4.1.1.2. Upper Layer Protocol

Intermediate LSRs like B,E, and H and egress LSRs like LSR C, F, and I need to have knowledge of the upper layer protocol for correct data handling when there are errors in MPLS messages. To achieve this, RSVP-TE has a L3PID object defined in label request message. If the intermediate or egress LSRs do not support the Layer 3 protocol (IP) as specified in L3PID, they should report an error using the RSVP PathErr message [Dani2000]. This feature is currently not supported in CR-LDP.

4.1.2. ER-LSP and CR-LSP Maintenance

The section discusses issues related to ER-LSP and CR-LSP maintenance, including rerouting, path protection, QoS guarantees, fault-tolerance, LSP priority, and GMPLS extensions for heterogeneous networks.

4.1.2.1. Scalability

This section addresses the scalability concerns of the LSPs between LSRs D, E, and F. These are backbone LSPs and should scale to handle high levels of backbone traffic. According to an article from DataConnection, “the scalability of a protocol should be considered in terms of the number of network flows the protocol supports, resources needed to maintain the protocol state at each node, and the CPU load at each node” [Data2000]. Both protocols have similar methods for LSP setup, an end-to-end request and an end-to-end response [Data2000]. The main difference between the two protocols is that RSVP-TE is a soft-state protocol while CR-LDP is a hard-state protocol.

RSVP-TE being a soft-state protocol must periodically transmit RSVP Path and Resv messages to refresh the protocol state of each LSP between adjacent nodes. The soft-state nature has differing results depending on the intent of the protocol’s use. According to the article from DataConnection, “the RSVP path message is of the order of 128 bytes, increasing by 16 bytes per IPv4 hop if an explicit route is used. An RSVP-TE resv message is of the order of 100 bytes. With 10,000 LSPs on a link and a refresh period of 30 seconds, the soft-state refresh consumes 600 Kbps of link bandwidth” [Data2000].

The type of link, type of the MPLS application, and the diameter of the RSVP network decide this statistic. If the topology is dynamic, soft-state protocols can pick up changes to the routing tree automatically. To counter the unreliability of the IP network layer the soft-state refresh mechanism can be used. In case of failures, valuable resources at the nodes are not kept unused and are released automatically when the refresh period passes by without any refresh messages. If the diameter of the RSVP network is large, soft-state refresh can consume considerable overhead in the form of network traffic. In soft-state protocols, state information about the flows through an LSR must be maintained at each LSR. This information is periodically refreshed using the soft-state refresh mechanism. This data must include traffic parameters, resource reservations, and explicit routes. According to the article from DataConnection, “this data amounts to an order of 500 bytes at each LSR per LSP. For the case of 10,000 LSPs, this amounts to about 5 Mb of memory space” [Data2000]. Core routers must possess large amounts of memory to handle the backbone load. But, this memory requirement may be considerable in the case of customer premises routers or small ISP-level routers. The central processing unit (CPU) load on the LSRs increases proportionally to the number of messages they must parse and act on. This load also varies by the processing complexity required for each message. In the soft-state refresh method, the CPU load will increase depending on the number of LSPs and the refresh timer settings. Figure 4.2 shows the soft-state refresh performance with respect to a high number of LSPs and low or high value of the soft-state refresh timer.

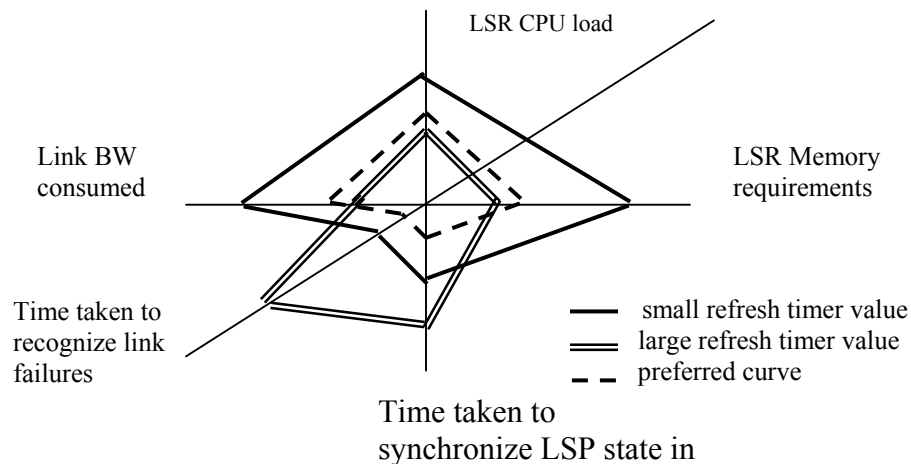


Figure 4.2. Soft-state refresh performance (high number of LSPs).

RSVP-TE is relieved of the RSVP scalability problems by using scalability extensions. In the proposed architecture, RSVP-TE is used in core networks that alleviate some of its disadvantages. In the core networks, RSVP-TE is used only to set up and maintain LSPs and is not used to maintain the user traffic flows that use the LSPs. The flows that use the LSPs are instead managed using DiffServ. DiffServ does not need per-flow state information maintenance in the backbone LSRs. Thus, the state information in core routers is maintained only for the LSPs. To limit the number of LSPs in the backbone, LSP aggregation and label merging [Eric2000] are considered in this architecture.

Extensions are proposed for RSVP-TE to counter the scalability, latency, and reliability problems with RSVP signaling and they are used in the proposed architecture. They include Bundle message extensions, MESSAGE_ID objects, the summary refresh mechanism, and the Hello protocol mechanism [Chuc2000a]. The Bundle message extension reduces the overall volume of RSVP messages by aggregating multiple RSVP messages within a single protocol data unit. A sub-message of the bundle message can be any RSVP message except another bundle message. The MESSAGE_ID and the MESSAGE_ID_ACK objects are defined to support reliable delivery of RSVP messages through an acknowledgement mechanism. The MESSAGE_ID object reduces refresh message processing by allowing the receiver to easily identify a message that contains unchanged state information. The MESSAGE_ID_ACK is meant to confirm the delivery of MESSAGE_ID object. The Summary Refresh message extension eliminates the need to generate periodic RSVP Path and Resv refresh messages by producing one summary refresh for all the flows that go through that LSR. The MESSAGE_ID extension is also used as part of the summary refresh extension. The Hello protocol extension detects the failure of a neighbor node or a reset of a neighbor's RSVP state information. In conventional RSVP, monitoring the neighbors occurs as part of the conventional RSVP's soft-state model. This requires a low refresh timer value to decrease the duration between two RSVP refresh messages for effective detection of failures. But, this leads to an increase in refresh message processing overhead. Use of the Hello protocol extension prevents low refresh timer values and, hence, leads to less processing overhead.

CR-LDP, on the contrary, is a hard-state protocol requiring no periodic refresh and works by using TCP as the transport protocol for control messages. CR-LDP also achieves reliable delivery of its LABEL_REQUEST and LABEL_MAPPING messages. CR-LDP uses Hello messages for neighbor connectivity and TCP keepalive messages for TCP connection maintenance. These messages are exchanged only on a per-link basis and not per-LSP. According to an article from DataConnection, "CR-LDP also requires state maintenance at ingress, egress, and intermediate LSRs but could be scaled down to 500 bytes at end points and about 200 bytes in the intermediate LSRs per LSP. If CR-LDP rerouting and resource modifications are needed, the overhead in the intermediate LSRs increases. The CPU load for CR-LDP messages is not a significant issue. The CPU load to re-route LSPs is likely to be even higher than LSP setup" [Data2000].

4.1.2.2. Availability

The LSPs between LSRs B, E, and F form the backbone LSPs. The backbone networks must have highly reliable links. In the case of backbone link failures, the signaling protocols do fault notification and fault handling.

Availability is a measure of the percentage of time that a node is in service. High reliability is often associated with having 99.999 percent availability. This reliability is achieved by ensuring fast detection, and fast revival from faults with no or minimal service disruption. Because RSVP works over the IP protocol, it easily adapts to failures

or online software upgrades. State refresh is another mechanism by which RSVP achieves fast revival from failures. According to an article from DataConnection, “CR-LDP on the other hand, assumes reliable delivery of control messages and so is not well placed to survive failures” [Data2000]. Another problem with CR-LDP is that if a TCP connection fails due to link failure, all the LSPs that are signaled by the connection will fail and, hence, re-establishment of all the affected LSPs is needed.

4.1.2.3. Failure Detection and Error Reporting

This section focuses on the signaling protocols’ ability to detect failures and their error handling features. Link level failure notifications, e.g., from point-to-point links, are used by the signaling protocols to detect failures. In cases where the link level information is not present, the signaling protocols must use other methods like checking for acknowledgements to detect failures.

CR-LDP uses LDP Hello messages for neighbor detection and failure detection. The discovery messages or Hello messages are sent periodically as a UDP packet to the LDP port at the “all routers on this subnet” group multicast address. When an LSR wants to initiate a connection with another LER, then TCP messages are used. LDP sessions between LSRs that are not connected directly are supported by the LDP extended discovery methods [RFC 3036]. CR-LDP uses the regular receipt of LDP discovery Hellos to indicate that the peer is up and that the peer is willing to continue using the advertised label space. A hold timer is associated with the Hello messages. There is also a keepalive timer for each peer session that it resets whenever it receives a LDP PDU from the peer. If the timer expires, closing the transport connection closes the LDP session and the failure of the peer is assumed [RFC 3036].

In RSVP-TE, Path and Resv refresh messages serve to discover active links. But, to maintain robust failure detection the refresh timer must be set to a small value. This small refresh timer value has other detrimental effects as shown in Figure 4.2. To alleviate the disadvantages of low refresh timer values and still maintain robust failure detection, the RSVP Hello extension is used [Dani2000]. This mechanism provides node-to-node failure detection. Another form of rapid failure detection is possible through the Notify and NotifyRequest objects [Pete2001a]. The label request message from RSVP-TE requests intermediate routers to provide a label binding. If a node is incapable of providing a label or faces other problems, it sends a PathErr message with an appropriate error code. After successful path messages, a node uses the label carried in the RSVP-TE LABEL object as the outgoing label associated with the sender. If an MPLS path is requested but an RSVP-incapable router stands in the path and cannot be circumvented, a PathErr is generated [Dani2000]. The Resv messages being sent upstream can encounter label errors, such as unacceptable label value for the range requested. In these cases, an RSVP-TE ResvErr message is generated with the appropriate error code. The presence of unrecognized RSVP objects in the RSVP-TE messages also results in error messages.

Establishment of a CR-LSP using CR-LDP can fail for a number of reasons. All such failures are reported using the LDP Notification message [Bile2000]. This message generally carries a Status TLV to report error reasons and codes. The error codes and details of the errors can be had from [RFC3036] and [Bile2000]. There are special status codes defined for CR-LDP errors [Bile2000].

4.1.2.4. Fault Tolerant Traffic Engineering

The RSVP-TE and CR-LDP protocols are studied with respect to their traffic engineering support, LSP modification, LSP protection, and LSP rerouting. The traffic engineering extensions of the two protocols are compared in Table 4.3.

Table 4.3. Traffic Engineering Extensions

Feature	RSVP-TE	CR-LDP
Label request	PATH message with a session type of LSP_TUNNEL_IPv4 or LSP_TUNNEL_IPv6 and a LABEL_REQUEST message.	LDP Label Request message with mandatory FEC, LSPID TLVs and optional ER, Traffic, Pinning, Resource Class, and Pre-emption TLVs.
Label mapping	LABEL object in RESV message.	LDP Label Mapping message with mandatory FEC-TLV, Label-TLV, and Label Request Message ID TLV. The optional TLVs are LSPID and Traffic.
Explicit route specification	EXPLICIT_ROUTE object in PATH message.	ER TLV with ER-Hop TLVs in the Label Request message.
Traffic parameters	SENDER_TEMPLATE, SENDER_TSPEC, FLOW_SPEC objects in the PATH message.	Traffic Parameters TLV in the Label Request message.
LSP priority	SESSION_ATTRIBUTE object in SESSION.	Preemption TLV with setup and holding priorities.
Policy support	POLICY_DATA object.	Not supported.
Restriction methods in resource selection	Not supported.	Resource Class TLV.
Loop detection	RECORD_ROUTE object.	LDP PathVector and Hop Count TLVs.
Error handling	PATHERR and RESVERR objects.	Notification and Label Release messages.
Tear down of an	PATHTEAR and RESVTEAR	Label Release and Label

CR-LSP	objects.	Withdraw messages.
Confirmation of reservations	RESVCONF message.	Not supported.

After a CR-LSP is established, its bandwidth reservation or other parameter values may need to be changed without service interruption. Thus, LSP modification is meant to change the traffic parameters for an LSP with rerouting as an optional side effect. RSVP-TE inherently supports this requirement. To increase bandwidth for an LSP, a new path message with a new LSP_ID object is sent while the current LSP_ID continues to be refreshed. According to Daniel, “if the network and the destination domains accept the increase, the current LSP can be torn-down after switching traffic to the new LSP” [Dani2000]. In CR-LDP, the bandwidth modification is achieved through the *modify* value for the *action indicator flag* in the LSPID TLV [Gera2001]. The current procedure is restricted to change requests by the ingress LSR. According to Gerald, “the traffic parameters TLV, the ER-TLV, the resource color TLV, and preemption TLV can have different values than those of the established LSP” [Gera2001].

LSP rerouting pertains to provisioning of a new route for an LSP on failure notification, topology change, or for optimization purposes. A strictly specified explicit route cannot be re-routed except by the ingress LSR [Data2000]. A loosely specified explicit route may be re-routed if any of the following conditions hold [Data2000].

- A failure of a link or neighbor is detected.
- A better route becomes available.
- The resources for the LSP are required by a new, higher priority LSP (LSP preemption).

CR-LSP re-routing is supported by both CR-LDP and RSVP-TE. RSVP-TE can install a new route by simply refreshing the path message for an LSP to a different next-hop as soon as the alternate route is available. The old path is removed following its normal time out [Data2000]. Another mechanism is “make-before-break,” where the old path is used while the new path is being setup. The LSR performing the re-routing swaps to the new path and then tears down the old connection. The “make-before-break” mechanism is supported using the shared explicit reservation style in RSVP-TE [Dani2000]. The LSPID TLV supports this mechanism in CR-LDP [Gera2001]. Re-routing of loosely specified parts of LSPs at intermediate LSRs when a better route becomes available can lead to thrashing in networks. Thrashing can be avoided through pinning the route. Pinning is supported in CR-LDP through the pinning TLV. In RSVP-TE, the initial route is specified with a loose hop. The RSVP-TE record route object is subsequently used on the Path and Resv messages to store information about the selected route and the information is then sent back to the ingress. The ingress uses this information to re-issue the path message with a strictly specified explicit route [Data2000].

LSP protection is meant to do automatic traffic transfer to the backup paths when the primary LSP fails. This mechanism is similar to re-routing, but is generally considered to be a time-critical operation instead of standard re-routing. Network survivability is the main goal behind network fault tolerance, resilience, and path protection. Due to the

current and future need of IP networks to carry mission-critical data, real-time, and high-priority traffic, network survivability becomes a key issue. The current routing protocols take on the order of seconds or minutes to recover from a failure and this can severely affect time-critical applications [LiMo2000]. Path-oriented technologies like MPLS can be used to provide network survivability for IP-based networks to efficiently support these application requirements for faster recovery [LiMo2000, KenO2000, Srin2000, Atsu2000, Srig2000].

The idea here is to pre-establish protection or backup LSPs for working LSPs, and achieve better protection switching times compared to legacy IP networks. This protection mechanism entails the following.

- Selection of the working and protection paths.
- Signaling the setup of working and protection paths.
- Determination of the specific level of protection.
- Fault detection to detect faults along the path [Atsu2000].
- Fault notification mechanism to notify the concerned network entity. The MPLS OAM functionality can be used for this purpose [Neil2001].
- Bandwidth reservation for protection LSPs [LiMo2000] [Srig2000].
- Switchover mechanism to move traffic over from a working LSP to the protection LSP.
- Switchback of traffic to the repaired primary LSP [KenO2000].

To understand the RSVP-TE and CR-LDP extensions for path protection, the following components are introduced.

- “A path switch LSR (PSL) is defined as the transmitter of both the working path traffic and its corresponding recovery path traffic. The PSL is responsible for switching of the traffic between the working and the recovery path. This functionality gets its decisions made by the MME in this architecture” [KenO2000].
- “A path merge LSR (PML) that receives both working path traffic and the recovery path traffic and either merges them both into a single outgoing path. The PML is the destination of the recovery path” [KenO2000].

The protection configuration consists of two steps, establishing the protection domain and creation of the fault notification mechanism. The fault notification mechanism could use the reverse notification tree (RNT) [KenO2000] or OAM functionality [Neil2001]. Establishing the protection domain is done using the signaling protocols and starts with the identification of the protection domain. This identification entails the identification of the PSL and the PML. Transferring the protection domain information to configure the nodes in the domain follows the previous step. The identification of the protection and working paths with the aid of the signaling protocols completes the protection set up procedure [KenO2000]. To identify the PSL, PML, and the nodes in the protection domain, an Explicit Route Protection sub-object can be added to the Explicit Route Object of RSVP-TE [Vish2000]. This identification process also requires the addition of the label-request object in the RSVP-TE Resv message and a label object in the RSVP-TE confirmation message. A new RSVP-TE notification message is proposed to carry the fault and fault recovery signal information.

CR-LDP supports Explicit Route Protection ER-Hop type to allow for the identification of PSL and PML. A Path Protection TLV is added to the CR-LDP label request message to configure the protection domain [Keno2000a]. The establishment of the protection path first needs the identification of the working path in both point-to-point and point-to-multipoint cases. The identification is done in the signaling protocols as given in Table 4.4. The nodes inside the domain will get configuration message updates from the PSL to use the configured protection LSP.

Table 4.4. Working Path Identification

	RSVP-TE	CR-LDP
Point-to-point	Same sender template consisting of sender IP address and LSPID	Same LSPID TLV
Point-to-multipoint	Same session object	Same FEC TLV

According to Sriganesh, “backup or protection LSPs can also be routed in such a way that bandwidth can be shared between backup LSPs of more than one working path while still guaranteeing recoverability for a set of failures and allows efficient bandwidth utilization” [Srig2000].

4.1.2.5. Generalized MPLS Signaling

The backbone networks and the ISP-level networks can be built from network components that differ in link layer and other protocols. A signaling protocol being used in this environment must be able to handle the heterogeneity. To account for the heterogeneity of the Internet networks, GMPLS is pursued. Differences in RSVP-TE and CR-LDP extensions for GMPLS are discussed below.

RSVP-TE for generalized MPLS signaling defines formats for GMPLS generalized label request, generalized label, waveband switching support, suggested label, and label sets [Pete2001c]. A generalized label request object in RSVP-TE is set by the ingress node, transparently passed by transit nodes and used by the egress node. The egress and PHP special case will generate an RSVP-TE Resv message. Bandwidth encoding is carried using SENDER_TSPEC and FLOWSPEC objects. The presence of both generalized label and the normal label object is a protocol error. The most important requirement from RSVP-TE for GMPLS is the need for bi-directional path establishment. The presence of an upstream label in the RSVP-TE path message indicates a bi-directional path establishment. An intermediate node must also allocate a label on the outgoing interface and fill in an upstream label before forwarding the path message. An important addition to RSVP-TE to aid GMPLS is the notion of expedited notification of failures and other events using the Notify Request, and Notify objects [Pete2001c]. The Notify Request object is used to request the generation of notifications. The Notify message provides a mechanism to inform non-adjacent nodes of LSP related events. This is different from the RSVP PathErr and ResvErr messages in that the Notify message

mechanism can be passed to a node other than the immediate upstream or downstream neighbor. An RSVP-TE Ack message is used for reliable delivery of the Notify messages. Another addition to RSVP-TE for support of GMPLS is the addition of the “path_state_removed” flag in the ERROR_SPEC of the RSVP PathErr message. This flag ensures that the idle state removal is expedited in the intermediate LSRs instead of waiting for the RSVP-TE PathTear messages. CR-LDP support for GMPLS [Pete2001c] involves a label request TLV for GMPLS label request, GMPLS bandwidth encoding in the traffic parameters TLV, and a generalized label for the GMPLS label.

CR-LDP does not support a rapid failure notification feature like the Notify and Notify Request message pair in RSVP-TE. According to Pete, “when a failure is detected, it is propagated with the CR-LDP label release and label withdraw messages generated from the point of failure” [Pete2001c]. CR-LDP supports the notion of LSP feedback to report resource information back to the source [Pete2000a]. This feature can increase the accuracy and completeness of the topology database by resolving and utilizing the latest information. This LSP feedback feature is not currently supported by RSVP-TE.

4.1.2.6. Traffic Control

According to an article from DataConnection, “RSVP-TE and CR-LDP perform resource reservation at different times in the process of LSP setup” [Data2000]. CR-LDP carries the full traffic parameters on the LABEL_REQUEST. This allows each hop to perform traffic control on the forward portion of the LSP setup. The traffic parameters are negotiated depending on the corresponding flag values in the traffic parameter TLV. Each LSR then decides on the final values depending on the network conditions in its own domain. These final values are read by the destination LSR and are passed back on the LABEL_MAPPING message from the LSR. Admission control and resource reservation are updated at each LSR using these values. This means that an LSP will not be established on a route with insufficient resources [Data2000]. RSVP-TE, on the contrary, carries a set of traffic parameters (the Tspec) in the path message. The Tspec describes the data that is likely to use the LSP. Intermediate LSRs can make only routing decisions based on this information; the information cannot be used for resource reservation. The resource reservation can only be done when the Tspec reaches the egress and the egress responds with a Flowspec. According to the DataConnection article, “this mechanism means that the reservation does not take place until the Resv passes through the network, with the result that LSP setup may fail on the selected route because of resource shortage”[Data2000]. RSVP-TE can use the Adspec object in the Path message to alleviate this problem. This Adspec object contains the resource information that is available in the intermediate LSRs. The egress can now send a Resv message taking into account the Adspec values. However, this Adspec mechanism can also fail, as there is no guarantee that the information in the Adspec is current at the time of its delivery to the egress LSR.

4.1.2.7. Policy Control

In the proposed architecture, MME and the edge LSRs communicate using the COPS protocol for policy decision information. The signaling protocol used in this scenario must be capable of transporting policy decisions to enable operation of this architecture. Policy support for a signaling protocol is important, especially for the backbone networks. RSVP-TE can explicitly work with COPS [RFC 2749] [Fran2000] [RFC 2750]. RSVP can work with COPS to outsource policy control decisions to policy servers. The RSVP Policy Data object has been defined for generic policy-based admission control [RFC 2750]. These policy objects can also be used for inter-domain policy enforcement. A new COPS client type called MPLS-COPS has been defined for use with the COPS-provisioning protocol (COPS-PR) to be used in MPLS and for traffic engineering purposes [Fran2000]. No RSVP-TE or CR-LDP specific extensions for COPS-PR have been defined until now in this work. According to the article from DataConnection, “CR-LDP currently carries implicit policy data in the form of destination addresses, and the administrative resource class in the traffic parameters”[Data2000]. The resource class parameter is meant to make the path setup choose or avoid certain links with specific attributes. The resource class parameter is not supported in RSVP-TE.

4.1.2.8. Quality of Service

In this architecture, LSRs D, E, and F form the backbone DiffServ domain where the LSPs are constructed and maintained by RSVP-TE. RSVP-TE protocol must have features for DiffServ support to enable this to be done. CR-LDP and RSVP-TE have different approaches to QoS guarantees. RSVP-TE, taking off from RSVP, follows a model close to the IP integrated services architecture, while CR-LDP follows the ATM traffic management model. The RSVP Tspec object carried in path messages describes the data that will flow through the LSPs rather than the QoS that is required from the connection. There are specifications for RSVP about how to map different QoS requirements to the Tspec parameters [RFC 2210]. In CR-LDP, the clients specify values for parameters like PDR, PBS, CDR, and CBS before hand. Traffic that does not conform to these parameter values are termed non-conforming and are subjected to traffic correction that could vary from marking to dropping packets in the traffic. This process achieves fine-grained quality of service signaling. MPLS path protection facilities have been found to aid DiffServ networks [Fran2001][Ragh2001]. A new RSVP-TE DiffServ object is defined for use by the path message.

4.1.3. LSP Teardown

Assume that a particular LSP through LSR A needs to be torn down. This decision could be due to overload at LSR A or administrative decisions from the MME that controls LSR A’s domain. The signaling protocols must support LSP teardown and the subsequent signaling to free resources maintained at the intermediate LSRs for that LSP.

In RSVP-TE, teardown messages remove path or reservation state immediately. The old paths must be explicitly torn down in RSVP-TE. There are two types of RSVP teardown messages. They are PathTear and ResvTear messages. According to [RFC2205], “A PathTear message travels towards all receivers downstream from its point of initiation and deletes path state, as well as all dependent reservation state, along the way. An ResvTear message deletes reservation state and travels towards all senders upstream from its point of initiation.” There is no reliable delivery of these messages. In CR-LDP, label release, label withdraw, and label abort messages are used for LSP teardown. In CR-LDP the path tear down is mandatory and is always explicit. There is reliable delivery of teardown messages using TCP. An LSR sends a label release message to an LDP peer to signal that the LSR no longer needs specific FEC-label mappings previously requested or advertised [RFC 3036]. The label withdraw message is used to signal a peer that the peer may not continue to use the specific FEC-label mapping. An LSR that receives a label withdraw message must reply with a label release message. In addition, a label abort request message in CR-LDP may be used to abort an outstanding label request message [RFC 3036]. This feature is not explicitly supported by RSVP-TE.

4.2. Summary

From the previous section, it is observed that both CR-LDP and RSVP-TE offer similar capabilities, but with some differences in their details. Table 4.5 summarizes the features of these protocols.

Table 4.5. Comparison of the RSVP-TE and CR-LDP Signaling Protocols

Topic	RSVP-TE	CR-LDP
Transport protocol used	UDP and mostly IP.	UDP and mostly TCP. Adds TCP overhead in connection initiation and connection maintenance.
Resource reservation	During the return of the Resv message. LSP setup may fail.	Mostly during the label_request message processing. LSP setup may not fail.
QoS mapping and QoS support	Coarse-grained QoS mapping in the form like that of IntServ.	Fine-grained QoS mapping followed in ATM traffic management.
State maintenance	Soft-state or semi-hard-state with refresh reduction.	Hard-state.
Availability with focus on fault notification and link fault sensitivity	Much better due to soft-refresh, hello extensions, rapid failure notification using notification messages.	Comparatively lower due to hard-state, no support for rapid failure notifications.
MPLS label distribution modes	Only downstream-on-demand. Needs LDP for other modes.	All modes supported because of its extensions from LDP.

Policy networking support	Easier with explicit RSVP POLICY_DATA objects.	More difficult. Only transparent support.
Previously used	Yes. In RSVP.	No. New protocol is defined for supporting MPLS.
Ease of traffic engineering	Easy. Overhead when route pinning is needed. Resource class not supported.	Easier. Route pinning and resource class are automatically supported.
Scalability	Only with extensions. Depends on use.	Natively supported due to hard-state properties.
Reliability	Only with Hello extensions.	Natively supported due to use of TCP.

The architecture proposed here uses RSVP-TE in the core networks and CR-LDP in the ISP periphery. This decision is based on the following features of these two protocols.

CR-LDP with TCP as the transport protocol hampers fast response to events such as link failures and LSP setup. Moreover, reliability in the backbone networks is high due to the use of fiber-based networks along with multiple levels of fault tolerance in the form of SONET and MPLS. The use of TCP in these already reliable links adds unwanted overhead. Lack of scalability is the main disadvantage of RSVP. CR-LDP is a much better fit. But, scalability extensions for RSVP are proposed in RSVP-TE. RSVP-TE is only used for LSP setup and not for maintaining user flows in the backbone networks. Moreover, in the backbone, using MPLS LSP aggregation and label merging, it is possible to significantly reduce the number of LSPs that need to be maintained. Also, the strict QoS control procedures in the ISP level networks bring in only conformant flows inside the backbone networks. CR-LDP with different label distribution modes is a good fit to counter the vagaries in the ISP level networks.

Given the choice of CR-LDP in the ISP periphery, the next chapter provides more specific quantitative analysis of CR-LDP based on simulation.

Chapter 5. Quantitative Analysis

The central idea behind the architecture proposed in this thesis and described in the previous chapters is the integration of MPLS and DiffServ in the backbone networks. In this chapter, quantitative analysis of DiffServ networks, MPLS networks, and integrated MPLS and DiffServ networks are presented using simulation experiments in a network simulator, ns-2. This is done to demonstrate the effectiveness of MPLS and DiffServ integration under specific circumstances.

5.1. Introduction to Network Simulator Ns-2

According to the ns-2 documentation, the network simulator Ns-2 is a “discrete event simulator targeted at networking research. It provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless local and satellite networks” [NS]. Ns-2 supports features like point-to-point links, LANs, unicast routing, multicast routing, transport protocols like UDP and TCP, application layer protocols, mobile IP, mobile ad-hoc networks, traces, and visualization [NS].

Ns-2 is object-oriented and is built using C++ and object-oriented Tcl called OTcl [NSa]. The C++ objects form the data plane in ns-2 with the OTcl objects as the control plane. There are TclCL linkage objects to connect the different types of objects. A generic set of steps to work with the simulator include [NS]:

- creation of event scheduler,
- optional trace enable,
- creation of network using node and link objects,
- setup of unicast or multicast routing,
- optional insertion of error modules,
- enabling transport layer protocols like UDP and TCP,
- enabling traffic generators for UDP and TCP, and
- association of application-level protocol to the chosen transport layer protocol.

The most commonly used internal components of ns-2 are node objects and link objects. The node objects comprise agents and classifiers. Agents are usually “protocol endpoints” and classifiers are generally “packet de-multiplexers”. The link objects “generally encapsulate a queue, delay, and possibly a time-to-live checker” [NS]. The node instance variable `entry_` carries the entry point to the node that will first handle all the packets arriving at that node. The classifiers are generally responsible for packet switching or packet forwarding. If the next hop of the packet is the same node itself, then it is sent to the node’s port classifier to determine the transport level agent to which the packet should be sent. The choice of this particular simulator is primarily due to its support for MPLS [MNS]. Other simulators like OPNET [Opnet] did not support MPLS at the time the simulation experiments were done. The other benefits of ns-2 for the purposes of this simulation include a version of MPLS and DiffServ patch [Sean2001], publicly available source code and software, an active user group, gentle learning curve, and good documentation.

The MPLS network simulator for ns-2 (MNS) is “a simulator that enables one to simulate various MPLS applications without constructing a real MPLS network” [MNS]. This simulator supports MPLS packet switching, LDP, and CR-LDP. MNS does not support RSVP-TE. Therefore, all of the simulation experiments that were carried out as part of this thesis use CR-LDP as the signaling protocol. The current version of MNS (version 2.0) features data-driven and control-driven LSP triggers with downstream and upstream modes of label allocation. MNS also supports independent and ordered mode label distribution control modes. In addition, MNS supports constraint-based routing with extensive support for ER-LSPs and CR-LSPs.

In this architecture [MNS], the `entry_` variable in the ns-2 node object contains a reference to a new classifier called MPLS classifier that determines whether the received packet is labeled or unlabeled. If labeled, the MPLS classifier does Layer 2 switching instead of Layer 3 routing. If unlabeled and an LSP for this packet has been setup previously through signaling protocols, the unlabeled packet is processed to add an MPLS label. In this case, the node acts as ingress LSR. Otherwise, the packet is forwarded to the default address classifier for Layer 3 routing. As in the usual case, if the packet is meant for itself, it is passed on to the port classifier for delivery to the agent of this node. Here the port classifier is modified for MPLS use to deliver the labeled packet to an LDP agent and unlabeled packets to the default agent. DiffServ support in ns-2 [Sean2001] includes features like DSCP addition to IP headers, conditioner components with profiles, and scheduler components. Profiles can be defined to suit the type of traffic like EF, and AF. These profiles are added to the conditioner. When a packet passes through the conditioner, the profiles that match the packet DSCP are chosen. The packet is checked whether it conforms to the traffic rate characteristics that are mentioned in the chosen profile. If the packet is non-conforming, different actions can be taken. In this patch, non-conforming EF packets are dropped and non-conforming AF packets are remarked with higher drop precedence. The scheduler in this patch consists of separate queues for EF, AF, and BE traffic classes. These queues are serviced using a variation of weighted round robin (WRR) schemes.

A new patch to the DiffServ features of ns-2 and MNS version 2.0 has been created for the purposes of this thesis. This new patch does not follow the procedure as outlined in [Fran2001]. The integration as done in this simulation is implicit rather than explicit integration followed in [Fran2001]. This implicit integration assumes that the two components of MPLS and DiffServ are each unaware of the existence of the other.

5.2. Simulation Aims and Methodology

The main function of this simulation is to quantitatively compare performance achieved by TCP traffic given different rates of UDP traffic and four different network configurations:

- only DiffServ is enabled,
- only MPLS is enabled,

- neither MPLS nor DiffServ is enabled, and
- both MPLS and DiffServ are enabled.

This comparison aims to support the most important claim for the proposed architecture, namely that there is benefit for the current Internet traffic provided by the co-existence of MPLS and DiffServ in backbone networks. TCP and UDP happen to be the most commonly used transport protocols in the current Internet. If the comparison shows better TCP and UDP performance for the case where both MPLS and DiffServ are enabled, the claim is demonstrated. The simulation setup consists of a wired network with a set of nodes and MPLS LSRs in a traditional “dumb-bell” shaped network topology as shown in Figure 5.1. The network traffic is assumed to be a mix of two types of traffic:

- File Transfer Protocol (FTP) traffic using TCP, and
- UDP traffic.

The TCP traffic sessions are of “infinite length” in that their duration is the same as that of the simulation experiment and offer infinite traffic to the network. The infinite traffic is in the form of an application data generator such that TCP will always have application data available to transmit. A single destination collects statistics for the two types of traffic. Apart from this, the FTP source also collects statistics about its view of traffic being sent into the network.

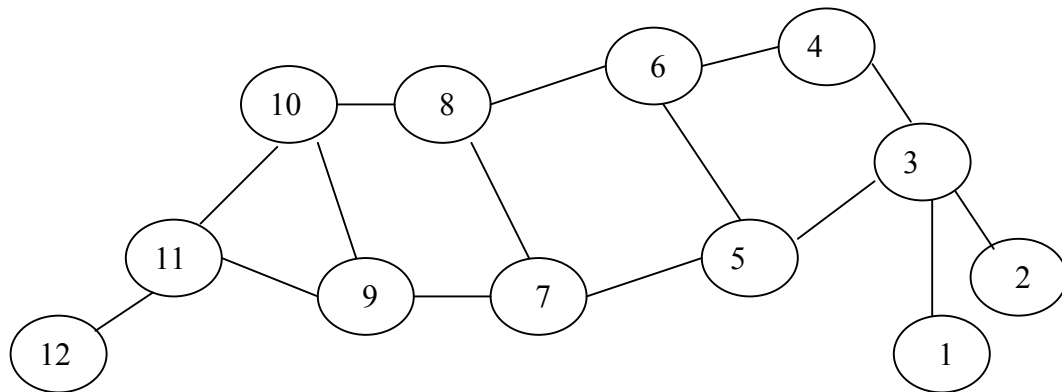


Figure 5.1. “Dumbbell” topology used in the experiments [MNS].

Varying rates of UDP traffic are mixed with the TCP traffic in a common link. This is done to simulate the effect of FTP and other “bulk” traffic, including HTTP traffic, competing with real-time multimedia traffic (RTP) in the same link. The nature of the FTP source in this simulation is such that it tries to utilize all of the bandwidth it can get from the network. The nature of the UDP source in this simulation is to generate constant bit rate (CBR) traffic. The rate of the UDP traffic is increased in different cycles of the simulation test. When there is an increase in the rate of UDP traffic, the network warns of possible congestion to both UDP and TCP sources either through ECN or RED mechanisms. The UDP and TCP sources react differently to these warnings. The UDP constant bit rate source with no inherent congestion control does not heed these network warnings and continues to send more traffic into the network. The TCP source however,

enters congestion-control mode, thus reducing its source traffic rate. But, since the UDP source does not reduce its traffic level in the network, the network remains congested and it exhibits explicit reactions to congestion in the form of equal proportion packet drops for both types of traffic. The UDP traffic source still does not respond to congestion due to its nature. This on-going congestion condition negatively affects the TCP traffic being sent through the network.

The QoS mechanism in such a network should perform preferential treatment to flows in the network. In the above case, the ill-behaved UDP traffic must be subjected to packet drops to realize fair access to network resources for both TCP and UDP traffic flows. The QoS mechanism deployed in this simulation is DiffServ. However, since DiffServ is an IP-based connection-less solution, the problems of network hot spots arise. Network hot spots can arise because of the nature of IP-layer destination-based shortest-path routing. If all the routers in a domain happen to decide on the same shortest-path to one of the gateway routers of the domain, the link leading to the domain gateway router becomes a network hot spot. This is because all the routers funnel their traffic into the same link thus exceeding the link's capacity. The traditional solution to such a problem is to increase the bandwidth of the bottleneck link. This does not always work and traffic engineering solutions with the use of multiple paths to the same destination alleviates the problem in the most efficient manner [Chuc2001]. DiffServ acting alone based on the connection-less IP model cannot support the use of multiple routes to a destination. Path-oriented mechanisms, like MPLS, can aid DiffServ in providing such a traffic engineering solution.

MPLS acting alone cannot provide much benefit because it has no inherent CoS or QoS features. It can only support an IP-based QoS or CoS architecture. There are other important benefits provided by MPLS in the MPLS and DiffServ integration as explored in this research. The benefits include effective handling of link failures and support for heterogeneous link layers. Handling of link failures has been studied in [Ragh2001]. However, these MPLS-based solutions are beyond the scope of the simulation experiments in this work.

Recently, the validity of simulations using “dumb-bell” topologies and long-lived FTP traffic session has been questioned for its non-conformance with existing traffic scenarios [Kath2001]. An extension of this simulation, to support different types of TCP traffic and different topologies, as advised in [Kath2001] is considered a relatively easy extension, but is left for future research.

5.3. Simulation Details

According to Jain, “there are three evaluation techniques, measurement, simulation and analytical modeling”. Table 5.1 gives an idea for selection of the criteria for selecting an evaluation technique [Jain1991].

Table 5.1. Criteria for Selecting an Evaluation Technique

Criterion	Analytical Modeling	Simulation	Measurement
Stage	Any	Any	Any
Time required	Small	Medium	Varies
Tools	Analysts	Computer Languages	Instrumentation
Accuracy	Low	Moderate	Varies
Trade-off evaluation	Easy	Moderate	Difficult
Cost	Small	Medium	High
Saleability	Low	Medium	High

The most important factor that forced the selection of simulation in this case is the type of technology that is being dealt with here. MPLS is a nascent technology that is still in research stages at the time of writing this thesis. Hence, simulations and analytical modeling are the two viable options. Of the two, simulation was chosen due to its higher accuracy and “saleability”. Also, at the time of this research, a complete MPLS module was readily available for the widely used network simulator ns-2.

As identified before, the goal of these simulation experiments is to show that there is benefit in combining MPLS and DiffServ in the backbone networks in this architecture. This goal is achieved through performance comparison of DiffServ, MPLS, and DiffServ plus MPLS acting in a network under similar network traffic conditions and other settings. According to Jain, “the complete lists of system and workload characteristics that affect the performance of the system are called parameters and parameters that are varied in the study are called factors” [Jain1991]. The simulation in this research is carried out with the same factors for all the experiments, so that a common ground can be reached for comparison purposes. It is this achieving common ground in factors and parameter settings that is given more focus in this simulation rather than the choice of the factors or parameter settings themselves.

The simulation is done in the form of several experiments. With each specific mix of UDP and TCP traffic, performance metrics are obtained in the network with different MPLS and DiffServ configurations. The comparison of the performance results thus obtained indicates the best level of MPLS and DiffServ integration with respect to QoS guarantees for the traffic types that are considered. There are two particularly important pre-conditions for controlling the number of simulation cycles and effective interpretation of results:

- the choice of the performance metrics, and
- the choice of the factors that are varied in the experiments.

There are different commonly used performance metrics like response time, throughput, utilization, reliability, and availability. For these simulation experiments, throughput was chosen to be the performance metric of interest. According to Jain, “throughput is defined as the rate at which the requests can be serviced by the system” [Jain1991]. The

efficiency of the network is the ratio of the maximum achievable throughput to nominal, or ideal capacity of the network. In this work, the efficiency achieved for TCP traffic is the primary performance metric, as shown below.

In these simulation experiments, the FTP (TCP) traffic is always vying for 100 percent efficiency. If the link is free of any other traffic, the nature of the TCP traffic in these experiments is such that it tries to match the nominal capacity of the link. The efficiency achieved by the TCP traffic is a good indicator of how the network treats well-behaved traffic in the presence of ill-behaved traffic. This efficiency can be measured in the form of nominal capacity or bandwidth measurement at both the destination and at the source. When UDP traffic is introduced in the network, i.e., load is increased in the network, the throughput or efficiency achieved by the TCP traffic measured at the destination and source will give an indication of fairness in the network. This TCP traffic efficiency in the wake of ever-increasing UDP traffic is measured for configurations of the network using only MPLS, only DiffServ, and MPLS plus DiffServ. The efficiency for the TCP traffic will then indicate the benefit of MPLS and DiffServ integration. To achieve common ground for the comparisons, the simulation settings must be the same for the MPLS, DiffServ, and MPLS plus DiffServ configurations. Also, the rate of increase in UDP traffic must be the same in all the experiments. Thus, there are three simulation experiments with many variations in each experiment. The number of variations is determined by the rate of increase in UDP traffic. Each variation will have different rates of UDP traffic. The same number of variations is present in each simulation experiment. UDP traffic originates from node 2 in the topology of Figure 5.1 and reaches node 12, mixing with the TCP traffic in the network segments between node 3 and node 11.

Additional simulation experiments are done to measure the overhead associated with the introduction of MPLS in the network. This overhead is categorized by measuring the signaling overhead in the network due to the use of MPLS. The simulation settings can be divided into general settings, simulation-run settings, DiffServ-specific settings, and MPLS-specific settings. These settings are the same for all of the experiments, but vary between different variations within an experiment.

The specific settings for each simulation run include the duration of the simulation run, start times of UDP and TCP traffic, and end times of UDP and TCP traffic. All the variations and all experiments follow the settings below.

- The duration of the simulation run is set at twenty-five seconds of simulated traffic.
- The start and end times of both TCP and UDP traffic are at five and twenty seconds, respectively.

The duration of the simulation is kept short, since FTP and HTTP connections in the Internet are relatively short-lived, in accordance with [Kath2001]. Durations are kept short to reduce simulation time. The variation in simulation duration is to be considered for future study and is discussed in the future work section.

The traffic rate, expressed as utilization, is given by the ratio of packet size expressed as transmission time, to the inter-arrival time. Different variations in a particular simulation

experiment can be obtained by varying UDP packet sizes or packet inter-arrival time. In these simulation experiments, the inter-arrival time is set to 0.005 seconds and the packet sizes are increased to achieve five variations of UDP traffic.

- Low (or zero), where there is no UDP traffic.
- Medium, where the packet size is 100 bytes yielding a rate of 0.16 Mbps.
- High, where the packet size is 200 bytes yielding a rate of 0.32 Mbps.
- Very High, where the packet size is 500 bytes yielding a rate of 0.8 Mbps.
- Huge, where the packet size is 1000 bytes yielding a rate of 1.6 Mbps.

Thus, there are five variations in each simulation experiment.

The link bandwidth or the link capacity of each link is set at 1 Mbps. The link from the last router, node 11 in Figure 5.1 to the destination, node 12 in Figure 5.1 has a capacity or bandwidth of 5 Mbps. This setting is to ensure that this link does not drop packets and negatively influence the measurements at the destination. All the links in all the experiments have a link propagation delay of 10 ms. These values are not based on any real system characteristics, but they are kept the same in all the experiments to achieve a common basis for comparison. The nature of the TCP traffic is from a FTP session that lasts for the duration of the simulation. The FTP session is initiated from node 1 towards node 12 in Figure 5.1. The Tahoe variation of TCP [RFC2001] is chosen in this simulation. A one-way TCP, i.e., simplex TCP data flow is chosen for this simulation because two-way TCP was still experimental in the Tahoe version of the simulation software [Nsb]. The other assumptions include error-free communication through the links and equal routing costs for all links.

The DiffServ factors for this simulation include the following.

- Peak token bucket rates in the DiffServ profiles for EF and AF traffic.
- Priority for the different types of traffic.
- DSCP tagging for different traffic types.
- Scheduling policies followed.
- Queue characteristics like queue limit.
- Nature of queues installed like drop at tail or class based queues.

The TCP traffic in this simulation is tagged “AF11,” with a peak rate set at 0.3 Mbps, while the UDP traffic is tagged “EF,” with a peak rate set at 0.8Mbps for the huge UDP traffic variation. In the simulation, non-conforming EF traffic is dropped and non-conforming AF traffic is remarked to indicate a higher drop priority. The UDP traffic is marked EF since it does not respond to network signals of congestion and, hence, can be dropped when it exceeds the set traffic levels. TCP traffic, meanwhile, reacts to network congestion due to TCP’s congestion control mechanisms and does not need strict peak rate controls. Moreover, the purpose of this simulation is to study the data rate achieved by TCP traffic and, hence, it is not rate-limited. DiffServ is, thus, configured with two profiles, one for AF and one for EF along with class-based queuing for scheduling purposes.

The MPLS-related factors that affect the outcome of the simulation include the following.

- Choice of LSP use.
- Data-driven mode or control-driven mode for the label trigger strategy.
- Downstream or upstream label distribution scheme.
- Ordered control or independent control label control mode.
- Conservative or liberal label retention mode.
- LSP characteristics of the rerouting ER-LSP, including link capacity and delay.

The data-driven mode sets up LSPs on detection of data, while the control-driven mode sets up pre-established LSPs before data transfer. Downstream label distribution operates with a LSR requesting labels for FECs from its downstream neighbor. The ordered control restricts an LSR to bind a label for an FEC only when it receives a label from its next hop LSR or when it is an egress LSR. The conservative label retention mode deletes the label binding received from LSRs that are not the next hop for that FEC, leading to fewer labels to be maintained. In this simulation, MPLS ER-LSPs is set up in two configurations. In one experiment, one ER-LSP is configured to carry both UDP and TCP traffic, while in another experiment, different ER-LSPs are configured to carry the two types of traffic. MPLS in this simulation is configured with ordered-control, the control-driven mode, and the downstream-on-demand label distribution scheme. Conservative label retention is followed. These MPLS settings are chosen to facilitate the creation of ER-LSPs and CR-LSPs.

5.4. Simulation Results

The simulation results presented below are based on experiments conducted using the topology as shown in Figure 5.1.

5.4.1. Results with MPLS and DiffServ Disabled

Figures 5.2 through 5.6 show simulation results for experiments conducted with both MPLS and DiffServ disabled. Figure 5.2 shows the case where there is no UDP traffic present in the network and, hence, the “infinite” TCP traffic consumes as much bandwidth as possible and reaches the theoretical limit of link capacity. Figure 5.3 shows results for the case where UDP traffic at a medium level mixes with the TCP traffic in the bottleneck link. As the results show, the TCP traffic consumes about eighty percent of the traffic with about twenty percent consumed by the UDP traffic. Figure 5.4 shows the case where the UDP traffic is high, occupying about forty percent of the link capacity. This decreases the instantaneous TCP bandwidth as seen at the destination. Another notable characteristic of the graphs is the difference between the TCP source’s instantaneous bandwidth and the actual bandwidth consumed by TCP as measured at the destination. From the graph, the TCP traffic at the source acts as expected with a significant increase in the rate of generated traffic even exceeding the link capacity. But, on receiving feedback from the network based on delay calculations from acknowledgements, the TCP source reduces its traffic generation rate. Figure 5.5 shows the effect of very high UDP traffic. The UDP traffic occupies eighty percent of the link capacity, and, hence, the TCP traffic bandwidth decreases as seen in the graph. Since there is no QoS or CoS mechanism enabled in the network, the network drops packets of

both traffic types equally with no preferential treatment given to the ill-behaved flow (the UDP traffic) or the well-behaved flow (the TCP traffic).

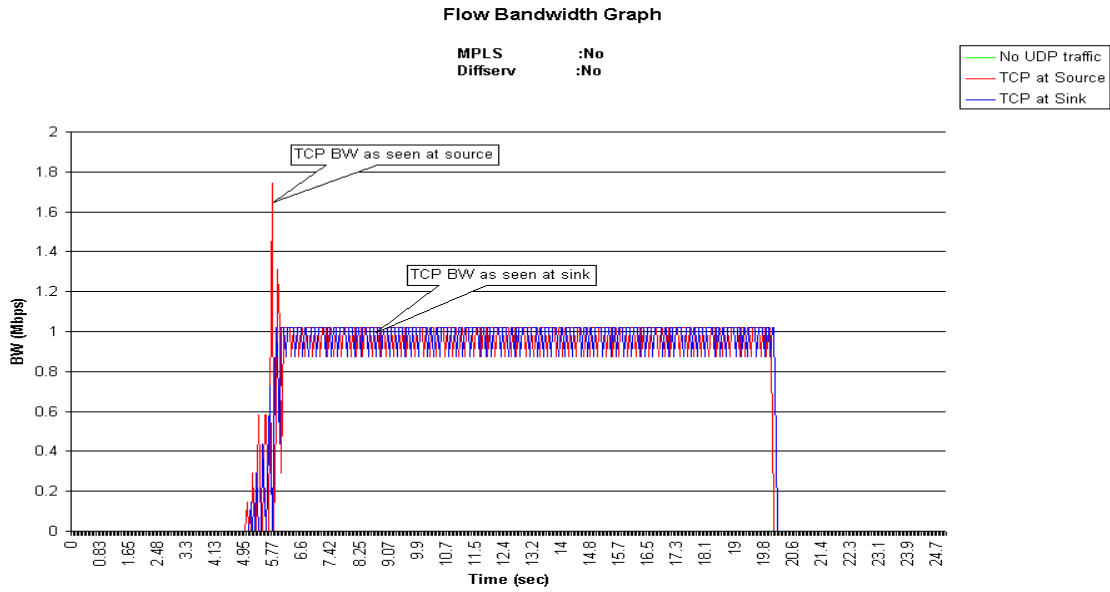


Figure 5.2. Performance with no UDP traffic (no MPLS, no DiffServ).

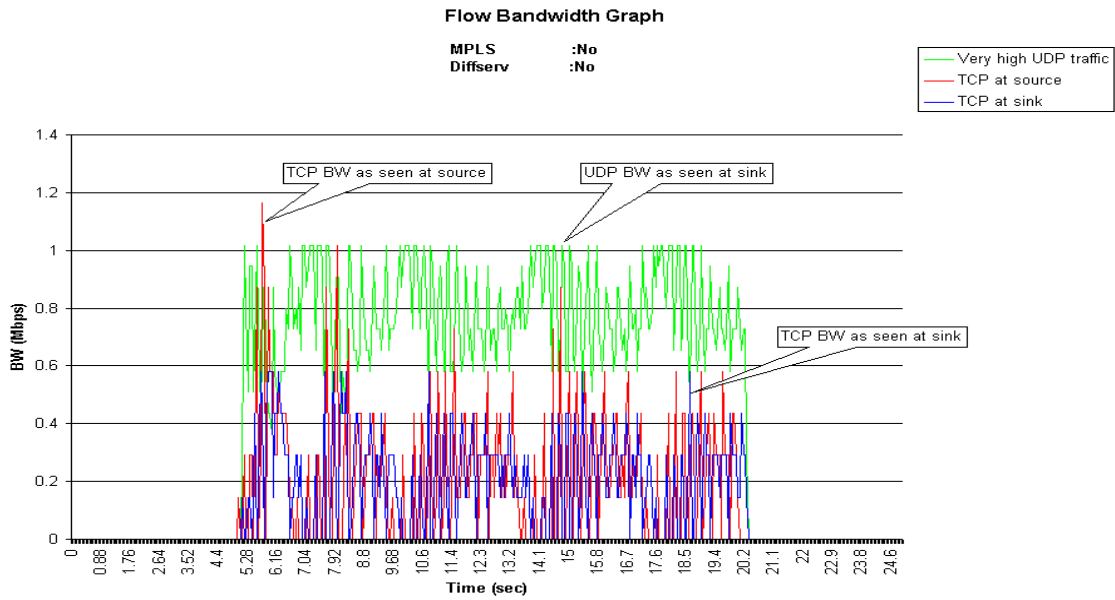


Figure 5.3. Performance with medium level of UDP traffic (no MPLS, no DiffServ).

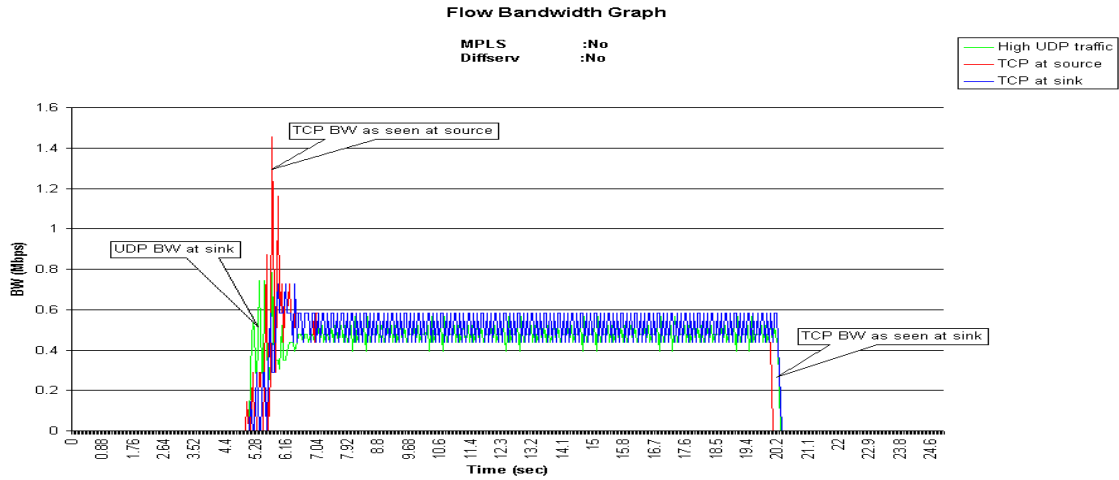


Figure 5.4. Performance with high level of UDP traffic (no MPLS, no DiffServ).

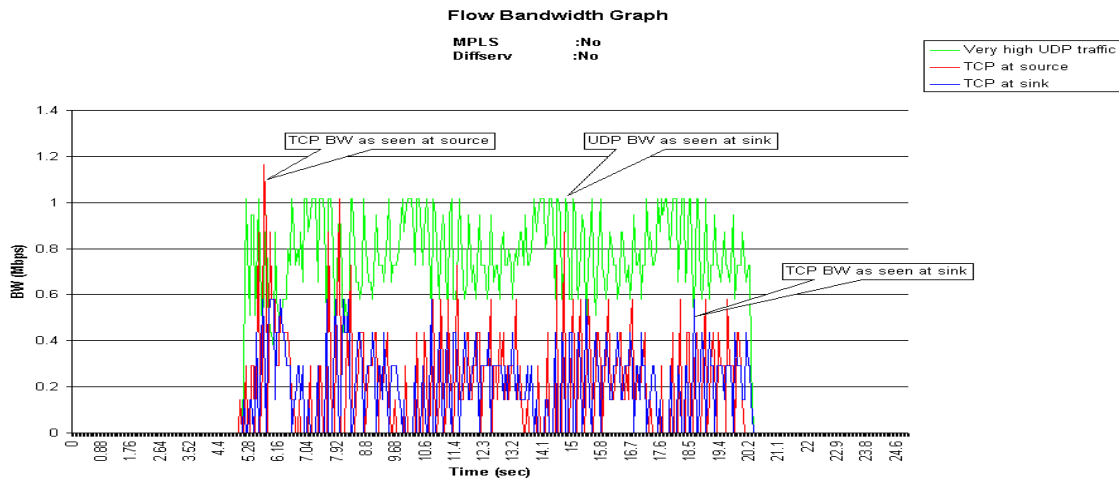


Figure 5.5. Performance with very high level of UDP traffic (no MPLS, no DiffServ).

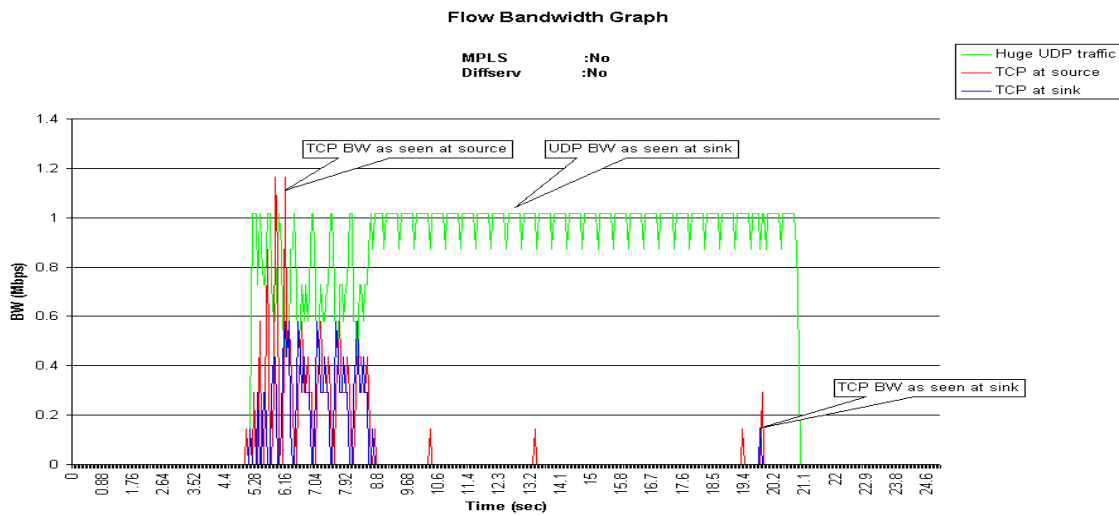


Figure 5.6. Performance with huge level of UDP traffic (no MPLS, no DiffServ).

Figure 5.6 shows the effect of a “huge” level of UDP traffic in the network with one hundred percent utilization of the link, effectively blocking most TCP traffic. Extremely low throughput is seen for the well-behaved TCP traffic. Table 5.1 shows the values for the mean, median, and mode for the throughput of UDP and TCP traffic for different levels of offered UDP traffic when MPLS and DiffServ are not used. A 95 percent confidence interval was used for the computation of these statistics.

Table 5.1. Statistics for Throughput (no MPLS, no DiffServ)

No Diffserv No MPLS		Mean (Mbps)	Median (Mbps)	Mode (Mbps)	Standard Deviation	Min (Mbps)	Max (Mbps)
Low UDP traffic	UDP	0	0	0	0	0	0
	TCP at source	0.9391	1.0182	1.0182	0.2057	0	1.7455
	TCP at sink	0.9223	1.0182	1.0182	0.2252	0	1.0182
Medium UDP traffic level	UDP	0.1564	0.1600	0.1745	0.0297	0	0.2618
	TCP at source	0.8090	0.8727	0.8727	0.1746	0	1.7455
	TCP at sink	0.7945	0.8727	0.8727	0.1902	0	0.8727
High UDP traffic level	UDP	0.4692	0.4800	0.4800	0.0794	0	0.7855
	TCP at source	0.5128	0.5818	0.5818	0.1342	0	1.4545
	TCP at sink	0.5036	0.5818	0.5818	0.1181	0	0.7273
Very high UDP traffic level	UDP	0.7797	0.7273	0.7273	0.1842	0	1.0182
	TCP at source	0.2224	0.1455	0	0.2283	0	1.1636
	TCP at sink	0.2065	0.2909	0.2909	0.1696	0	0.5818
Huge UDP traffic level	UDP	0.9316	1.0182	1.0182	0.1726	0	1.0182
	TCP at source	0.0727	0	0	0.1805	0	1.1636
	TCP at sink	0.0589	0	0	0.1405	0	0.5818

From Table 5.1, it can be observed that the maximum bandwidth utilized by TCP traffic at the destination decreases as the UDP traffic increases. The throughput degradation experienced by the TCP traffic is evident from observing the mode values in the table that go down to zero in the presence of a huge level of UDP traffic. The mean and median values for the TCP throughput at the sink also decrease with increasing UDP traffic. The increase in UDP traffic at the sink is evident from observing the maximum and mode values of the UDP rows for the five variations of traffic. The mode values of UDP traffic for the five classes increases from 0 to 1.0182 Mbps.

Figure 5.7 indicates bandwidth utilized by TCP traffic at the sink. The X-axis is time in seconds while the Y-axis is the total throughput achieved by the TCP traffic over time. Since, the TCP source is an “infinite” FTP traffic source, the total amount of transferred data over time is a good indication of the quality of the network service as experienced by

TCP traffic. From Figure 5.7, it can be observed that with no UDP traffic, the TCP source is able to transfer about 250 Mb of data while it can transfer only about 30 Mb of data with huge UDP traffic.

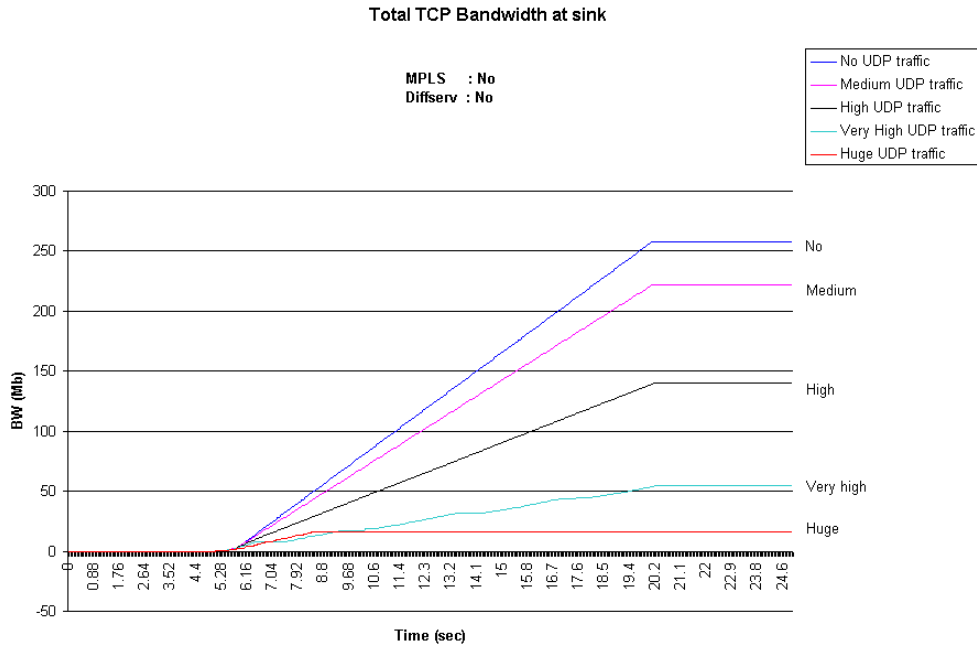


Figure 5.7. Total TCP throughput at the sink for different levels of UDP traffic (no MPLS, no DiffServ).

The simulation results indicate that mixing well-behaved and ill-behaved traffic in a common link without proper QoS can lead to significant disadvantages for the well-behaved flow.

5.4.2. Results for MPLS Enabled and DiffServ Disabled

Figures 5.8 through 5.12 show the simulation results for experiments conducted with MPLS enabled and DiffServ not enabled. In these graphs, MPLS has been enabled in the form of a single ER-LSP that carries both UDP and TCP traffic. The ER-LSP was constructed from node 3 to node 11 through nodes 5,6, and 7 in the topology of Figure 5.1 using the CR-LDP signaling protocol.

Figure 5.8 shows a case where there is no UDP traffic and, hence, the infinite TCP traffic consumes as much bandwidth as possible and reaches the theoretical limit of the link capacity. Results for this case do not differ from the case where both MPLS and DiffServ are disabled. Similarly, there is no difference between using MPLS and not using MPLS for medium and high levels of UDP traffic as shown in Figure 5.9 and 5.10 respectively. The reason is that a single ER-LSP carries both of the two types of traffic. The TCP source adjusts to the available bandwidth after the initial burst of traffic that tries to exceed the link capacity of 1 Mbps. Figure 5.11 shows the effect of very high UDP traffic. The UDP traffic occupies eighty percent of the link capacity and, hence, the TCP traffic suffers in instantaneous available bandwidth as can be seen from the graph of

Figure 5.11. In this case, enabling MPLS does not offer any benefits for the TCP traffic. This is because MPLS does not have any inherent CoS or QoS services. It does not cause any change to the IP's default best effort service. Figure 5.12 shows the effect of huge UDP traffic that utilizes 100 percent of the link capacity. This causes extremely low throughput for the TCP traffic. Table 5.2 summarizes the simulation results for the case where MPLS, but not DiffServ, is used.

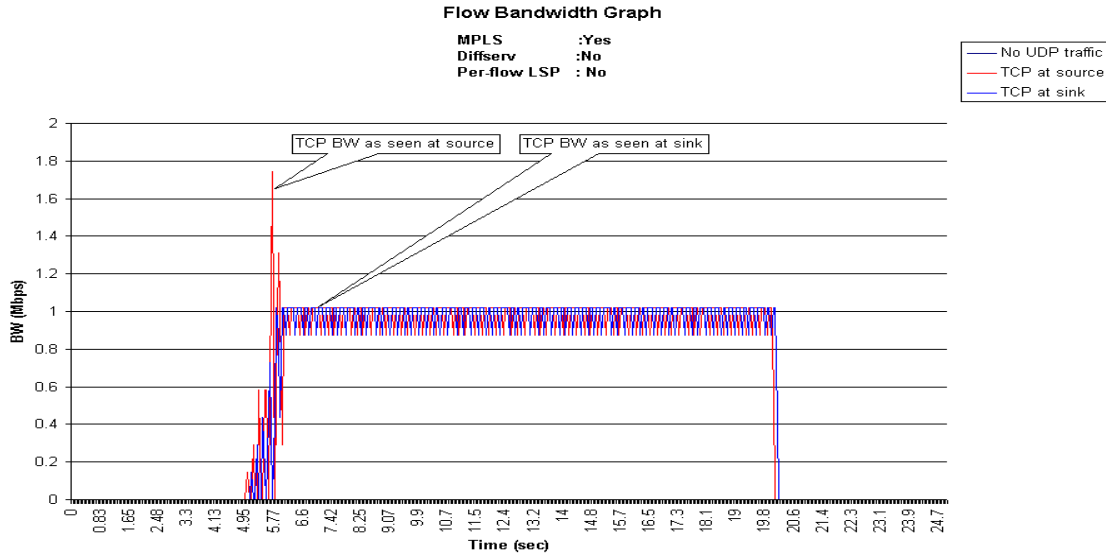


Figure 5.8. Performance with no UDP traffic (MPLS, no DiffServ).

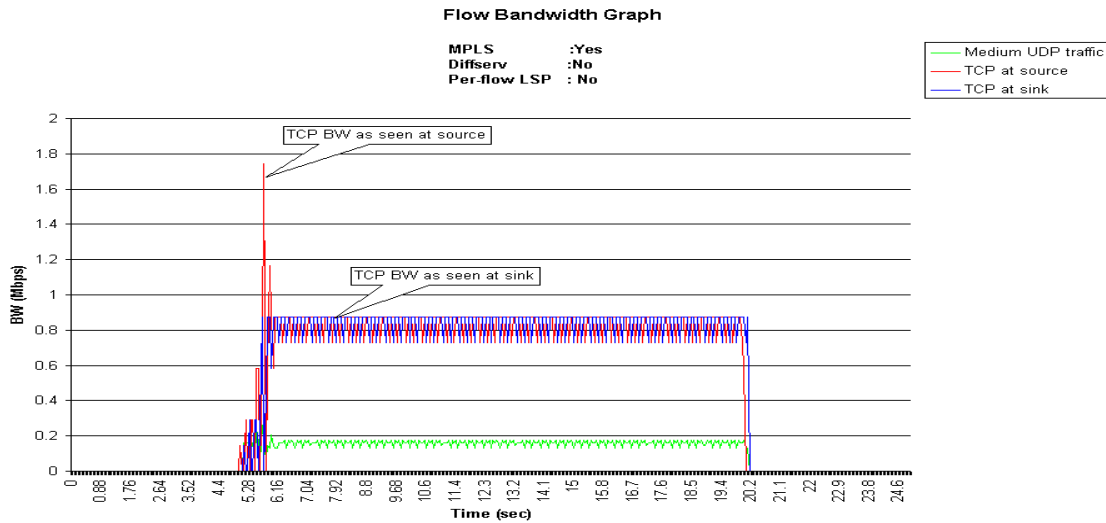


Figure 5.9. Performance with medium level of UDP traffic (MPLS, no DiffServ).

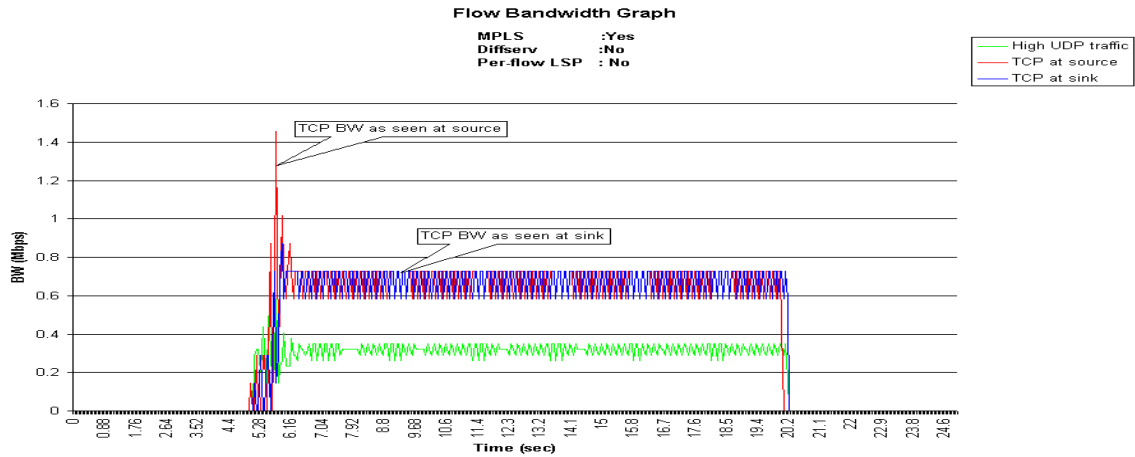


Figure 5.10. Performance with high level of UDP traffic (MPLS, no DiffServ).

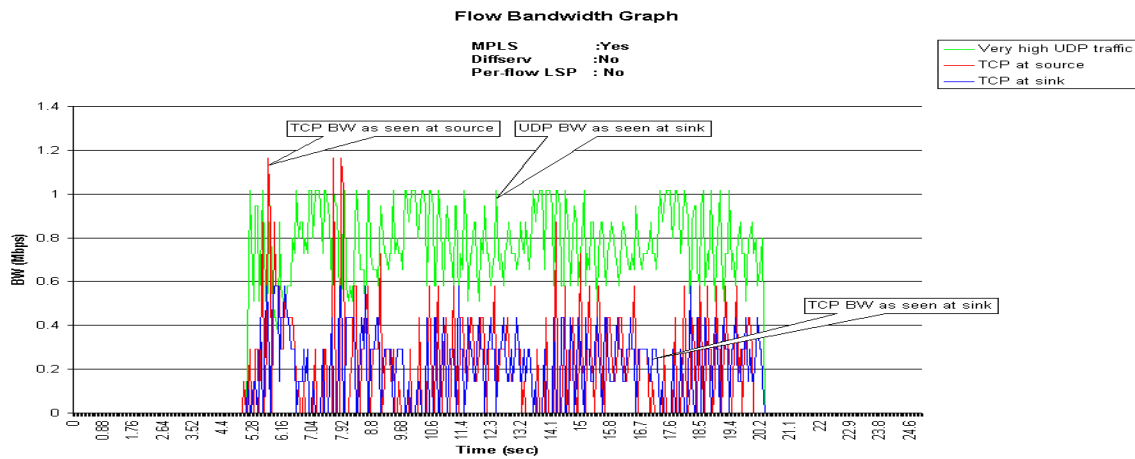


Figure 5.11. Performance with very high level of UDP traffic (MPLS, no DiffServ).

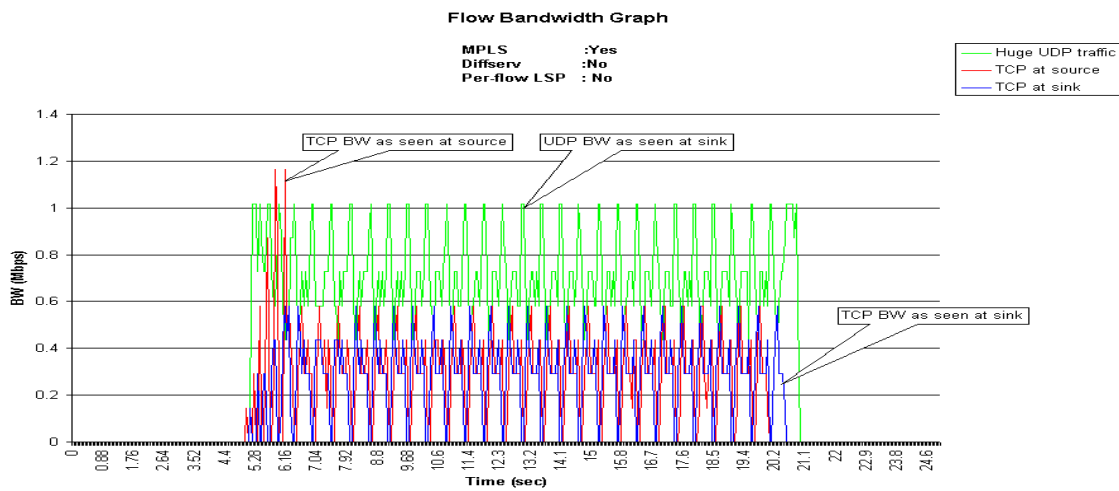


Figure 5.12. Performance with huge level of UDP traffic (MPLS, no DiffServ).

Table 5.2. Statistics for Throughput (MPLS, no DiffServ)

MPLS No Diffserv		Mean (Mbps)	Median (Mbps)	Mode (Mbps)	Standard Deviation	Min (Mbps)	Max (Mbps)
Low UDP traffic	UDP	0	0	0	0	0	0
	TCP at source	0.9391	1.0182	1.0182	0.2057	0	1.7455
	TCP at sink	0.9223	1.0182	1.0182	0.2252	0	1.0182
Medium UDP traffic	UDP	0.1564	0.1600	0.1745	0.0295	0	0.2618
	TCP at source	0.8090	0.8727	0.8727	0.1746	0	1.7455
	TCP at sink	0.7945	0.8727	0.8727	0.1902	0	0.8727
High UDP traffic	UDP	0.3128	0.3200	0.3491	0.0564	0	0.5818
	TCP at source	0.6604	0.7273	0.7273	0.1452	0	1.4545
	TCP at sink	0.6486	0.7273	0.7273	0.1524	0	0.8727
Very high UDP traffic	UDP	0.7786	0.7273	0.7273	0.1808	0	1.0182
	TCP at source	0.2230	0.2182	0	0.2359	0	1.1636
	TCP at sink	0.2091	0.2909	0.2909	0.1707	0	0.5818
Huge UDP traffic	UDP	0.7090	0.7273	0.7273	0.1916	0	1.0182
	TCP at source	0.2904	0.2909	0.2909	0.2040	0	1.1636
	TCP at sink	0.2815	0.2909	0.2909	0.1781	0	0.5818

From the data in Table 5.2, it can be observed that the values are similar to the simulation results with both MPLS and DiffServ disabled. Figure 5.13 shows the results for the total TCP throughput achieved at the destination. As expected, there is no significant difference observed from the case shown in Figure 5.7 where MPLS and DiffServ are not enabled. The increase in bandwidth for the very high and huge UDP traffic cases can be attributed to the use of MPLS-based Layer 2 switching which is more efficient than Layer 3 routing.

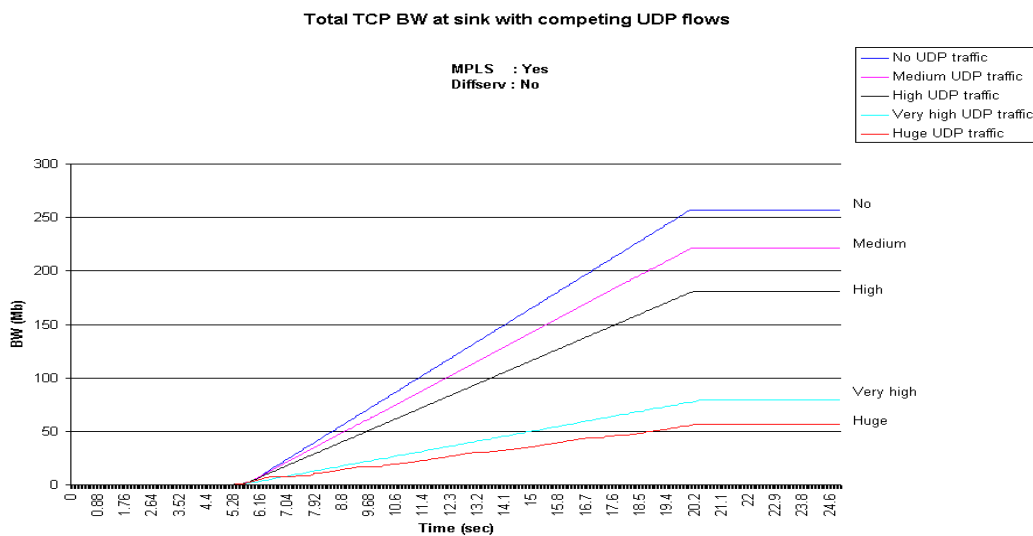


Figure 5.13. Total TCP throughput at sink for different levels of UDP traffic (MPLS, no DiffServ).

From the simulation results, it can be seen that MPLS alone does not give significant benefits as compared to using neither MPLS nor DiffServ. The increase in total TCP throughput achieved for the experiments where UDP traffic is huge or very high is due to faster Layer 2 switching.

5.4.3. Results with MPLS Disabled and DiffServ Enabled

Figures 5.14 through 5.18 show the results of experiments with DiffServ enabled and MPLS not enabled. DiffServ as modeled in ns-2 consists of three modules: the DSCP tag generator module, the conditioner and profiler module, and the scheduler module [Sean2001]. The tag generator module modifies the IP packet header to include a DSCP tag. In the simulation, EF and AF11 are used to tag the flows. Profiles in the DiffServ component define the features of the traffic for which the profile is created. Peak rate and DSCP tag values are some traffic features that are defined. These profiles are added to the conditioner module. The conditioner module, on receiving IP packets, searches through the profiles to match the traffic type to the profile. The matching is based on the DSCP tag. Once the traffic is matched against a profile, the conditioner module checks whether the traffic conforms to the traffic rate parameters described in the matched profile. If the traffic is conforming it is passed on to the scheduler module. Non-conforming traffic is either marked for a higher drop precedence or is dropped. The scheduler module consists of separate queues for the different traffic classes. In this implementation, the queues are serviced using a simple weighted round robin (WRR) scheduling scheme.

Figures 5.14 and 5.15 show the experiments with no UDP traffic and with medium UDP traffic, respectively. Figure 5.16 shows the experiment with very high UDP traffic. It can be observed from Figure 5.16 that the TCP throughput using DiffServ alone is not degraded as much as with MPLS and DiffServ disabled, as shown in Figure 5.11. This observation can be attributed to DiffServ's capability to prohibit flows from "stealing" bandwidth meant for other traffic types. Here, the UDP traffic is marked EF and its peak rate is restricted to 0.8 Mbps. If the traffic increases beyond this rate, packets belonging to that traffic type are dropped. To highlight the effectiveness of DiffServ, statistics from the conditioners used in the simulation are shown in Table 5.3. All the packets that reach the conditioner are shown. The number of scoped packets shows the number of packets that are scoped by the profiles that are defined in the conditioner. In the simulation, the number of scoped packets equals the total number of packets since there are only two profiles defined and the traffic seen at the conditioner matches one of the two profiles. The next two rows in Table 5.3 show the conformant and non-conformant traffic as perceived by the EF profiler in the conditioner module. The last two rows in Table 5.3 show the conformant and non-conformant traffic as perceived by the AF11 profiler in the conditioner module. Traffic is termed conformant if the network traffic reaches the conditioner in accordance with the peak rates defined for the profiler for that particular type of traffic. If not, the traffic is termed non-conformant.

Table 5.3. DiffServ Conditioner Statistics (DiffServ, no MPLS)

DiffServ, no MPLS	Low UDP	Medium UDP	High UDP	Very high UDP	Huge UDP
Number of packets	1769	4524	3957	3923	2845
Number of scoped packets	1769	4524	3957	3923	2845
Number of scoped CBR (EF) packets	0	3000	3000	3000	1924
Number of non-conformant CBR (EF) packets	0	0	0	0	374
Number of scoped TCP (AF11) packets	1769	1524	957	923	921
Number of non-conformant TCP (AF) packets	1206	961	394	360	359

From Table 5.3, it can be observed that the number of TCP packets that are seen at the conditioner before applying QoS, decreases with increasing UDP traffic. This shows that TCP decreases its rate of traffic on network congestion. The nature of the infinite FTP source is also evident from the fact that the number of non-conforming TCP packets is high when no UDP traffic is present. There is a large difference in non-conforming traffic between the two types of traffic due to the settings in their profiles. The EF profile is set to have a peak rate of 0.8Mbps and the AF profile is set to have a peak rate of only 0.3 Mbps. It can be observed that the total number of packets for the huge level of UDP traffic experiment has fewer packets than that for the experiment with very high level of UDP traffic. This is because the capacity of the link from the CBR UDP source to the LSR at node 3 has a capacity of only 1 Mbps. Since the source UDP traffic rate for this particular simulation run is set at 1.6 Mbps, a large number of packets are dropped at the source. Increasing the capacity of the link leading to the LSR at node 3 alleviates this situation.

Table 5.4 shows the statistics obtained for the UDP traffic at the sink, TCP traffic at the source, and the TCP traffic at the sink.

Table 5.4. Statistics for Throughput (DiffServ, no MPLS)

Diffserv No MPLS		Mean (Mbps)	Median (Mbps)	Mode (Mbps)	Standard Deviation	Min (Mbps)	Max (Mbps)
Low UDP traffic	UDP	0	0	0	0	0	0
	TCP at source	0.9391	1.0182	1.0182	0.2057	0	1.7455
	TCP at sink	0.9223	1.0182	1.0182	0.2252	0	1.0182
Medium UDP traffic	UDP	0.1564	0.1600	0.1600	0.0298	0	0.2473
	TCP at source	0.8090	0.8727	0.8727	0.1759	0	1.7455

	TCP at sink	0.7945	0.8727	0.8727	0.1898	0	0.8727
High UDP traffic	UDP	0.4692	0.4800	0.4364	0.0907	0	0.7418
	TCP at source	0.5080	0.5818	0.5818	0.1246	0	1.1636
	TCP at sink	0.4989	0.5818	0.5818	0.1247	0	0.8727
Very high UDP traffic	UDP	0.5067	0.5091	0.5091	0.0986	0	1.0182
	TCP at source	0.4900	0.4364	0.4364	0.1269	0	1.1636
	TCP at sink	0.4812	0.4364	0.4364	0.1136	0	0.5818
Huge UDP traffic	UDP	0.5104	0.4364	0.4364	0.1189	0	1.0182
	TCP at source	0.4889	0.4364	0.4364	0.1285	0	1.1636
	TCP at sink	0.4802	0.4364	0.4364	0.1192	0	0.5818

The results shown in Table 5.4 indicate the benefit to TCP traffic of DiffServ. The mode values in the table specify the value for TCP throughput that occurs with the maximum frequency. Comparing the mode values from Table 5.2 (MPLS, no DiffServ) to those of Table 5.4 (no MPLS, DiffServ), we observe that there is about a fifty percent increase in TCP throughput when MPLS is disabled and DiffServ is enabled. Comparing the results of Table 5.1 (no MPLS, no DiffServ) to those of Table 5.4, we observe about a one-hundred percent increase in TCP throughput. Also, we notice from mode values in Table 5.4 that, the UDP traffic and TCP traffic each get around forty percent of the link capacity. The mean value for the TCP throughput also increased when compared to the corresponding values in Tables 5.1 and 5.2.

Figure 5.14 shows the total TCP throughput results with DiffServ enabled and MPLS disabled. It is observed that the total TCP throughput at the sink in the face of very high and huge UDP traffic levels is considerably increased when compared to the cases where DiffServ is not used. The DiffServ settings in the simulation give equal bandwidth to TCP and UDP traffic. The DiffServ settings can be altered to give preference to TCP traffic, thus increasing its throughput.

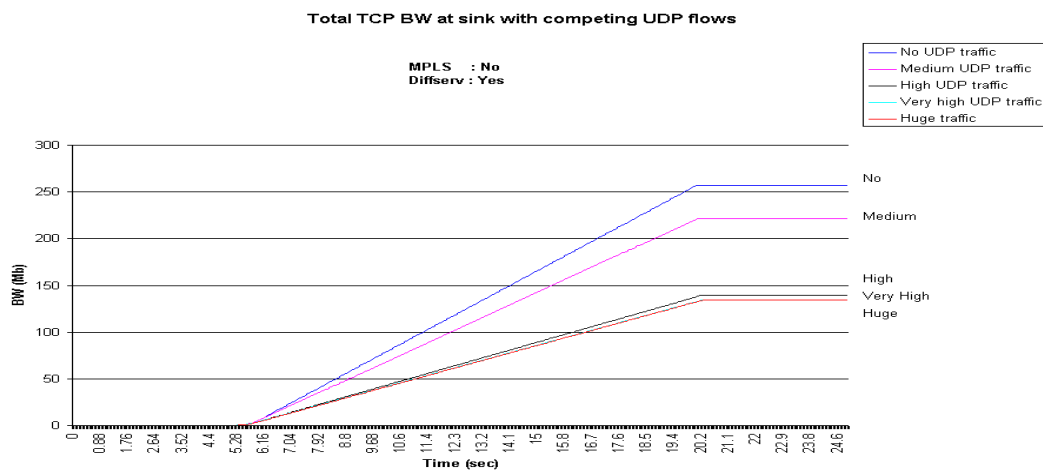


Figure 5.14. Total TCP throughput at sink for different levels of UDP traffic (DiffServ, no MPLS).

To summarize, DiffServ can help networks provide selective or preferential treatment to certain classes of traffic and can also promote fairness. This fairness or preferential treatment is provided by a combination of tagging of IP packets, traffic conditioning, and scheduling mechanisms present in DiffServ.

5.4.4. Results with MPLS and DiffServ Enabled with a Single ER-LSP

Figure 5.15 shows results when both MPLS and DiffServ are enabled. The integration of MPLS and DiffServ can be done in two ways. In the first method, the traffic flows in question are subject to DiffServ conditioning, but are routed through a single ER-LSP or CR-LSP. Another method is to use the traffic engineering features of MPLS in combination with DiffServ conditioning to provide a separate LSP for each traffic class. For example, using the second method, two ER-LSPs can be constructed with one ER-LSP carrying only EF traffic and one ER-LSP carrying only AF traffic. This leads to an efficient combination of MPLS and DiffServ to provide benefits of both traffic engineering and QoS. Figures 5.15 through 5.19 show the results for MPLS and DiffServ integration achieved by using only a single ER-LSP for routing both TCP and UDP traffic. The results are similar to results where DiffServ is enabled and MPLS is disabled. This shows that the addition of MPLS adds no observable improvements to TCP performance. Table 5.5 and Figure 5.20 show the statistics achieved by using this combination.

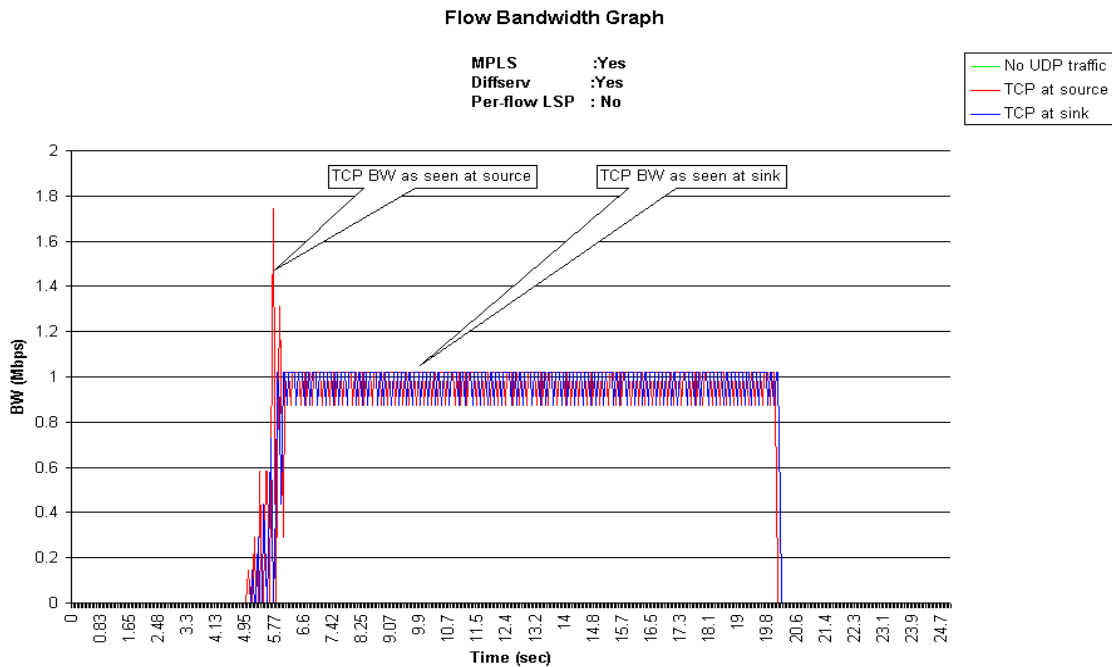


Figure 5.15. Performance with no UDP traffic (MPLS, DiffServ, single LSP).

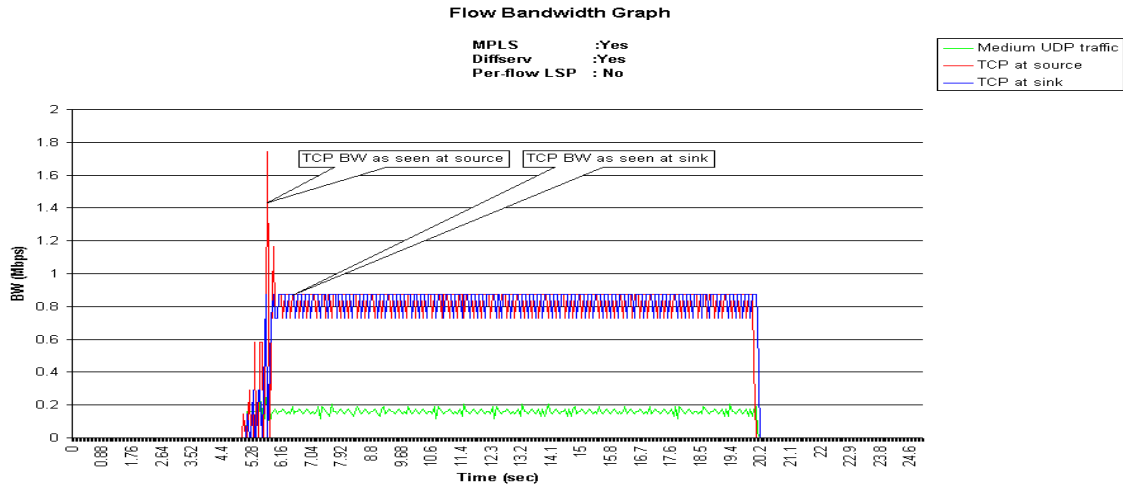


Figure 5.16. Performance with medium level of UDP traffic (MPLS, DiffServ, single LSP).

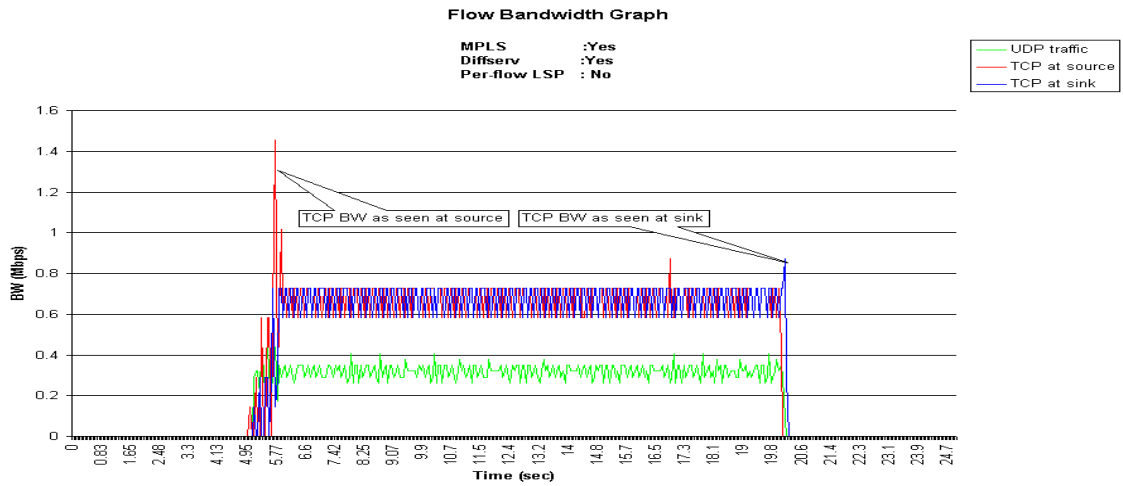


Figure 5.17. Performance with high level of UDP traffic (MPLS, DiffServ, single LSP).

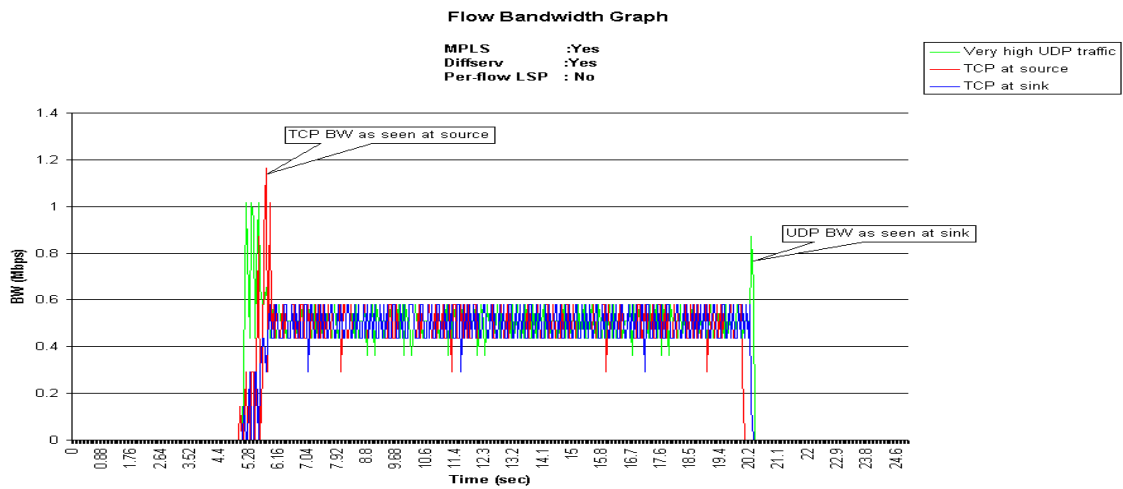


Figure 5.18. Performance with very high level of UDP traffic (MPLS, DiffServ, single LSP).

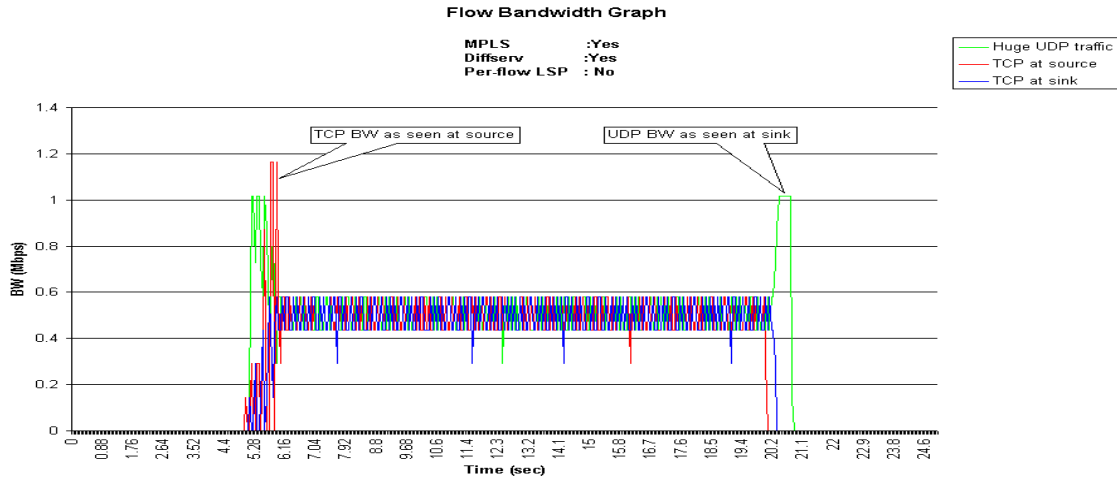


Figure 5.19. Performance with huge level of UDP traffic (MPLS, DiffServ, single LSP).

Table 5.4. DiffServ Conditioner Statistics (MPLS, DiffServ, single LSP)

DiffServ, MPLS, All-in-one ER-LSP	Low UDP	Medium UDP	High UDP	Very high UDP	high	Huge UDP
Number of packets	1801	4555	4273	3955		2872
Number of scoped packets	1769	4523	4241	3923		2846
Number of scoped CBR (EF) packets	0	3000	3000	3000		1924
Number of non-conformant CBR (EF) packets	0	0	0	0		374
Number of scoped TCP (AF11) packets	1769	1523	1241	923		922
Number of non-conformant TCP (AF) packets	1206	960	678	360		359

As can be observed from the results in Table 5.4, the number of scoped TCP packets and the number of non-conforming TCP packets remain almost the same when compared to values for the case where only DiffServ is used. The increase in the number of TCP packets in the experiments with medium UDP traffic and high UDP traffic is due to faster switching at Layer 2 with MPLS. However, when the UDP traffic increases beyond a certain level, the number of TCP packets getting through is reduced. Table 5.5 gives statistics for this experiment.

Table 5.5. Statistics for Throughput (MPLS, DiffServ, single LSP)

Diffserv MPLS All-in-one LSP	Mean (Mbps)	Median (Mbps)	Mode (Mbps)	Standard Deviation	Min (Mbps)	Max (Mbps)

Low UDP traffic	UDP	0	0	0	0	0	0
	TCP at source	0.9391	1.0182	1.0182	0.2057	0	1.7455
	TCP at sink	0.9223	1.0182	1.0182	0.2252	0	1.0182
Medium UDP traffic	UDP	0.1564	0.1600	0.1600	0.0297	0	0.2473
	TCP at source	0.8085	0.8727	0.8727	0.1768	0	1.7455
	TCP at sink	0.7940	0.8727	0.8727	0.1898	0	0.8727
High UDP traffic	UDP	0.3128	0.3200	0.3491	0.0588	0	0.4364
	TCP at source	0.6588	0.7273	0.7273	0.1453	0	1.4545
	TCP at sink	0.6470	0.7273	0.7273	0.1538	0	0.8727
Very high UDP traffic	UDP	0.5067	0.5091	0.5091	0.1007	0	1.0182
	TCP at source	0.4900	0.4364	0.4364	0.1269	0	1.1636
	TCP at sink	0.4812	0.4364	0.4364	0.1142	0	0.5818
Huge UDP traffic	UDP	0.5099	0.4364	0.4364	0.1176	0	1.0182
	TCP at source	0.4894	0.4364	0.4364	0.1379	0	1.1636
	TCP at sink	0.4807	0.4364	0.4364	0.1168	0	0.5818

The values in Table 5.5 (MPLS, DiffServ, no per-flow LSP) are almost the same as those in Table 5.4 (no MPLS, DiffServ). This shows that MPLS does not introduce any significant advantages to DiffServ networks if a single ER-LSP is used to route all types of traffic. Figure 5.20 shows the total TCP throughput achieved at the sink for this case.

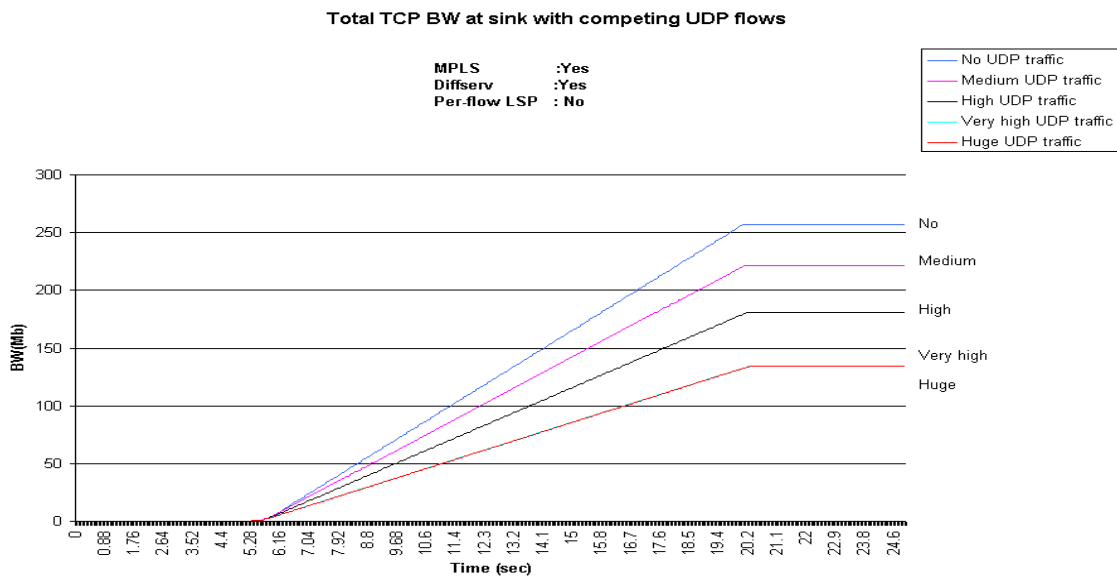


Figure 5.20. Total TCP throughput at sink for different levels of UDP traffic (MPLS, DiffServ, single LSP).

5.4.5. Results with MPLS, DiffServ Enabled and with Multiple ERLSPs

This section discusses the case where both DiffServ and MPLS are enabled. Unlike the case of Section 5.4.4, in this case, two ER-LSPs are configured, where one is assigned the EF flow and the other is assigned the AF-flow. Table 5.6 shows statistics for the DiffServ conditioner.

Table 5.6. DiffServ Conditioner Statistics (MPLS, DiffServ, per-flow LSP)

DiffServ, MPLS, per-flow ER-LSP	Low UDP	Medium UDP	High UDP	Very high UDP	Huge UDP
Number of packets	1801	4801	4801	4797	4797
Number of scoped packets	1769	4769	4769	4765	4765
Number of scoped CBR (EF) packets	0	3000	3000	3000	3000
Number of non-conformant CBR (EF) packets	0	0	0	0	1489
Number of scoped TCP (AF11) packets	1769	1769	1769	1765	1765
Number of non-conformant TCP (AF) packets	1206	1206	1206	1202	1202

The number of TCP packets that came to the DiffServ AF conditioner module specified in Table 5.6 shows the advantage provided by MPLS traffic engineering in DiffServ networks. The graphs also show that when separate ER-LSPs are used for routing different traffic flows, TCP throughput remains unaffected by the level of UDP traffic since each type of traffic follows separate MPLS paths. This feature can be provided only by path-oriented mechanisms and cannot be provided by connection-less IP routing. The graphs in Figure 5.21 through 5.25 indicate the benefits of using MPLS-based traffic engineering with multiple LSPs in conjunction with DiffServ. The TCP throughput remains at its maximum even when UDP traffic is increased to huge levels. This is because different paths are taken by these two component traffic flows.

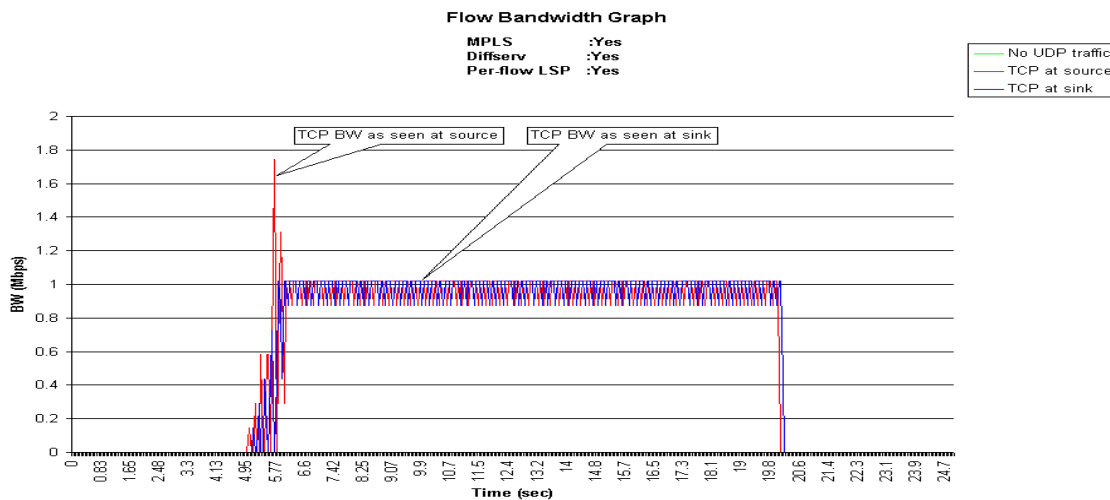


Figure 5.21. Performance with no UDP traffic (MPLS, DiffServ, per-flow LSP).

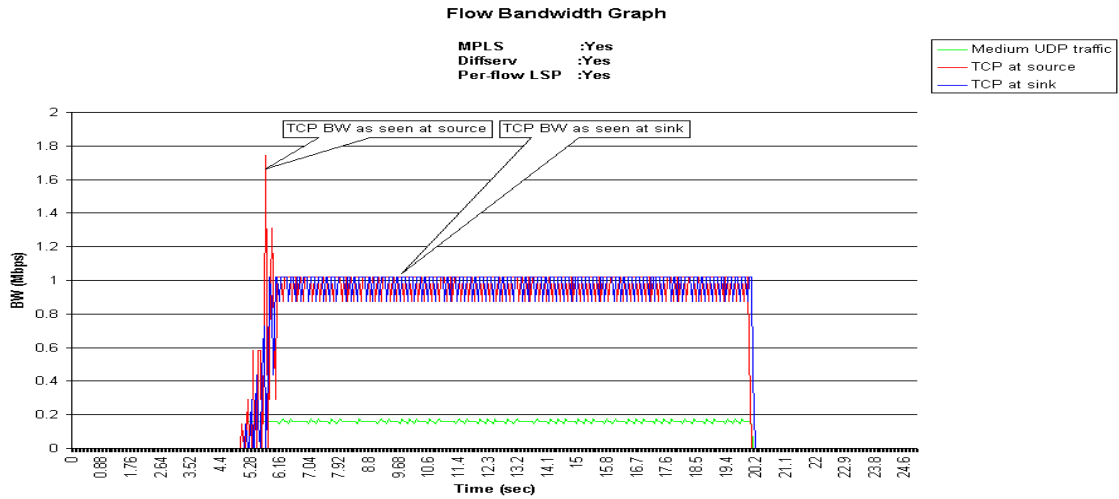


Figure 5.22. Performance with medium level of UDP traffic (MPLS, DiffServ, multiple LSPs).

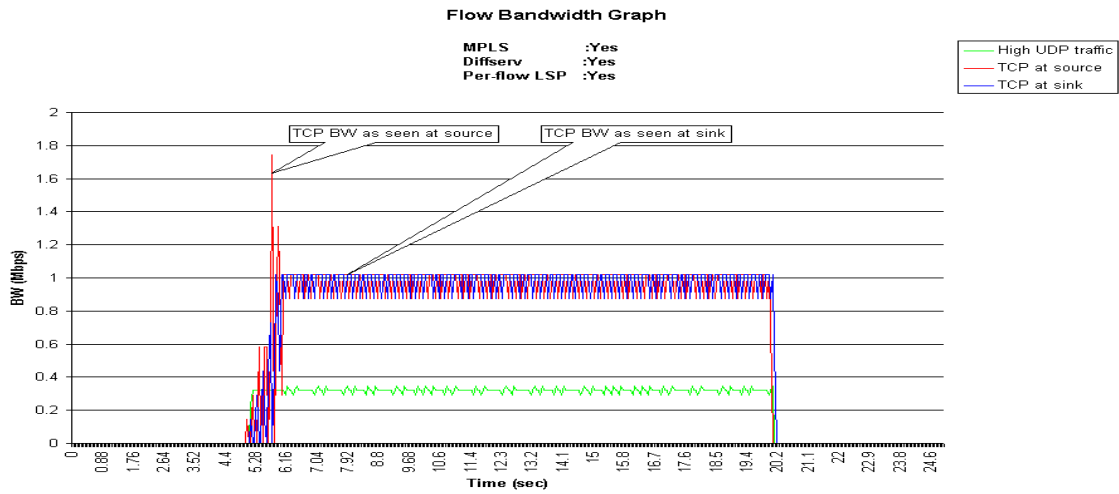


Figure 5.23. Performance with high level of UDP traffic (MPLS, DiffServ, multiple LSPs).

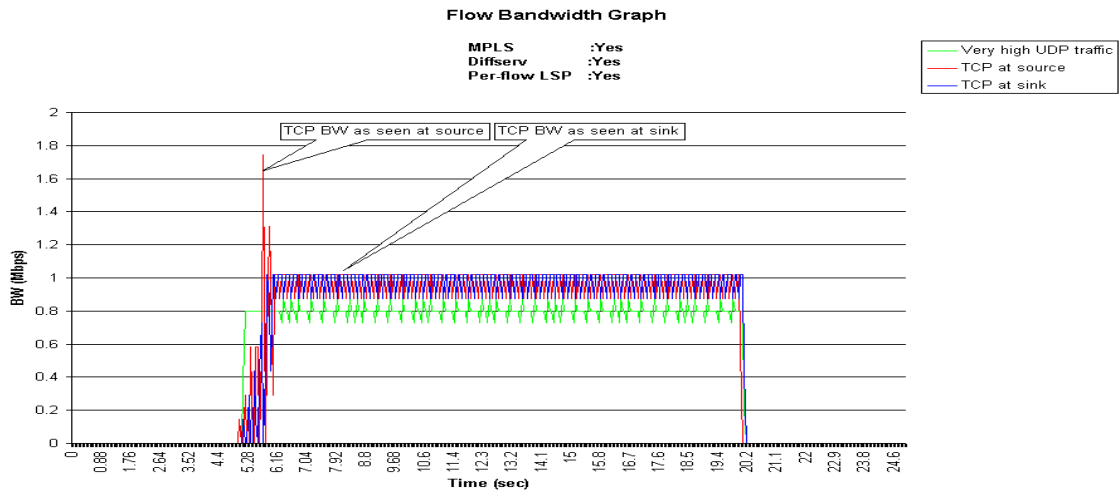


Figure 5.24. Performance with very high level of UDP traffic (MPLS, DiffServ, multiple LSPs).

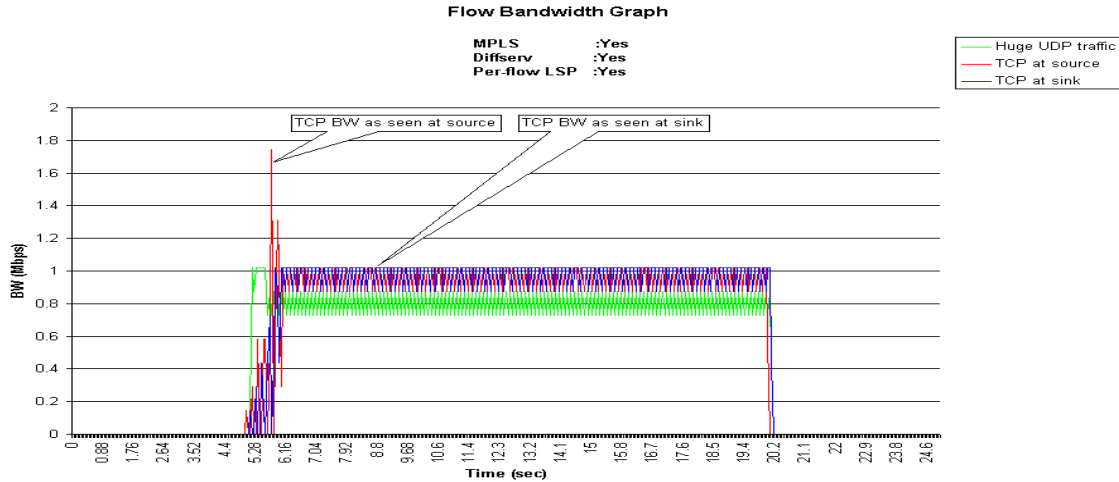


Figure 5.25. Performance with huge level of UDP traffic (MPLS, DiffServ, multiple LSPs).

Table 5.7 further indicates the advantages obtained by using multiple ER-LSPs.

Table 5.7. Statistics for Throughput (MPLS, DiffServ, per-flow LSP)

Diffserv MPLS All-in-one LSP		Mean (Mbps)	Median (Mbps)	Mode (Mbps)	Standard Deviation	Min (Mbps)	Max (Mbps)
Low UDP traffic	UDP	0	0	0	0	0	0
	TCP at source	0.9391	1.0182	1.0182	0.2057	0	1.7455
	TCP at sink	0.9223	1.0182	1.0182	0.2252	0	1.0182
Medium UDP traffic	UDP	0.1564	0.1600	0.1600	0.0247	0	0.1745
	TCP at source	0.9391	1.0182	1.0182	0.2057	0	1.7455
	TCP at sink	0.9223	1.0182	1.0182	0.2252	0	1.0182
High UDP traffic	UDP	0.3128	0.3200	0.3200	0.0493	0	0.3491
	TCP at source	0.9391	1.0182	1.0182	0.2049	0	1.7455
	TCP at sink	0.9223	1.0182	1.0182	0.2242	0	1.0182
Very high UDP traffic	UDP	0.7820	0.8000	0.8000	0.1212	0	0.8727
	TCP at source	0.9370	1.0182	1.0182	0.2056	0	1.7455
	TCP at sink	0.9202	1.0182	1.0182	0.2250	0	1.0182
Huge UDP traffic	UDP	0.7877	0.8727	0.8727	0.1401	0	1.0182
	TCP at source	0.9370	1.0182	1.0182	0.2056	0	1.7455
	TCP at sink	0.9202	1.0182	1.0182	0.2250	0	1.0182

As shown in Table 5.7, the maximum TCP throughput that is obtained with no UDP traffic is similar to that obtained even with the huge UDP traffic level. The mode values in the table also indicate the same performance.

Figure 5.26 shows the total TCP throughput observed at the sink. It can be seen that the TCP throughput achieved is the same in all cases. This configuration provides better TCP throughput than cases where MPLS or DiffServ are used separately. However, using per-flow ER-LSPs does not come without cost. MPLS signaling adds overhead and should be considered before adopting this configuration. To analyze the MPLS signaling overhead, the simulations were run with trace enabled. This trace gives an idea of the number of different types of packets that traverse the network during the simulation. Table 5.8 characterizes traffic for the case where only MPLS is enabled. The total number of bytes of MPLS LDP signaling messages that were routed in the network for the duration of simulation suggests the level of signaling overhead.

Table 5.8. Traffic Characterization (MPLS, no DiffServ)

UDP traffic Level	LDP – MPLS signaling messages	Distance Vector routing protocol messages	TCP	TCP Ack packets	CBR (UDP packets)	Total (MBytes)
Low	536	6336	10614000	10614000	0	21.23
Medium	536	6336	9144000	9144000	1800000	20.09
High	536	6336	7464000	7464000	3600000	18.53
Very High	536	6336	2444000	2444000	8974000	13.86
Huge	536	6336	3282000	3282000	10624000	17.19

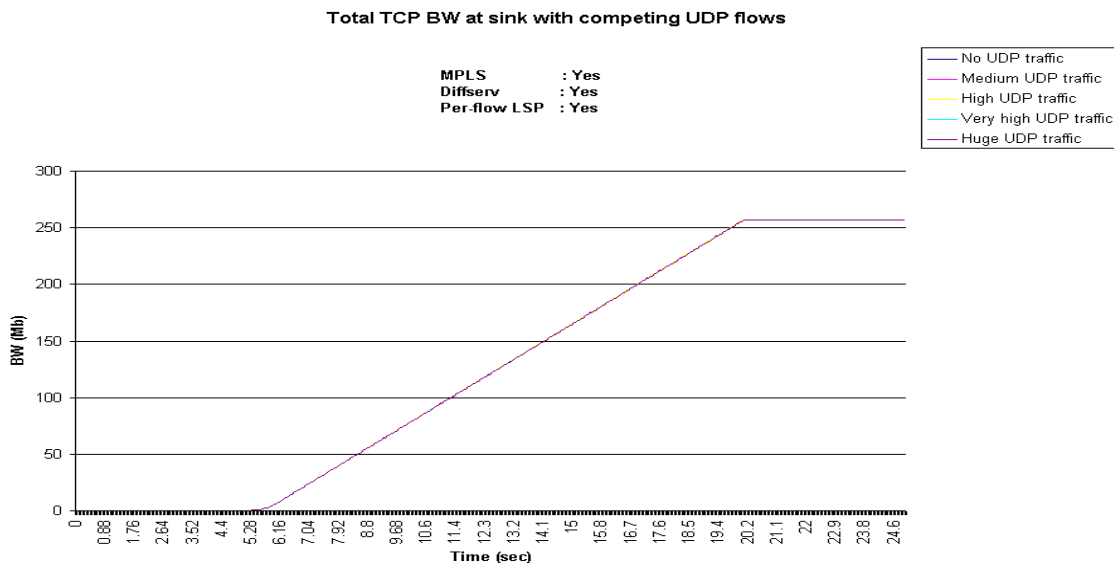


Figure 5.26. Total TCP bandwidth at the sink (MPLS, DiffServ, per-flow LSP).

Table 5.9 characterizes the network traffic when both DiffServ and MPLS are enabled and only one ER-LSP is used. Table 5.10 shows the results for the simulation run with

both DiffServ and MPLS enabled with multiple ER-LSPs. It is observed that increasing the number of ER-LSPs from one to two results in an increase in signaling traffic by a factor of about twenty for the network setup show in Figure 5.1.

Table 5.9. Traffic Characterization (MPLS, DiffServ, single LSP)

Level of UDP traffic	LDP – MPLS signaling messages	Distance Vector routing protocol messages	TCP	Ack packets	CBR (UDP packets)	Total (MBytes)
Low	536	6516	10614000	10614000	0	21.23
Medium	536	6516	9138000	9138000	1800000	20.08
High	536	6516	7446000	7446000	3600000	18.49
Very High	536	6516	5538000	5538000	6906000	17.98
Huge	536	6516	5532000	5532000	8686000	19.75

Table 5.10. Traffic Characterization (MPLS, DiffServ, multiple LSPs)

UDP traffic level	LDP – MPLS signaling messages	Distance Vector routing protocol messages	TCP	Ack packets	CBR (UDP packets)	Total (MBytes)
Low	10704	6516	10614000	10614000	0	21.24
Medium	10704	6516	10614000	10614000	2100000	23.34
High	10704	6516	10614000	10614000	4200000	25.44
Very High	10704	6516	10590000	10590000	10500000	31.69
Huge	10704	6516	10590000	10590000	12066000	33.26

5.5. Summary

From the simulation studies, one can observe that QoS mechanisms like DiffServ are needed to ensure protection of well-behaved traffic flows from rogue or ill-behaved traffic flows. A general characteristic of rogue flows is that they can easily grab bandwidth from the well-behaved flows while not responding to network congestion. Moreover, path-oriented mechanisms like MPLS are needed to efficiently integrate traffic engineering with QoS mechanisms. A simple traffic engineering example shown as part of the simulation study was to utilize different paths depending on the level of QoS expected by the traffic. Simulation studies show that MPLS and DiffServ integration yields better results than when each is used separately. This supports the claim made for the proposed architecture that MPLS and DiffServ must be integrated and must interoperate in backbone networks. Simulation results were shown to support the claim that, for this architecture and for certain types of traffic scenarios, integration of path-oriented mechanisms like MPLS and connection-less mechanisms like DiffServ is essential for consistent QoS guarantees. Results were also shown, that demonstrate DiffServ or MPLS acting separately cannot significantly improve performance.

Chapter 6. Conclusion and Future Work

This chapter concludes the thesis with a summary of the research followed by conclusions and future work.

6.1. Summary

This thesis was aimed at developing an architecture that can handle the current Internet requirements in a better way than existing ATM-based or DiffServ-based architectures. The requirements include a common QoS architecture that can handle the scalability requirements and large volume of traffic in the Internet backbone, while simultaneously supporting the varying QoS requirements of heterogeneous ISP-level networks. MPLS, with its capability to handle large volumes of traffic through traffic engineering and its support for heterogeneity, forms the basis for this architecture. However, MPLS does not have a built-in QoS infrastructure. Hence, a multi-level QoS architecture that behaves in a different way in the network periphery and in the backbone networks is proposed in Chapter 3. In the network periphery, the QoS guarantees and rules are strictly imposed using ATM-like QoS guarantees and dynamic MPLS LSPs constructed using the CR-LDP signaling protocol, thus ensuring more manageable traffic flows in the backbone. These flows are then carried through the backbone networks through a combination of pre-fabricated MPLS LSPs and DiffServ QoS guarantees. Heterogeneity in the ISP networks in the form of pockets of optical segments and, potentially, wireless segments is handled through MPLS in this architecture.

6.2. Conclusions

From the multi-level architecture proposed in Chapter 3, we can conclude that this architecture is more scalable than the present day QoS architectures. This is because, ill-behaved flows are prevented from entering the network core, thus providing a network flow that can be handled using a stateless, coarse-grained QoS architecture like DiffServ. This leads to high scalability in the network backbone, since more flows can be bundled based on similar QoS requirements and can be switched quickly using MPLS LSPs. This architecture is more fault-tolerant than present day architectures since MPLS is a path-oriented mechanism employing fast rerouting mechanisms to achieve high fault tolerance. Traffic engineering using MPLS is better for this architecture, thus preventing network “hot-spots” and the related disadvantages. This architecture is more robust since the strict QoS handling in the network periphery leads to flows that are not negatively affected due to ill-behaved flows. This architecture is also heterogeneous since MPLS can handle wireless, optical, and copper-based network segments with a high level of transparency. This architecture is “ISP-friendly” due to the presence of the distributed MPLS management entities that can recover from network failures and, also, facilitate easy billing and accounting. This architecture may also allow ISPs to recover previous investments since resources like ATM switches, FR devices, and other legacy network equipment can be upgraded to support MPLS. This section presents conclusions that can

be drawn from the simulation results and from the architecture. From the simulation results of Chapter 5, we can conclude that combining path-oriented MPLS and using DiffServ to ensure QoS guarantees for the backbone networks is mutually beneficial to both MPLS and DiffServ. We can also conclude that if MPLS and DiffServ were to act alone in the backbone networks in this architecture, the benefits gained are much less compared to that of the combination.

6.3. Future Work

There are two promising opportunities for future work. There can be improvements to the architecture itself and there can be improvements to the simulations to better validate the claims made of the architecture. Currently the proposed uses different MPLS signaling protocols in different parts of the network. This gives rise to interface issues when the two protocols meet in the network. Currently this is solved by a combination of transparency and through the use of a management entity that can “speak” both the signaling protocols. MPLS is a nascent technology. Further MPLS research can provide a solution based on a single signaling protocol that can behave according to differing requirements.

Further, this architecture is still theoretical and needs support from either simulations or small-scale implementations to better validate the claims made for this architecture. The simulation model and experiments, as presented in this thesis, are small scale with the sole aim of providing a corroborative demonstration for the claim made for this architecture, that there is benefit in combining MPLS and DiffServ in the backbone networks. This simulation uses a single topology, a single signaling protocol, and a single type of traffic, among other simplifications. The simulator itself should be improved taking into account complex interactions as found in the Internet today. Also, the simulator that was used in this research supported only CR-LDP. Valuable insight could be gathered if it were able to compare performance for similar network configurations using the RSVP-TE signaling protocol.

References

- [CiscoJun99] Cisco Systems, "Introduction: Quality of Service Overview," http://www.ieng.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcintro.htm, (current Jan. 10, 2001).
- [SDJul99] StarDust, "Introduction: What is Quality of Service?" <http://www.winsock2.com/qos/intro.htm>, (current Jan. 10, 2001).
- [Chen99] C. Shigang, "Routing Support for Providing Guaranteed End-to-End Quality of Service," Ph.D. dissertation, University of Illinois at Urbana-Champaign, May 1999.
- [RFC2386] E. Crawley, et al., "A Framework for QoS-based Routing in the Internet," RFC 2386, Aug. 1998.
- [RFC2210] J. Wroclawski, "Use of RSVP with IETF Integrated Services," RFC 2210, Sept. 1997.
- [RFC2475] S. Blake, et al., "An Architecture for Differentiated Services," RFC 2475, Dec. 1998.
- [RFC791] J. Postel, "Internet Protocol Specification," RFC 791, Sept. 1981.
- [Para2001] AllianceDataCom, "Why Frame Relay and what is an SLA?," <http://www.alliancedatacom.com/frame-relay-service-level-agreement.htm>, (current Jan.10, 2001).
- [Xipe1999] X. Xiao et al., "Internet QoS: A Big Picture," *IEEE Network*, vol. 13, no. 2, pp. 8-18, March 1999.
- [Cisco2000] Cisco Systems, "QoS Solutions Configuration Guide", http://www.ieng.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/index.htm, (current Jan.10, 2001).
- [Armi2000] A. Grenville, *Quality of Service in IP Networks: Foundations for a Multi-Service Internet*, MacMillan Technical Publishing, New York, NY, 2000.
- [Ser2000] S. Rares, "Internet QoS Signaling Protocols," Presentation, <http://www-sop.inria.fr/rodeo/rserban/sigprot.html>, (current Jan.10, 2001).
- [Enn1999] Ennovate Networks, "Network Based IP VPNs, the Role of MPLS," <http://www.ennovatenetworks.com/technology/apps/newipvpn/5.htm>, (current Jan.10, 2001).
- [Cisco2000a] Cisco Systems, "The BPX Switch: Functional Overview," http://www.ieng.com/univercd/cc/td/doc/product/wanbu/9_3_10/bpx/bpxi01.htm#xtocid1108310, (current Jan.10, 2001).
- [Bruc2000] B. Davie, et al., *MPLS: Technology and Applications*, Morgan Kaufmann Publisher, New York, NY, 2000.
- [Eric2000] E. Rosen, et al., "MPLS Architecture," work in progress, draft-ietf-mpls-arch-07, July 2000.
- [Chuc2000] Juniper Networks "MPLS: Enhancing Routing in the New Public Network," <http://www.juniper.net/techcenter/techpapers/200001.html> (current Jan.10, 2001).
- [Anoo1999] A. Ghanwani, et al., "Traffic Engineering Standards in IP Networks Using MPLS," *IEEE Communications Magazine*, vol. 37, no. 12, pp. 49- 53, Dec. 1999.

- [Dani2000] D. Awduche, et al., "Extensions to RSVP for LSP Tunnels," work in progress, draft-ietf-mpls-rsvp-lsp-tunnel-08, Feb. 2001.
- [Dani2000a] D. Awduche, et al., "Multi-Protocol Lambda Switching," work in progress, draft-awduche-mpls-te-optical-02, June 2001.
- [Pete2000] P. A. Smith, et al., "Generalized MPLS-Signaling Functional Description," work in progress, draft-ietf-mpls-generalized-signaling-01, Nov. 2000.
- [Antt2000] A. Kankkunen, et al., "VoIP over MPLS framework," work in progress, draft-kankkunen-vompls-fw-01, July 2000.
- [Kart2000] K. Muthukrishnan, et al., "A Core MPLS IP VPN Architecture," RFC 2917, Sept. 2000.
- [Cisco1996] Cisco Systems, "Policy-Based Routing," http://www.ieng.com/warp/public/732/Tech/policy_wp.htm, (current Jan. 10, 2001).
- [Bile2000] B. Jamoussi, et al., "Constraint-based LSP Setup using LDP," work in progress, draft-ietf-mpls-cr-ldp-05, March 2001.
- [Law2000] L. Roberts, "Internet Founder Ponders the Web's Future," *IT Professional*, vol. 2, no. 5, pp. 16-20, Sept. 2000.
- [Web2000] International Engineering Consortium, "MPLS Tutorial," <http://www.iec.org/tutorials/mpls/topic04.html>, (current Jan 24, 2000).
- [Luiz2001] L. DaSilva, "*Network Quality of Service*," course notes, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, Spring 2001.
- [Lmds2001] VT-LMDS Group, "LMDS at Virginia Tech," <http://www.lmds.vt.edu/vtlmds.htm>, (current March 04, 2001).
- [VT2001] VT-LMDS Group, "Virginia Tech Networking," <http://www.lmds.vt.edu/ppt/VT-Networks11-15-00.ppt> (current March 04, 2001).
- [Canet2001] CA* net 3 Optical Internet Backbone, "Optical Internet Backbone," <http://www.canet3.net/optical/optical.html> (current March 04, 2001).
- [Tom2001] T. Worster, et al., "MPLS Label Stack Encapsulation in IP," work in progress, draft-worster-mpls-in-ip-04, Feb. 2001.
- [RFC3034] A. Conta, et al., "Use of Label Switching on Frame Relay Networks Specification," RFC 3034, Jan. 2001.
- [RFC3032] E. Rosen, et al., "MPLS Label Stack Encoding," RFC 3032, Jan. 2001.
- [Mich2001] M. Behringer, "Analysis of the Security of the MPLS Architecture," work in progress, draft-behringer-mpls-security-00, Feb. 2001.
- [Tiss2001] T. Senevirathne., "Secure MPLS Domain of Interpretation for ISAKMP," work in progress, draft-tsenevir-smpls-doi-00.txt, Feb. 2001.
- [Tiss2001a] T. Senevirathne., "Secure MPLS – Encryption and Authentication of MPLS Payloads," work in progress, draft-tsenevir-smpls-01, Feb. 2001.

- [Pun2001] P. Agarwal., "TTL Processing in MPLS Networks," work in progress, draft-agarwal-mpls-ttl-00, Feb. 2001.
- [Sat2001] S. Matsushima, "TTL Processing Expansion for 1-hop LSP," work in progress, draft-satoru-mpls-1hop-lsp-00.txt, Feb. 2001.
- [Yash1999] Y. Ohba., "Issues on Loop Prevention in MPLS Networks", *IEEE Communications Magazine*, vol. 37, no. 12, pp. 64-68, Dec. 1999.
- [Mark2000] M. Shayman, et al., "Using ECN to Signal Congestion within an MPLS Domain," work in progress, draft-shayman-mpls-ecn-00, Nov. 2000.
- [Kkra1999] K. K. Ramakrishnan, et al., "A Proposal to Incorporate ECN in MPLS," work in progress, draft-ietf-mpls-ecn-00.txt, June 1999.
- [Eric2001] E. Rosen, et al., "BGP/MPLS VPNs," work in progress, draft-rosen-ietf2547bis-03, Feb. 2001.
- [Ili1999] I. Andrikopoulos, et al., "Supporting Differentiated Services in MPLS Networks," *IEEE International Workshop on Quality of Service*, pp. 207 – 215, 1999.
- [Xipe2000] X. Xiao, et al., "Traffic Engineering with MPLS in the Internet," *IEEE Network*, vol. 14, no. 2, pp. 28 – 33, March 2000.
- [Srig2000] S. Kini, et al., "Shared Backup Label Switched Path Restoration," work in progress, draft-kini-restoration-shared-backup-00, Nov. 2000.
- [Atsu2000] A. Iwata, et al., "Crankback Routing Extensions for MPLS Signaling," work in progress, draft-iwata-mpls-crankback-00, Nov. 2000.
- [Srin2000] S. Makam, et al., "Framework for MPLS-based Recovery," work in progress, draft-makam-mpls-recovery-fmwrk-01, July 2000.
- [Srin2000a] S. Makam, et al., "Protection/Restoration of MPLS Networks," work in progress, draft-makam-mpls-protection-00, April 2000.
- [KenO2000] K. Owens, et al., "A Path Protection/Restoration Mechanism for MPLS Networks," work in progress, draft-chang-mpls-path-protection-02, Nov. 2000.
- [LiMo2000] L. Mo, et al., "General Considerations for Bandwidth Reservation in Protection," work in progress, draft-mo-mpls-protection-00, July 2000.
- [Joan2000] J. Cucchiara, et al., "Definitions of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol (LDP)," work in progress, draft-ietf-mpls-ldp-mib-07, Aug. 2000.
- [Jona2000] J. P. Lang, et al., "Link Management Protocol", work in progress, draft-ietf-mpls-lmp-01, Dec. 2000.
- [Fran2000] F. Reichmeyer, et al., "COPS Usage for MPLS/TE," work in progress, draft-franr-mpls-cops-00, July 2000.
- [Dirk2001] D. Ooms, et al., "A Framework for IP Multicast in MPLS," work in progress, draft-ietf-mpls-multicast-05, Jan. 2001.
- [Dino2000] D. Farinacci, et al., "Using PIM to Distribute MPLS Labels for Multicast Routes," work in progress, draft-farinacci-mpls-multicast-03, Nov. 2000.

- [Cisco2000b] Cisco Systems, "Cisco IOS™ Software Quality of Service Solutions," http://www.ieng.com/warp/public/cc/techno/protocol/tech/qosio_wp.htm, (current March 2001).
- [Nata2000] N. Giroux, et al, *Quality of Service in ATM Networks: State-of-the-Art Traffic Management*, Prentice Hall, Upper Saddle River, New Jersey, 2000.
- [QBone2001] Internet2, "Internet2 Glossary and Style Guide," <http://www.internet2.edu/resources/i2gsg.shtml#Glossary>, (current March 2001).
- [QBone2001a] Internet2, "QBone Architecture," <http://sss.advanced.org/arch/>, (current March 2001).
- [QBone2001b] Internet2, "QBone Bandwidth Broker Architecture," <http://qbone.internet2.edu/bb/bboutline2.html>, (current March 31, 2001)
- [Webp2001] Web ProForums, "Tutorials," <http://www.iec.org/tutorials/>, (current March 2001).
- [RFC2748] D. Durham, et al., "The Common Open Policy Service Protocol," RFC 2748, Jan. 2000.
- [Cops2001] K. Ho Chan, et al., "COPS usage for Policy Provisioning (COPS-PR)," work in progress, draft-ietf-rap-pr-05, Oct. 2000.
- [Cris1998] C. Aurrecochea, et al., "A Survey of QoS Architectures," *Multimedia Systems*, Springer-Verlag, vol. 6, pp. 138-151, 1998.
- [Yako2000] Y. Rekhter, "MPLS and the Evolution of IP Routing," <http://www.ail.gmu.edu/MPLS2000/proceedings/MONDAY/03-Rekhter.pdf>, (current March 24, 2001).
- [Prav1999] P. Bhaniramka, et al., "Quality of Service using Traffic Engineering over MPLS: An Analysis," work in progress, draft-bhani-mpls-te-anal-00, March 1999.
- [Dani2001] D. O. Awduche, et al., "RSVP –TE: Extensions to RSVP for LSP Tunnels," work in progress, draft-ietf-mpls-rsvp-lsp-tunnel-08, Feb. 2001.
- [Paul1997] P. White, "RSVP and Integrated Services in the Internet: a tutorial," *IEEE Communications Magazine*, vol. 35, no. 5, pp. 100-106, May 1997.
- [RFC3036] L. Anderson, et al., "LDP Specification," RFC 3036, Jan. 2001.
- [Chuc2000a] C. Semeria, "RSVP Signaling Extensions for MPLS Traffic Engineering," Juniper Networks, <http://www.juniper.net/techcenter/techpapers/200006-07.html>, (current March 26, 2001).
- [Tele2000] S. Masud "MPLS: Trust but Verify," <http://www.telecomsmag.com/issues/200008/tcs/mpls.html>, (current March 25, 2001) Telecommunications Online.
- [Hari1998] H. Balakrishnan, "Improving TCP/IP Performance over Wireless Networks," <http://www.cs.berkeley.edu/~hari/papers/snoop.html>, (current March 27,2001), 1998.
- [Birk1997] B. S. Bakshi, et al., "Improving Performance of TCP over Wireless Networks," *Int'l Conf. On Distributed Computing Systems*, pp. 365-374, 1997.
- [Greg2000] G. Wright, et al., "CR-LDP Extensions for Interworking with RSVP-TE," work in progress, draft-wright-mpls-crlldp-rsvp-te-iw-00, <http://www.watersprings.org/links/mlr/id/draft-wright-mpls-crlldp-rsvp-te-iw-00.txt>, (current March 27, 2001), March 2000.
- [Fran2001] F. Le. Faucheur, et al., "MPLS Support of Differentiated Services," work in progress, draft-ietf-mpls-diff-ext-08, Feb. 2001.

- [Petr2000] P. Aukia, et al., "RATES: A Server for MPLS Traffic Engineering," *IEEE Network*, vol. 12, no. 2, pp. 34 – 41, March 2000.
- [Eric2000a] E. Osborne, "How to Use MPLS TE to Dynamically Find Optimum Routes through Large Networks," <http://www.cisco.com/warp/public/784/packet/oct00/p83-cover.html>, (current March 2001), 2001.
- [RFC 2702] D. Awduche, et al., "Requirements for Traffic Engineering Over MPLS," RFC 2702, Sept. 1999.
- [Rob1997] R. Guerin, et al., "QoS Routing Mechanisms and OSPF Extensions," RFC 2676, Aug, 1999.
- [Mura2000] M. Kodialam, et al., "Minimum Interference Routing with Applications to MPLS Traffic Engineering," *Proc. INFOCOM*, pp. 884-893, March 2000.
- [Dave2001] D. Katz, et al., "Traffic Engineering Extensions to OSPF," work in progress, draft-katz-yeung-ospf-traffic-04, Aug. 2001.
- [Sudh2001] S. Dharanikota, et al., "OSPF, IS-IS, RSVP, CR-LDP extensions to support inter-area traffic engineering using MPLS TE," work in progress, draft-dharanikota-interarea-mpls-te-ext-01, 2001.
- [Fran2000a] F. L. Faucheur, et al., "Extensions to IS-IS, OSPF, RSVP, and CR-LDP for support of diff-serv-aware MPLS Traffic Engineering," work in progress, ospf-diff-te-00, July 2000.
- [RFC 2370] R. Coltun., "The OSPF Opaque LSA option," RFC 2370, July 1998.
- [Hui1991] H. Zhang, et al., "Comparison of Rate-Based Service Disciplines," *ACM SIGCOMM*, pp. 113-122, 1991.
- [Cisco1999] Cisco Systems, "Migrating to Differentiated Services – Today," *Cisco Packet Magazine*, 1999.
- [Srin2001] S. Keshav, "Flow Control," http://www.cs.cornell.edu/home/skeshav/book/slides/flow_control/ppframe.htm, (current March 27, 2001).
- [Andr1998] A. T. Campbell, et al., "A Continuous Media Transport and Orchestration Service," *Proc. ACM SIGCOMM*, Baltimore, pp. 99-110, Aug. 1992.
- [Neil2001] N. Harrison, et al., "OAM Functionality for MPLS Networks," work in progress, draft-harrison-mpls-oam-00, Feb. 2001.
- [Bija2000] B. Jabbari, et al., "Label Switched Packet Transfer for Wireless Cellular Networks," *IEEE Wireless Comm. and Networking Conf.*, 2000.
- [Davi1998] D. Kidston, et al., "Comma, a Communication Manager for Mobile Applications," *Proc. 10th Annual Intl. Conf. On Wireless Comm.*, pp. 103-116, July 1998.
- [Marc1997] M. Lioy, et al., "Providing Network Services at the Base Station in a Wireless Networking Environment," *Proc. Wireless '97 Conf.*, pp. 29-39, July 1997.
- [Pinn2000] Pinnacle Comm. Inc., "Understanding Wireless WAN Communications," <http://www.pinnaclecomm.com/wireless/wirelessinfo.html>, (current April 3, 2001), 2000.
- [Jun2001] Jun K. Choi, et al., "Extension of LDP for Mobile IP Service through the MPLS Network," work in progress, draft-choi-mobileip-ldpext-01, Feb. 2001.

- [Javi1998] J. Gomez, et al., "The Havana Framework for Supporting Application and Channel Dependent QoS in Wireless Networks," *Proc. Seventh Annual Int'l Conf. On Network Protocols*, pp. 300-310, 1998.
- [Seou2000] S. Lee, et al., "INSIGNIA: An IP-Based Quality of Service Framework for Mobile ad Hoc Networks," *Journal of Parallel and Distributed Computing*, vol. 60, pp. 374-406, 2000.
- [Pete2001] P. A. Smith, et al., "Generalized MPLS- Signaling Functional Description," work in progress, draft-ietf-mpls-generalized-signaling-02, March 2001.
- [Kire2000] K. Komella, et al., "LSP Heirarchy with MPLS TE," work in progress, draft-ietf-mpls-lsp-hierarchy-00, July 2000.
- [Osam2001] O. Aboul-Magd, et al., "Automatic Switched Optical Network (ASON) and its Related Protocols", work in progress, draft-aboulmagd-ipo-ason-00, Feb. 2001.
- [Pete2001a] P. A. Smith, et al., "GMPLS Architecture," work in progress, draft-many-gmpls-architecture-00, work in progress, Feb. 2001.
- [Pete2001b] P. A. Smith, et al., "Generalized MPLS- CR-LDP Extensions", draft-ietf-mpls-generalized-cr-ldp-01, March 2001.
- [Pete2001c] P. A. Smith, et al., "Generalized MPLS- RSVP-TE Extensions," work in progress, draft-ietf-mpls-generalized-rsvp-te-01, March 2001.
- [Bala2000] B. Rajagopalan, et al., "IP over Optical Networks: Architectural aspects," *IEEE Communications Magazine*, vol. 38, no. 9, pp. 94-102, Sept. 2000.
- [Greg2000a] G. Bernstein, et al., "IP-Centric Control and Management of Optical Transport Networks", *IEEE Communications Magazine*, vol. 38, no. 10, pp. 2-8, Oct. 2000.
- [Osam2001a] O. Aboul-Magd, et al., "Signaling Requirements at the Optical UNI," work in progress, draft-bala-mpls-optical-uni-signaling-01, May 2001.
- [John2000] J. Yu, et al., "RSVP Extensions in Support of OIF Optical UNI Signaling," work in progress, draft-yu-mpls-rsvp-oif-uni-00, May 2001.
- [Osam2001b] O. Aboul-Magd, et al., "LDP Extensions for Optical UNI Signaling," work in progress, draft-ietf-mpls-ldp-optical-uni-00, April 2001.
- [Darr2000] D. Freeland, et al., "Considerations on the development of an Optical Control Plane," work in progress, draft-freeland-octrl-cons-01, Nov. 2000.
- [Greg2000b] G. Bernstein, et al., "Some Comments on GMPLS and Optical Technologies," work in progress, draft-bernstein-gmpls-optical-00, Nov. 2000.
- [Kire2001] K. Kompella, et al., "LSP Heirarchy with MPLS TE," work in progress, draft-ietf-mpls-lsp-hierarchy-02, Aug. 2001.
- [Sall1993] S. Floyd, et al., "Random Early Detection for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol.1, no.4, pp.397-413, Aug. 1993.
- [RFC2990] G. Huston., "Next Steps for the IP QoS Architecture," RFC 2990, Nov. 2000.
- [Data2000] Data Connection, "MPLS Traffic Engineering, A Choice of Signaling Protocols," <http://www.dataconnection.com/mpls/mplswpd1.htm>, (current Apr. 07, 2001) 2000.

- [Mich2001] M. Behringer, "Analysis of the security of the MPLS Architecture," work in progress, draft-behringer-mpls-security-00, Aug. 2001.
- [Pete2001a] P. A. Smith et al., "Generalized MPLS signaling – RSVP-TE Extensions," work in progress, draft-ietf-mpls-generalized-rsvp-te-01, March 2001.
- [Gera2001] G. Ash, et al., "LSP Modification using CR-LDP," work in progress, draft-ietf-mpls-crlsp-modify-03, Sept. 2001.
- [Vish2000] V. Sharma, et al., "Extensions to RSVP-TE for MPLS Path Protection," draft-chang-mpls-rsvpte-path-protection-ext-01, work in progress, Nov. 2000.
- [Keno2000a] K. Owens, et al., "Extensions to CR-LDP for MPLS Path Protection," work in progress, draft-owens-crlsp-path-protection-ext-00, Nov. 2000.
- [Pete2000a] P. A. Smith, et al., "Improving Topology Data Base Accuracy With LSP Feedback via CR-LDP," work in progress, draft-ashw-mpls-te-feed-01, Aug. 2000.
- [RFC 2749] S. Herzog, et al., "COPS Usage for RSVP," RFC 2749, Jan. 2000.
- [RFC 2750] S. Herzog, et al., "RSVP Extensions for Policy Control," RFC 2750, Jan. 2000.
- [NS] Network Simulator, "Network Simulator – ns-2," <http://www.isi.edu/nsnam/ns/index.html> (current Apr. 29, 2001).
- [NSa] MIT Object Tcl, "otcl," <http://www.isi.edu/nsnam/otcl/> (current Aug. 05, 2001).
- [Opnet] Optimum Network Performance, "OPNET," <http://www.mil3.com> (current Apr. 29, 2001).
- [MNS] MPLS Network Simulator, "MPLS Network Simulator ver.2.0," <http://www.raonet.com/introduction.shtml> (current Apr. 29, 2001).
- [Sean2001] S. Murphy, "The ns MPLS/DiffServ Patch," <http://www.teltec.dcu.ie/~murphys/ns-work/mpls-diffserv/> (current Apr. 29, 2001).
- [Kath2001] K. Nichols, "An Opinionated View of the Current State of IP Differentiated Services," ftp://ftp-eng.cisco.com/ftp/kmn-group/docs/kmn_ucbmm.pdf (current 29 Apr 2001).
- [Ragh2001] S. Raghavan, "DiffServ and MPLS: Concepts and Simulation," http://csgrad.cs.vt.edu/~sraghava/report_raghavan.pdf (current July 25, 2001).
- [Jain1991] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*, John Wiley and Sons, New York, NY, 1991.
- [Nsb] NS Documentation, "TCP Agents," <http://www.isi.edu/nsnam/ns/doc/node301.html> (current Sept. 16, 2001).

Appendix. Glossary

ATM	Asynchronous Transfer Mode
BB	Bandwidth Broker
BDI	Backward Defect Identifier
COPS	Common Open Policy Service
CR-LDP	Constraint Routed Label Distribution Protocol
DiffServ	Differentiated Services
FDI	Forward Defect Identifier
FSC	Fiber Switch Capable
FWNN	Fixed Wireless Network Node
LER	Label Edge Router
LSC	Lambda Switch Capable
LSR	Label Switch Router
MME	MPLS Management Entity
MPLS	MultiProtocol Label Switching
NS	Network Simulator
PSC	Packet Switch Capable
RAA	Resource Allocation Answer
RAR	Resource Allocation Request
RSVP-TE	Resource Reservation Protocol with Traffic Engineering extensions
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing

Vita

Srihari Raghavan hails from Madras, a city in Southern part of India. He graduated in 1997 with a Bachelor of Engineering degree in Computer Science from the University of Madras. He was an engineer at the Cisco Systems' offshore development facility in Madras, before joining the graduate program in Computer Science at Virginia Tech. He will be graduating with a Master of Science degree in Computer Science in December 2001.