MULTIPATH BASED TRAFFIC ENGINEERING IN MPLS NETWORKS

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

İbrahim Hökelek

Sep 2002

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Nail Akar (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Erdal Arıkan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Ezhan Karaşan

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

MULTIPATH BASED TRAFFIC ENGINEERING IN MPLS NETWORKS

İbrahim Hökelek
M.S. in Electrical and Electronics Engineering
Supervisor: Assist. Prof. Dr. Nail Akar
Sep 2002

Traffic Engineering (TE) is the process of controlling how traffic flows through a network so as to optimize resource utilization and network performance. In this thesis, we propose a new TE architecture for IP networks with Multi Protocol Label Switching (MPLS) backbones. In this architecture, two (as opposed to single) Label Switched Paths (LSPs), one being the primary LSP and the latter being the secondary LSP, are established among every pair of IP routers located at the edge of an MPLS cloud. In the data plane of the core MPLS Label Switched Routers (LSR), the architecture suggests traffic from primary LSPs to be served with a higher priority than traffic from secondary LSPs. This prioritization in the data plane is to avoid the so-called "knock-on" effect that arises in the context of load balancing systems. Traffic is then split between the two LSPs using the feedback information received from the network in the form of an "available bandwidth". The most challenging requirement regarding to traffic splitting schemes is that packet mis-ordering within an individual traffic flow should be avoided. The proposed traffic engineering architecture in this thesis handles this problem using the idea of burst routing. In burst routing, two successive bursts can be forwarded independently over the primary and the

secondary LSPs if the difference between their arrivals in time is large enough to ensure that packet misordering would not take place. With this architecture, we show that using multiple paths provides significantly and consistently better results than the case of a single LSP in terms of overall throughput.

Keywords: Traffic Engineering, MPLS, Multipath, Traffic Splitting, Load Balancing.

ÖZET

MPLS AĞLARINDA ÇOK YOL TABANLI TRAFİK MÜHENDİSLİĞİ

İbrahim Hökelek
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Assist. Prof. Dr. Nail Akar
Eylül 2002

Trafik mühendisliği ağ üzerinden trafik akışlarının nasıl geçmesi gerektiğini belirleyen, böylece kaynakların en iyi şekilde kullanılmasını ve ağ performansının en iyi olmasını sağlayan bir işlemdir. Bu tezde omurga olarak Çoklu Protokollü Etiket Anahtarlama'nın (Multi Protocol Label Switching-MPLS) kullanıldığı IP ağları için yeni bir trafik mühendisliği mimarisi önerilmektedir. Bu mimaride MPLS bulutunun kenarlarında bulunan her bir IP yönlendirici çifti arasında iki (tek yol yerine) Etiket Anahtarlamalı Yol (Label Switched Path-LSP), biri asıl LSP ve diğeri ikincil LSP, kurulmaktadır. Önerilen mimari, MPLS Etiket Anahtarlamalı Yönlendirici'lerin (Label Switched Router-LSR) veri düzleminde asıl LSP'ye ait trafiğe ikincil LSP'ye ait trafiğe göre daha öncelikli davranmasını önermektedir. Veri düzlemindeki bu önceliğin nedeni yük dengeleme sistemlerinde oluşan ve "knock-on" etkisi olarak bilinen etkiden kurtulmaktır. Trafik ağdan mevcut bantgenişliği formunda alınan geri besleme bilgisi kullanılarak iki LSP arasında bölünür. Trafik bölme probleminin en zor kısmı tek bir trafik akımı içerisindeki paketlerin hedef düğüme sıralamalarının bozulmadan iletilebilmesidir. Bu tezde önerdiğimiz trafik mühendisliği mimarisinde, eğer arka arkaya gelen paketler arasındaki zaman farkı paketlerin

hedef düğüme sıralamalarının bozulmadan iletilebileceğini garantileyecek kadar büyükse, bu paketler birbirinden bağımsız olarak asıl LSP ya da ikincil LSP üzerinden gönderilebilmektedir. Bu mimari ile iki yol kullanılmasının tek yol durumuna göre toplam iş yapma yeteneği açısından istikrarlı olarak ve önemli derecede daha iyi sonuçlar sağladığı gösterilmektedir.

Anahtar kelimeler: Trafik Mühendisliği, MPLS, Çoklu yol, Trafik Bölme, Yük Dengeleme.

ACKNOWLEDGMENTS

I gratefully thank my supervisor Assist. Prof. Dr. Nail Akar for his supervision, guidance, and suggestions throughout the development of this thesis.

Contents

List of Figures

List of Tables

To My Family ...

Chapter 1

Introduction

Internet Service Providers (ISPs) face challenging problems today due to the unprecedented growth in Internet traffic demands. To cope with such traffic growth scenarios, ISPs need to find ways to sustain an acceptable quality of service (QoS) for their customers and they also need to achieve operational efficiencies, if they can, to reduce their costs. A report from the U.S. Department of Commerce [1] suggests that the rate at which the Internet has been adopted has surpassed all other technologies preceding it, including radio, television, and the personal computer. Three technical instruments can be employed by ISPs to cope with the challenge of Internet growth:

- network planning,
- capacity engineering,
- traffic engineering.

Network planning deals with the abstract structure of networks, the components of the network, their functions, and the relations between them. It is a long-term process used to develop an architecture and to build a physical network around this architecture for long-term traffic growth. The second instrument that

can be employed to respond to traffic growth is capacity engineering. It is also a long-term process and aims to establish capacity according to changing traffic needs. However, capacity expansion is a preventive engineering technique and since the capacity establishment process requires a fairly long period of time (e.g., in the order of months), this technique may lead to overbooked links and therefore wasted investments, in case the estimate for capacity needs is sceptical.

The third instrument is to apply Traffic Engineering (TE) which is a short-term process. Internet TE is defined as the set of mechanisms required for enhancing the performance of an operational network. This is accomplished by addressing traffic oriented performance requirements related to delay, delay variation, loss, and throughput, while utilizing network resources economically and reliably. TE should therefore be viewed as the process of controlling how traffic flows through a network so as to optimize resource utilization and network performance [2]. TE capabilities are used mainly to:

- provide ISPs precise control over the placement of traffic flows,
- balance the traffic load on the various links, routers, and switches in the network so that none of these components is overutilized,
- provide more efficient use of available aggregate bandwidth,
- enhance the traffic-oriented performance characteristics of the network.

TE mainly applies to backbone networks and the methodology for traffic engineering is strictly dependent on the underlying backbone architecture. Therefore, we first overview the alternative backbone architectures currently used in the Internet, in the following section.

1.1 Network Architectures

In this section, we will briefly explain three commonly used backbone network architectures, namely

- Pure Internet Protocol (IP) networks,
- Asynchronous Transfer Mode (ATM) networks,
- Multiprotocol Label Switching (MPLS) networks.

1.1.1 Pure IP Networks

The Internet Protocol (IP) is the most widely-used method for transporting data within and between communications networks. IP provides a connectionless, unreliable, best-effort packet delivery system to applications. It also defines the basic data unit of data transfer and also performs the routing function. IP is relied upon by higher-level protocols to create more complex services . The two principal transport layer protocols that make use of IP are User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) [3].

IP centers around the concept of a packet which is also known as a datagram. Datagrams of different sizes are used by IP for transmission over networks with different maximum data unit sizes. An IP datagram consists of a header and a payload. An IP header contains the following fields: version number of IP, size of packet, source and destination addresses, counter controlling lifetime of packet, service type, etc. Source and destination addresses are globally unique, 32-bit numbers called IP addresses. An IP address is divided into two main parts: the network number and the host number. The network number identifies a network and must be assigned by the Internet Network Information Center (InterNIC) if the network is to be part of the Internet. An Internet Service Provider (ISP)

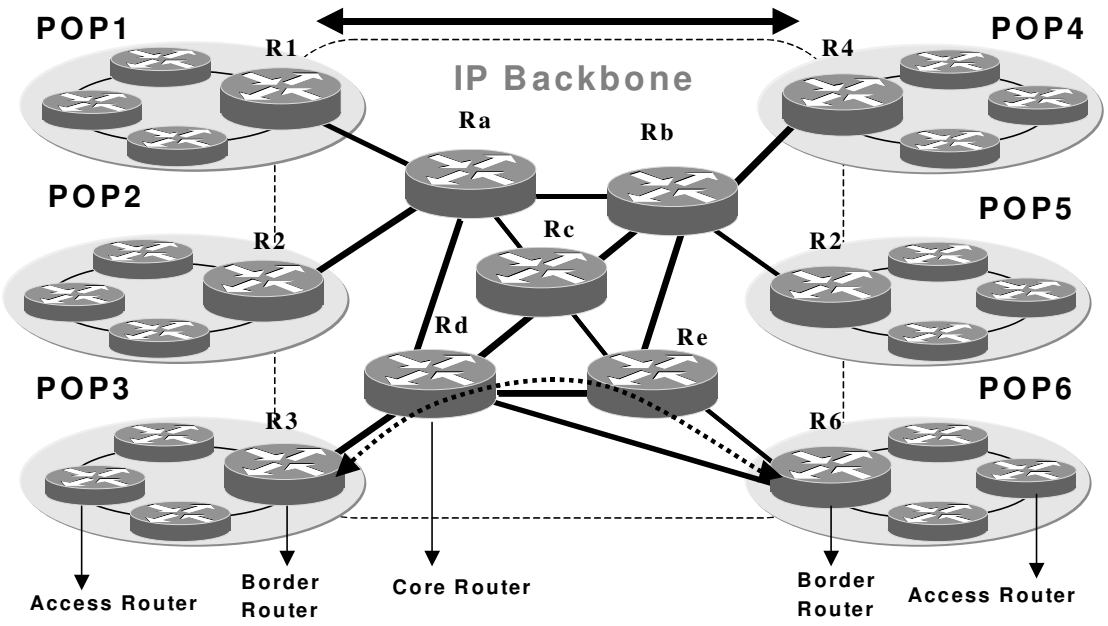

Figure 1.1: A Pure IP Network.

can obtain blocks of network addresses from the InterNIC and can itself assign address space as necessary. The host number identifies a host on a network and is assigned by the local network administrator. Globally unique addresses permit IP networks anywhere in the world to communicate with each other.

Internet Service Providers (ISPs) provide IP connectivity to their customers. ISPs have connection points, called Point of Presence (POP) in which they receive traffic from their customers or exchange traffic with other ISPs. In each POP, there will be many customer access routers and also one (or two for redundancy purposes) border (edge) router. The connectivity among the core routers is provided by a *backbone network*. In Figure 1.1, an ISP network is shown in which such a connectivity among core routers is provided through point-to-point links (edge-core or core-core). This connectivity is called a "partial mesh" as opposed to a "full mesh" since there are router pairs in this figure among which there is no physically established link. For example, any traffic originating at POP3 and destined to POP6 has to go through other routers in the backbone network. Since the backbone network does not have switching capability which

is available in ATM and MPLS networks and since it consists of only IP routers, such a backbone network architecture is called a "pure IP network".

In order to transfer packets from a sending host to a destination host, the network layer must determine the path or route that the packets are to follow. This is the job of the network layer routing protocol such as OSPF [4], IS-IS [5] and BGP-4 [6]. The purpose of a routing algorithm is simple: given a set of routers, with links interconnecting the routers, a routing algorithm finds a shortest path from the source to the destination. This path is shortest in the sense that the sum of the link costs (or weights) comprising the path is smaller than the same sum for other paths between the source and the destination. As an example, in Figure 1.1, assume that all the link costs are set to unity. Then, the shortest path between R3 and R6 is the path shown by the dashed line. Each router runs a shortest path algorithm like Dijkstra [3] to keep a routing table which consists of destination address/next hop pairs. When a packet arrives at a router, the next hop is calculated by matching the destination address within the IP header with an entry in the current node's routing table. This type of routing is therefore called "destination based routing" which uses a longest prefix match algorithm at every hop. An IP datagram originating at the access routers of POP3 and destined to POP6 is first forwarded to R3. R3 runs a longest prefix match algorithm and forwards this datagram to Rd. Rd also runs a longest prefix match algorithm and forwards this datagram to Re. Using a similar procedure at the router Re, the IP datagram reaches R6. Because routing tables can be modified at any time, packets sent from one node to another may follow different paths through the network and may arrive out of order.

In IP backbone networks, core routers run a longest prefix match algorithm for every packet, which is a relatively expensive task compared to switches that do simple table lookups. Moreover, traffic is typically carried through the shortest path even when this path is congested. On the other hand, in "explicit routing",

traffic is forced to flow through arbitrary paths, not necessarily the shortest path. Providing features for explicit routing is a key problem for IP networks. Explicit routing can be carried out using a technique called source routing [3] by adding the information about the explicit route to the packet header. However, this task is not only computationally intensive but also requires a substantial bandwidth overhead for carrying the source route information. To overcome these problems, switched architectures like Asynchronous Transfer Mode (ATM) [7, 8] and Multi-Protocol Label Switching (MPLS) [9, 10, 11], are proposed as backbone architectures for IP networks. Next, we overview these two architectures, starting first with ATM.

1.1.2 ATM Networks

Broadband Integrated Services Digital Network (B-ISDN) [12] is conceived to provide high-speed data,voice, and video communications to end users. ATM is the selected high-speed technology to deliver the B-ISDN service [7]. ATM has the capability for carrying different types of traffic such as voice, video and data with high performance and QoS (Quality of Service) support. It combines the switching and multiplexing functions over communication networks, which is well suited for bursty network traffic.

ATM is a cell-oriented technology and packets in the ATM network (cells) have a fixed size that is consists of a 48-octet data and a 5-octet header. This short and fixed cell size results in a performance increase in hardware switching, forwarding and routing. This makes ATM a high-speed technology compared with pure IP networks. Also, fixed cells ensure that time-critical information such as voice or video is not adversely effected by long size packets or data frames.

ATM is a connection-oriented technology and requires end-to-end connections to be established before the traffic can start flowing. ATM has two types of connections:

Virtual Circuit (VC): A VC is a virtual channel that carries cells from one end user to another end user in an order. All the cells in the VC will follow the same route resulting in a high-speed connection. VCs can be dynamically configured using signaling as Switched Virtual Circuits (SVC) or statically configured as Permanent Virtual Circuits (PVC). At every interface in an ATM network, each VC using that interface is locally associated with a common unique identifier referred to as the Virtual Channel Identifier (VCI).

Virtual Path (VP): A VP can be thought as a group of VCs bundled together. A VP can be established between two end interfaces in the ATM domain. In this case all the traffic using the VP will follow the same specified path. As an example many different VCs of different users between two ATM switches can be bundled together as a single VP. Virtual Path Identifier (VPI) is used to identify a virtual path.

To set up a connection, the source node has to exchange control information such as destination addresses and bandwidth requirements with the network using a signaling protocol. User Network Interface (UNI) [13] and Private Network to Network Interface (PNNI) [14] standards specify how to establish VCs, bandwidth reservation, flow control, and policing. Once a connection in the form of a VC or a VP is established, the connection is assigned a local VPI/VCI value at the user-network interface. Every cell to be transmitted to the destination is then sent to the network by the user with the assigned VPI/VCI value included in the header of the cell. The network nodes(i.e., ATM switches) then use this VPI/VCI value to forward the cells to the destination by swapping the VPI/VCI values of cells and forwarding these cells onto the right interfaces. Thus the

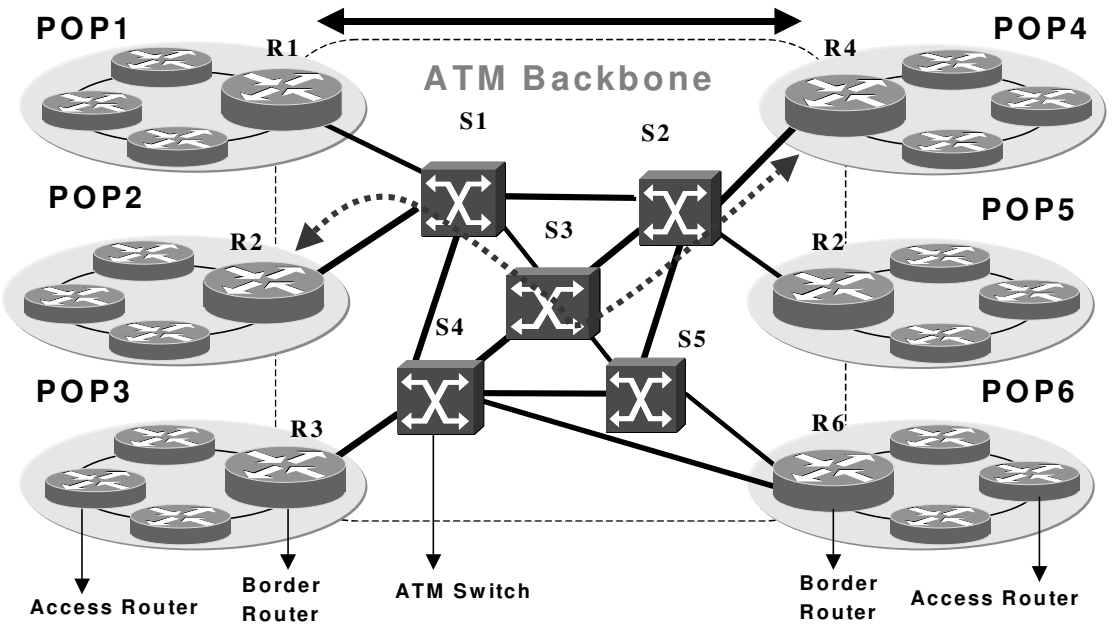

Figure 1.2: An ATM Network.

destination address itself is no longer needed by the network during the data transfer phase. This technique is generically known as *label swapping*.

The typical application of ATM is, as a backbone technology, to transport IP traffic which is called the "IP over ATM" model shown in Figure 1.2. In this figure, ATM switches are used in the backbone to provide connectivity between the six border routers. Any traffic originating at POP2 and destined to POP4 has to go through a virtual channel (VC). This VC can use a longer path (dashed line) instead of the shortest path by using explicit routing. The border router R2 examines the destination address of the IP datagram and indexes its routing table and determines the IP address of the next border router. Then, R2 indexes an ATM Address Resolution Protocol (ARP) table with the IP address of the next border router and determines the ATM address of the next border router and passes the IP datagram to the ATM layer by using ATM Adaptation Layer (AAL5) and segmentation of the datagram into ATM cells [8]. From now, it is the job of the ATM transport network to deliver the ATM cells to their destination

by using a VC between the source and the destination. At the receiving side of the VC, these cells are reassembled back into the original datagram.

ATM Forum UNI 4.0 specification [13] dictates five service classes: Constant Bit Rate (CBR), Variable Bit Rate Non-Real Time (VBR-NRT), Variable Bit Rate Real Time (VBR-RT), Available Bit Rate (ABR), and Unspecified Bit Rate (UBR). Of these, CBR and VBR classes are for real-time applications such as voice and video. They are guaranteed delivery services where low transit delay and jitter are required. However, ABR and UBR are designed for best-effort data networks where delay is not a critical requirement.

In the UBR service, users negotiate only their Peak Cell Rates (PCR) when setting up the connection. Then, they can send bursts of cells as desired at any time at a rate that does not exceed the PCR. If the total traffic at a switch exceeds the output capacity, the excessive traffic will be dropped by the switch.

The ABR service plays a crucial role in the development of this thesis. Therefore, the next subsection is devoted to an overview of ABR.

ABR

In the ABR service, sources are informed by the network about the rate at which they should be transmitting. For this purpose, the switches monitor their load and compute the available bandwidth and divide it fairly among active flows. The source node sends periodically a Resource Management (RM) cell to the network, which is then returned back by the destination to the source. The RM cells contain the source's Current Cell Rate (CCR) and Minimum Cell Rate (MCR). The RM cells also have several fields that can be used by the switches to provide feedback to the sources. These fields are Explicit Rate (ER) field, Congestion Indication (CI) flag, and No Increase (NI) flag. The ER field indicates the rate that the network can support at the particular instant in time. At the

transmission of a cell, its ER field is set to Peak Cell Rate (PCR) and CI and NI flags are cleared. On the way to the destination, the RM cell is not modified. When the RM cell is on its way back from the destination to the source, each switch sets the ER field to the minimum of the current ER value in the RM cell and the maximum rate the switch can support. In certain cases, it can also set the CI and NI flags. When a source receives the returning RM cell, it computes its Allowed Cell Rate (ACR) using the current ACR, CI, NI flags, and the ER field of the RM cell. A detailed explanation of source and end system behavior is available in [15]. If the traffic demand is higher than the calculated ACR, part of the demand is sent with a rate ACR and the remaining part is delayed at the edge. ABR is a rate-based flow control mechanism for ATM networks but in our TE approach to be described later, we will make use of the ABR feedback mechanism for traffic engineering purposes.

We note that a major disadvantage of the IP over ATM model is that IP and ATM use completely different addressing and routing schemes, which amounts to the need for managing two separate networks. Moreover, ATM uses 53-octet cell sizes and every IP datagram which can be up to 1500 octets must be segmented into 53-octet ATM cells and this requires additional processing. This may be very costly, especially at very high speeds. The IP over ATM approach also has a scalability limitation. To establish full meshed logical connections between N edge nodes, each node must set up logical connections to $(N-1)$ other nodes. Thus, $N*(N-1)$ logical connections have to be established for the full-mesh virtual network. This is known as the "N-square" problem in the literature [16]. Therefore, a new technology called the Multi-Protocol Label Switching (MPLS) is proposed to use the flexibility of IP routing and the efficiency of ATM forwarding. Next, we overview the MPLS technology.

1.1.3 MPLS Networks

Overview

MPLS is an emerging technology that offers new capabilities for IP datagram networks and provides solutions for the problems mentioned in the previous section for the "IP over ATM" overlay model. It is called "multiprotocol" because this technique can be applied to any network layer protocol. It is called "label switching" because the per-hop forwarding behavior of a packet in MPLS network is achieved by swapping the MPLS labels attached to each packet. MPLS combines the flexibility of IP routing and the efficiency of the ATM label swapping. MPLS improvements and developments are:

- MPLS provides traffic-engineering capabilities for packet
- MPLS provides routing strategies on the basis of attributes other than the destination addresses,
- MPLS supports hierarchical label stacks and tunneling.

Operation

When a packet arrives at the edge Label Switched Router (LSR), it is first mapped into a Forwarding Equivalence Class (FEC) according to a predefined policy. A FEC is defined as a group of packets that can be treated equally by the network for the purposes of forwarding. FECs may be based on service requirements for a group of packets or they may simply be based on destination addresses. For example, all packets or all high-priority packets destined to the same destination border router may form a FEC. The aim of classifying packets into FECs is to enable the service provider to traffic engineer the network and route each FEC in a specified manner. This is achieved by mapping arriving

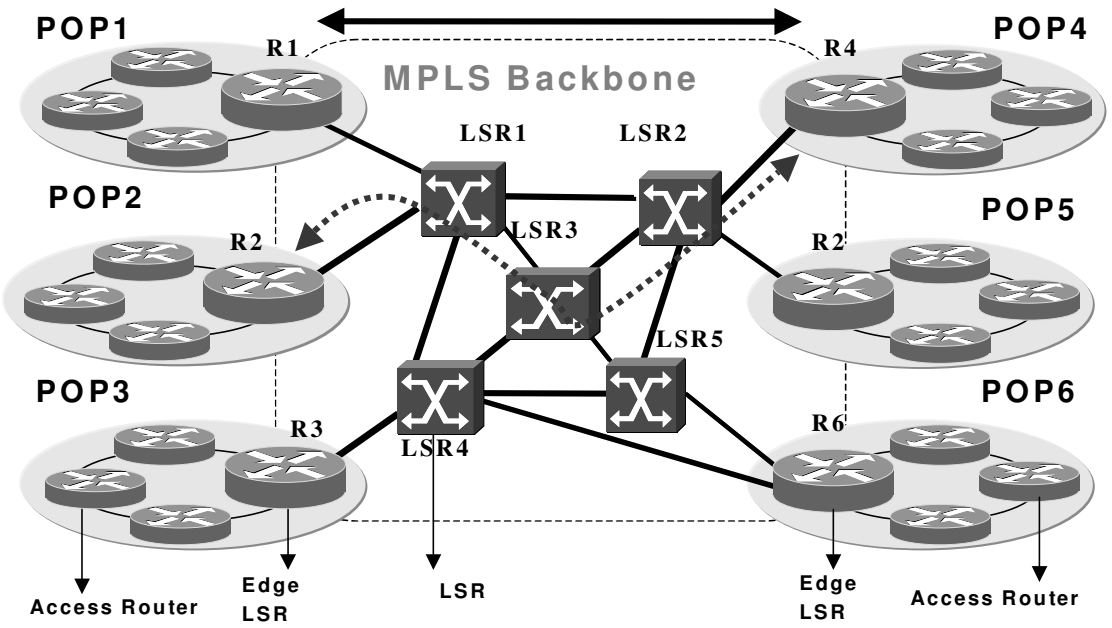

Figure 1.3: An MPLS Network.

packets belonging to a FEC to one of the LSPs associated with the FEC. After a packet is assigned to a FEC, it is tagged with a label, and forwarding to the destination border router is done using only the label and the encapsulated IP address is not used in the MPLS network anymore.

Label Switched Paths (LSP) are a sequence of labels at each and every node along the path from source to destination. LSPs are established either prior to a data transmission or upon detection of a certain data flow. MPLS provides two different ways to set up an LSP:

Hop-by-hop routing: Like the methodology used in IP networks, traffic will follow the shortest path using routing protocols like OSPF.

Explicit routing: This is very similar to source routing in IP networks. In this methodology, the ingress LSR specifies the list of the nodes that the Explicitly Routed LSP (ER-LSP) will traverse throughout its journey in the network.

In Figure 1.3, MPLS is used as the backbone architecture, in which LSRs replace the core routers in the pure IP network and the ATM switches in the ATM network. The edge-LSR operates between the access network and the MPLS network. When an IP packet originating at POP2 and destined to POP4 is first received by R2, R2 sends a label request toward R4. This request goes through LSR1, LSR3 and LSR2. It is a longer path instead of the shortest path since the explicit routing capability of MPLS makes it possible. When the request reaches the egress router R4, it assigns a label to the flow and sends its label mapping back to the upstream router. Each intermediate LSR assigns a label and keeps a forwarding table which specifies an output interface and output label pair for a particular input interface and input label pair. Thus, an LSP is said to be established between R2 and R4.

Once an IP packet is assigned to a particular FEC, its IP address is not analyzed by the subsequent LSRs. When a core LSR receives a labeled packet it first extracts the label from it and uses it as an index into the forwarding table and determines the outgoing interface and the outgoing label. After inserting the outgoing label to the packet as a new label, the core LSR forwards the packet to the outgoing interface and to the outgoing hop that are specified in the forwarding table. This forwarding paradigm brings a number of advantages when compared with the traditional IP routing.

One of the key benefits offered by MPLS is the support for Differentiated Services (Diffserv) [17] which allows the operator to support TE as well as differentiated services. There are two methods to support diffserv with MPLS. The first one is the EXP-inferred-LSP (E-LSP) method which uses the three EXPerimental (EXP) bits in the MPLS header to classify a packet into 8 different service classes. In this method, only one LSP is used between a source and a destination for a given FEC, but packets belonging to this LSP can be placed in different

queues based on the value coded in the EXP bits. In the second method, packets with different QoS requirements use different LSPs. This method is known as Label-inferred-LSP (L-LSP) method, in which packets join different queues based on the label information in the MPLS header. In the latter method, practically there is no limit on the number of different service classes that can be offered by the operator. In our work, we propose to use the E-LSP method to implement the MPLS-diffserv model since only three service classes are needed. Two service classes are used to differentiate among primary and secondary paths and one service class is used for feedback purposes.

1.2 TE Techniques

TE techniques can be classified under two broad categories: connectionless and connection-oriented. These categories are based on how traffic is transferred through communication networks. It is currently debated whether next-generation routing and TE in IP networks will be connectionless or connection-oriented. The connectionless approach evolves current shortest path algorithms or influences routing metrics in the pure IP networks. In the connection-oriented approach, signaling is used to set up a path between a source and a destination by technologies such as ATM or MPLS. All packets belonging to this source and destination are transmitted along this path. For both connectionless and connection-oriented technologies, the multipath TE approach can be applied, in which traffic between a source and a destination is transmitted over multiple paths instead of a single path.

1.2.1 Connectionless TE

Connectionless Single Path TE

Traditional IP networks use shortest path routing using simple metrics such as hop-count or delay. The shortest path routing is the main limitation in achieving TE goals in the connectionless model. Although the simplicity of this approach allows IP routing to scale to very large networks, it does not make the best use of the network resources. The quality of shortest path routing depends highly on the choice of link metrics. The link metrics are often set to unity or set as inversely proportional to the capacities of the links, based on the Cisco [18] recommendation, without taking any knowledge of the demand into consideration. Consequently, the link along the shortest path between the two nodes may become congested, while other links along alternate paths remain underutilized.

Routing provides connectivity between network elements and involves two types of operations: data-plane and control-plane operations. The data-plane operations are those that are performed on every packet, whereas the control plane sets up information to facilitate data-plane operations. While address matching and forwarding are examples of data-plane operations, the local forwarding tables at the intermediate nodes are set up by the control-plane function. In the hop-by-hop model, the data-plane forwarding algorithm is coupled with the control-plane algorithm because both algorithms use the same global identifiers such as IP addresses and link metrics. When the control-plane algorithm is significantly changed, the forwarding algorithm should be changed. Therefore, TE efforts using this model have mostly focused on optimizing the performance via parametric or indirect methods such as optimizing link weights. Protocol extensions for TE in this model have not been deployed because they require major upgrades due to the coupling of data and control planes.

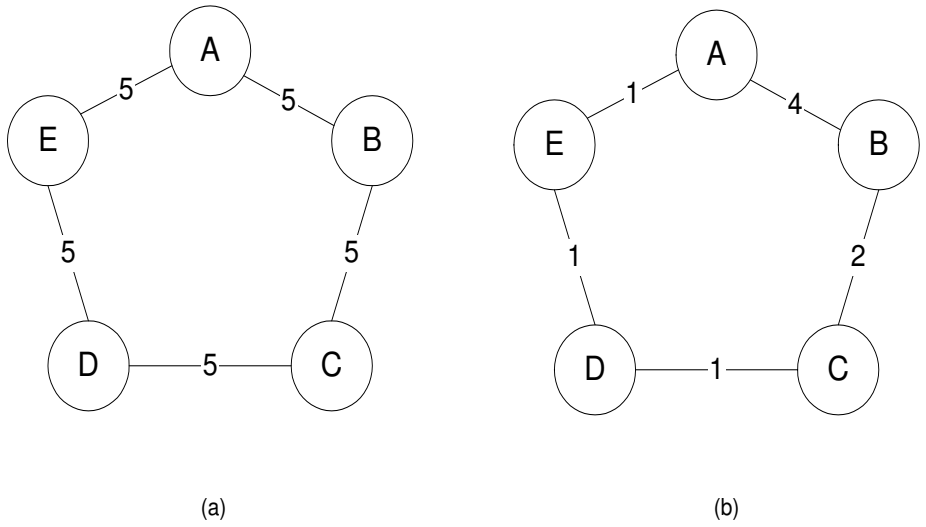

Figure 1.4: (a) Link capacities. (b) Optimal link weights.

Several researches have proposed the computation of optimal link metrics, which requires that the traffic demands are known, a-priori [19], [16], [20]. These approaches are generally based on centralized mechanisms and they have been shown to improve the efficiency of the network. The following simple example illustrates how these approaches work.

In a simple network topology depicted in Figure 1.4(a), each link has a capacity of 5 units and there are demands from A to B, B to C, and A to C. Each demand needs bandwidth of 5 units. Link capacities are assumed to be bidirectional for simplicity. When the traffic demands are transmitted over the optimal routes, the traffic distribution is balanced. The optimal routes can be calculated using linear programming formulation [21] and is as follows in this simple case.

- 5 unit demand from A to B goes over path AB.
- 5 unit demand from B to C goes over path BC.
- 5 unit demand from A to C goes over path AEDC

After finding the optimal routes, these approaches simply calculate and set the appropriate link weights on the links, and the shortest path routing will be

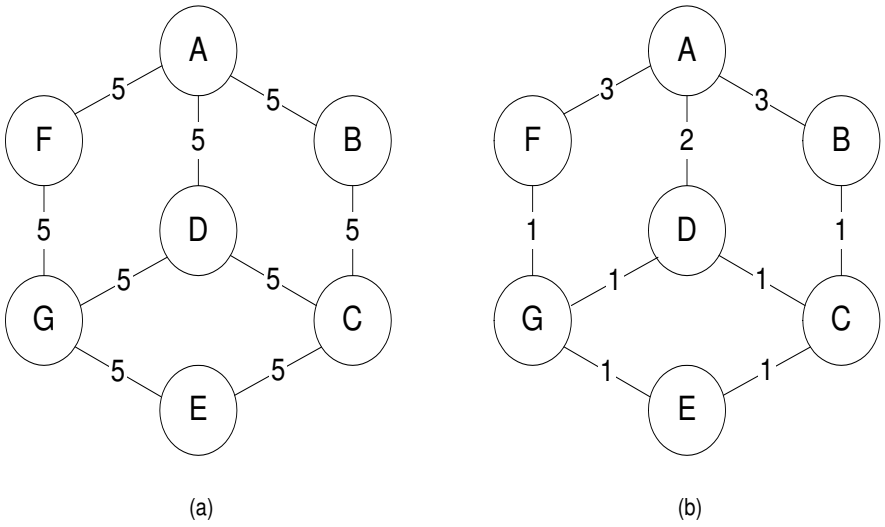

Figure 1.5: (a) Link capacities. (b) Optimal link weights.

responsible for calculating the paths by itself. The link weights under which the shortest paths match the optimal routes exactly are shown in Figure 1.4(b). The shortest path is AB for traffic from A to B, BC for traffic from B to C, and AEDC for traffic from A to C under these link weights. When we set the link metrics inversely proportional to the capacities of the links without taking any knowledge of the demand into account, it is not possible to carry all demands.

Connectionless Multipath TE

There may exist multiple equal cost paths formed by OSPF which is true of any link state protocol. There are techniques utilized to divide traffic evenly among the available equal cost paths. These techniques have been referred to as Equal Cost Multi-Path (ECMP) [19]. The next link on all shortest paths to all possible destinations are stored in a table in the router, and a demand going into the router is sent to its destination by splitting between the links that are on the shortest path to the destination. The effectiveness of this approach is dependent on how many equal cost shortest paths exist between each source and destination pair. The following simple example given in [16] illustrates the benefits of ECMP by optimizing link metrics.

In a simple network topology depicted in Figure 1.5(a), each link has a capacity of 5 units and there are demands from A to B, A to F, B to F and A to E. Each demand needs bandwidth of 4 units. Link capacities are assumed to be bidirectional. The optimal routes are as follows in this simple case.

- 4 unit demand from A to B goes over path AB.
- 4 unit demand from A to F goes over path AF.
- 2 unit demand from B to F goes over path BCDGF and the other 2 unit over path BCEGF.
- 2 unit demand from A to E goes over path ADCE and the other 2 unit over ADGE.

This traffic distribution provides every link with 4 unit load, and thus link utilization is 80% uniformly for the entire network. The link weights under which the shortest paths match the optimal routes exactly are shown in Figure 1.5(b). The shortest path is AB for traffic from A to B and AF for traffic from A to F under these link weights. For traffic from B to F and traffic from A to E, there are two equal-cost shortest paths for each of them. Both demands are distributed between their respective two equal cost shortest paths.

Even traffic splitting among the available equal cost paths does not provide a sufficiently fine level of granularity. When traffic can be divided unevenly among the available paths, it may provide a better resource utilization. However, routers have no knowledge of traffic loading on the distant links and therefore it still is not possible to allocate traffic optimally using the current OSPF protocol. OSPF Optimized Multipath (OSPF-OMP) [22] is a compatible extension to OSPF. In OSPF-OMP, loading information on network links is assumed to be flooded using an enhanced interior gateway protocol and a load adjustment algorithm is proposed for performance improvement.

These approaches have a number of advantages. First, they retain the simplicity of IP routing and require little changes to the basic Internet architecture. Once the link weights are calculated and set, the shortest path routing can calculate the paths in the normal way, and packets are forwarded along the shortest paths. Second, they eliminate the "N-square" problem all together and they are void of messaging overhead in setting up logical connections.

There are a number of drawbacks of the connectionless TE approach. The first drawback is the inadequacy of measurement functions. A traffic matrix is a basic data set needed for TE. It is difficult to estimate the traffic matrix from interface statistics on IP routers. Secondly, although it is shown that there is an optimal route set, finding optimal link weights which yield the optimal route set for a given traffic demand is a NP-Hard problem [19]. The third disadvantage is that changing link weights leads to a transient period when the routers will be advertising new link weights and re-computing the shortest paths. Therefore, the frequent modification of link metrics in case of congestion or substantial changes in traffic demands may lead to undesirable network-wide effects [23] due to the flooding mechanism used in the underlying routing protocol. For example, change in routes may lead to out-of-order arrival of TCP packets and formation of transient loops.

1.2.2 Connection-oriented TE

In order to overcome the limitations of connectionless TE approaches, the connection-oriented TE methods are used.

Connection-oriented Single Path TE

In the connection-oriented approach, service providers establish logical connections between the edge nodes of a backbone, and then overlay these logical connections onto the physical topology. The hop-by-hop routing paradigm using shortest paths is then overwritten in this approach since the logical connections can take any feasible path through the network. The logical connections are typically set up as ATM Permanent Virtual Circuits (PVCs) or ATM Switched Virtual Circuits using QoS signaling and routing protocols like PNNI [14]. The emergence of Multi Protocol Label Switching (MPLS) also provides mechanisms in IP backbones for explicit routing to facilitate TE [2], [11]. In ATM or MPLS backbones, one can use a constraint-based routing scheme so that traffic may optimally be controlled to flow through certain routes [24], [25]. In the example given in the previous section (Figure 1.5), implementing the optimal routes with the overlay approach is quite straightforward. We simply set up six logical connections: AB, AF, BCDGF, BCEGF, ADCE and ADGE, and run IP over them. BCDGF and BCEGF appear as equal-cost paths, so routing protocols such as OSPF will perform load sharing over them.

Typically, in the connection-oriented approach, an initial logical connection layout is obtained for TE using a long-term traffic matrix and constraint-based routing. The bandwidth allocated to each virtual connection in this layout is directly related to the actual long-term traffic between the end points of the virtual connection [26]. However, the actual traffic may occasionally deviate significantly from the long-term value and this deviation generally may not be predicted in advance. In case of a significant traffic increase for the actual traffic that a certain logical connection would carry, additional bandwidth allocation will be needed and the network will be signaled for a change in allocated bandwidth. In case bandwidth is available, this request will be accepted and additional allocation will be made. Otherwise, a new constraint-based route will be sought and if

found the original logical connection will be torn down and the new logical connection will be established. In case of a significant traffic decrease, the network will be signaled for the deallocation. A crucial performance factor is the amount of traffic change required to initiate such a signaling process. If this signaling process is carried out very frequently and even for small traffic changes, there will be a significant amount of message processing. Besides, since there is a possibility of a route change as a result of this process, the network generally may be forced for reconfiguration too often, which then leads to instability. On the other hand, if only very long-term changes in the traffic matrix are taken into account, the network may not adapt itself to significant short-term variations in the traffic matrix, which then leads to degradation in performance.

There are fundamental drawbacks to the IP over ATM overlay model. The most significant one is the need to build and manage two different networks. Secondly, there is a scalability problem. The number of adjacencies in the overlay graph generally increases quadratically with the number of routers. Other drawbacks include the quantization and encapsulating overhead associated with ATM and difficulty in performing segmentation and reassembly (SAR) at very high speeds. MPLS combines the flexibility of IP routing and efficiency of the ATM connection-oriented feature and overcomes the problems presented above.

Connection-oriented Multipath TE

Another approach taken in the literature is that multiple (as opposed to a single) logical connections using disjoint paths are established between the two end points of a network. These paths are generally determined using the long-term traffic matrix. The goal is then load balancing short-term fluctuations among multiple paths so as to increase the performance of the network. Multipath routing has a potential to aggregate bandwidth on various paths, allowing a network to support data transfer higher than what is possible with one single

path. In the example given in section 1.2, it is impossible to carry all demands by using only a single path while the optimum traffic distribution is provided using the multipath approach. Multipath algorithms can be applied in both connectionless and connection-oriented models.

MPLS protocol has a capability to establish one or more explicit paths to a given egress router within the MPLS domain. MPLS Optimized Multipath (MPLS-OMP) [27] is the technique used to balance load across the network topology. The technique uses OSPF or IS-IS with extensions to the link state protocol to flood loading information. The ingress router computes a hash with a sufficiently fine level of granularity based on the IP source and destination. According to the outcome of the hash, the traffic is placed onto one of the explicit paths. MPLS-OMP does not require additions or modifications to the MPLS protocols.

In [23], probe packets are transmitted periodically to the egress node, which returns them back to the ingress node. Based on the information in the returning probe packets, the ingress node computes the one-way statistics like delay and loss for all the paths, and uses a gradient projection algorithm for traffic splitting among multiple disjoint paths. The mixed integer programming (MIP) formulation of the multiple-path finding problem is proposed in [28]. The authors [28] first split the traffic demand over the network links so that the maximum link utilization in the network is minimized while the traffic demand is satisfied. And then, they find paths from the split pattern by using the algorithm for maximum flow path for each traffic demand. In [29], they propose a feedback-based multipath routing algorithm which applies additive-increase and multiplicative-decrease (AIMD) for MPLS networks. Each LSP in the network periodically sends binary congestion-state information to all sources, and each source adapts its rate based on this feedback information. All sources try to send the traffic demands on their primary paths, and the secondary paths are utilized only when

the primary paths are insufficient to meet the desired sending rate. The enhanced routing scheme for load balancing by separating long-lived and short-lived flows is proposed and it is shown that congestion can be greatly reduced [30]. In [31], a dynamic multipath routing algorithm in connection-oriented networks has been proposed, where the shortest path is used under light traffic condition and multiple paths are utilized as the shortest path becomes congested.

1.3 Our Proposed TE

In the previous sections, we overviewed three Internet backbone architectures namely, the pure IP network, the ATM network and the MPLS network and four TE techniques namely, connectionless single path, connectionless multipath, connection-oriented single path and connection-oriented multipath. The pure IP network does not provide explicit routing capabilities, therefore it has limited options for TE. On the other hand, in the IP over ATM model, IP and ATM use completely different addressing and routing schemes, which amounts to a need for managing two separate networks. Moreover, the process of segmentation and reassembly in IP over ATM networks with high speeds may be very costly. Therefore, we propose to use the connection-oriented MPLS technology as the Internet backbone architecture in our proposed TE approach.

A large number of research studies exist for the single path connection-oriented TE method [2], [24], [25]. This line of research generally assumes that the traffic matrix is known a-priori. In this thesis, we propose to use a connection-oriented multipath TE method that does not require the availability of any prior information on the traffic matrix. Our main goal in this thesis is to increase the total amount of carried traffic, or equivalently throughput, by using two paths (as opposed to a single path) for every source-destination pair in the network.

In our proposed TE architecture, two MPLS LSPs are established between every pair of IP routers. We propose that these two LSPs are link disjoint, or equivalently, they do not share a common path since otherwise a single congested link would possibly jeopardize the traffic belonging to a certain source-destination pair. Moreover, motivated from the restoration literature, we propose to use one of these two paths as the primary path and the other one as the secondary path. In our proposed TE scheme, the primary path is one of the min-hop paths and the secondary path is obtained by pruning the primary path from the network graph and then obtaining one of the min-hop paths from the remaining graph. This choice of two paths ensures link disjointness.

In multipath routing studies, all paths between a source and a destination are mostly treated equally at the data plane. Since traffic flow consumes bandwidth and buffer resources at all the links along a path, path length is also an important factor to be taken into consideration. As a general principle, to use shortest paths than paths of longer length (alternative paths) is always preferable, and thus the so-called "knock on" effect [32], [33] is limited. The knock on effect refers to the phenomenon where using alternative paths by some sources force other sources whose minhop paths share links with these alternative paths to also use alternative paths. This cascading effect can result in a drastic reduction of the overall resource utilization of the network. In order to overcome the knock on effect, we propose in our TE architecture that traffic belonging to the primary LSPs using minhop paths will have a preferential treatment over the traffic of the secondary LSPs that use longer paths. Such a mechanism is possible using the MPLS-diffserv QoS model. This mechanism also ensures that traffic between a certain source-destination pair will be routed over the secondary (i.e., longer) path only if there is no more bandwidth available for the primary path.

We also assume that the ingress LSR periodically sends probe packets to the egress LSR, for both primary and secondary LSPs. The probe packets bring the

explicit rate and delay information back to the ingress LSR. We propose that the ER information is calculated similar to the ABR service in ATM networks. We assume that MPLS switches are capable of measuring the ER and delay statistics, which is currently not standardized. Traffic is then divided among these LSPs using the feedback information received from the network. With this new TE architecture, we conjecture that using multiple paths is always better than the case of a single LSP in terms of overall throughput.

While multi-path routing allows multiple paths between a source and destination, how packets will be split over these paths is an important issue. Sending TCP packets from a single flow on multiple paths with different round-trip-times (RTT) would lead to out of order packet arrivals and hence, performance degradation [34]. In order to minimize the out of order packet arrivals, we use the idea of *burst routing*. We assume that the arrival of packets has a bursty nature or equivalently packet arrivals alternate between active and idle periods. In burst routing, traffic is classified into a number of bins in that all packets belonging to the same flow are in the same bin. We also assume that traffic for a given bin is bursty, or equivalently there are gaps between consecutive activity periods. For each bin, bursts are identified and a decision is made on which path to route the burst. The idea behind burst routing is that in the burst identification stage, if two successive bursts are sufficiently distant from each other, then these two bursts can be routed independently over a number of paths without taking into consideration the packet ordering problem.

The rest of the thesis is organized as follows. In Chapter 2, we present our TE architecture along with our MPLS-diffserv model. We describe the simulation framework to verify the effectiveness of this approach and we present our performance results in Chapter 3. The conclusions and future work items are provided in the final chapter.

Chapter 2

Architecture

In this section, a new multipath TE architecture for connection-oriented networks will be introduced. The architecture proposed here is for MPLS networks but can be used in ATM networks by some modifications. Our proposed TE architecture is comprised of four components:

- MPLS feedback mechanism,
- our MPLS queuing architecture,
- establishment of multiple LSPs using link disjoint paths,
- traffic splitting algorithms implemented at the edge nodes.

2.1 MPLS Feedback Mechanism

The feedback information received from the network plays a crucial role in our TE approach. MPLS technology does not currently have a standard-based feedback mechanism, but we propose that a feedback mechanism very similar to the ABR feedback mechanism in ATM networks, is to be used in MPLS networks as well.

The ingress node of each LSP periodically sends Probe Packets (PP) to the network, which is then returned back by the egress node to the ingress node. The PPs have several fields that can be used by the switches to provide feedback to the sources. These fields are Explicit Rate (ER), Delay Estimate (DE), Congestion Indication (CI) flag, and No Increase (NI) flag. The ER field indicates the rate that the network can support at the particular instant in time. The DE field indicates the current end-to-end delay of a packet. The length of a PP is 50 bytes, which we believe has sufficient length to carry the above information.

Currently used MPLS switches do not have standards-based capabilities to calculate the ER and DE values, but we assume that they will have such capabilities in this study. The MPLS switch runs an independent explicit rate algorithm to calculate the ER for all LSPs and for each of its interfaces. In our experimental studies, we use the ERICA ABR explicit rate algorithm [42], which is known to allocate the max-min fair share in a wide variety of scenarios. The ERICA algorithm is briefly explained in Appendix A. It is originally designed for ATM switches and is based on the ATM cell structure. We modified this algorithm for variable size packets in such a way that we count bytes instead of 53 byte ATM cells in our counters which are used to calculate the ER information.

In this architecture, we assume three different services, for probe packets, higher priority packets, and lower priority packets, respectively. As a second modification to the ERICA algorithm, the residual capacity remaining from the higher priority service is used as the available capacity of the lower priority service. Each switch calculates its own mean delay estimate for a particular service by dividing the number of total bytes in its queue to the available capacity of this service. The delay estimate of an LSP can be obtained by accumulating delay estimates of all switches used by this LSP. In [23], the probe packet is time-stamped at the ingress node at time T_1 and recorded at the egress node at time T_2. Then, total packet delay is T_2-T_1. In this method, probe packets should

join the same queue as its data packets and encounter the same queueing delay. This approach has a drawback of increasing the feedback delay. In our approach, probe packets have the highest priority over all packets so that prompt feedback is provided.

PPs are emitted to the network after every NPPth (10 is default value) data packet for each LSP service. To be able to provide feedback in the case of no data transmission or when the data transmission rate is low, extra PPs are emitted to the network in every EPP seconds (20 ms is default value) if no PP is sent within EPP seconds . The mean size of IP packets in the Internet are about 500 bytes [36], therefore there is about 2% overhead for each LSP due to PPs (NPP is 10). On the way to the destination, the PP is not modified. When the PP is on its way back from the destination to the source, three main operations are performed. Firstly, each switch sets the ER field to the minimum of the current ER value in the PP and the maximum rate the switch can support. Thus, every source sends at a rate no more than the ER calculated at its bottleneck point, otherwise the excessive traffic will be dropped at the bottleneck point. Secondly, the CI is set if the buffer occupancy of the switch is more than T_{buf}. When the sending source receives the ER, CI and NI information, traffic will be sent according to the standard based ABR source behaviour [15]. Finally, each switch accumulates its mean delay estimate to the DE field of the PP. There is a delay in per-destination buffers at the edge node, and this delay is calculated by the same method used in the intermediate switch. This delay estimate is added to the DE of the probe packet to obtain the final delay estimate for this particular service class. The final delay estimate is the end-to-end delay that a packet joining a particular service class will encounter.

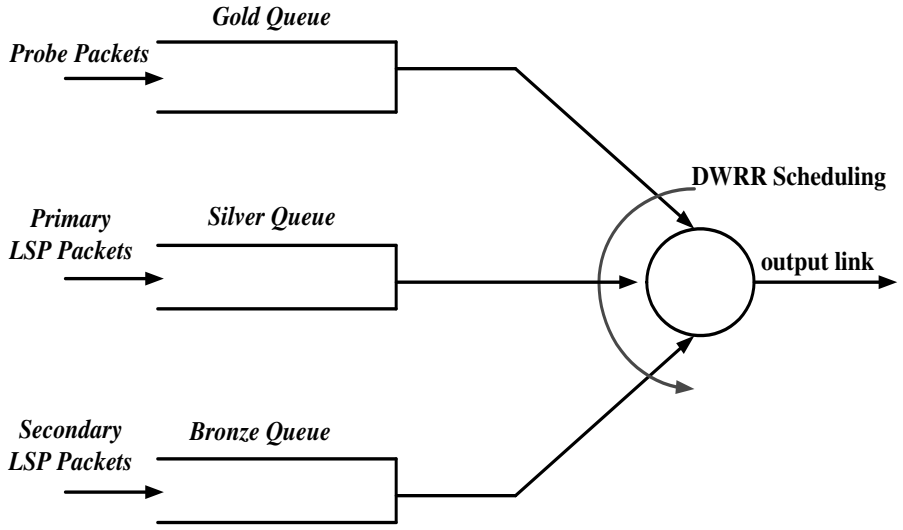

Figure 2.1: Queuing architecture for the MPLS core switch.

2.2 MPLS Queuing Architecture

One of the most important features of MPLS is the capability of supporting differentiated services. MPLS-diffserv implementation allows TE while supporting differentiated services. In our proposed architecture, we assume three differentiated services, namely gold, silver, and bronze services [17]. To achieve differentiation, we propose to use the E-LSP method in which 3 EXP bits in the MPLS header can be used to determine which of the services a packet will receive.

The envisioned MPLS queuing architecture is given in Figure 2.1. As depicted in this figure, we propose that all probe packets will receive the gold service by joining the so-called gold queue. On the other hand, packets routed over the primary LSPs will receive a silver service and those routed over the secondary LSPs will receive the bronze service, by joining their respective queues.

The queuing discipline we use is Deficit Weighted Round-Robin (DWRR) [35] scheduling in which the gold queue has a higher priority than the silver and bronze queues, and the silver queue has a higher priority than the bronze queue. In the DWRR queuing algorithm, a weight is assigned to each queue to determine the percentage of the bandwidth a queue receives. For example, if the weight of

the silver queue is twice of the weight of the bronze queue, then the silver queue will receive twice as much bandwidth as the bronze queue when both queues are active. In our proposed architecture, the weights of each queue are chosen in such a way that the gold queue has "almost" strict priority over all other queues, and the silver queue has "almost" strict priority over the bronze queue. To provide prompt feedback information, we give the highest priority to the PPs. The motivation behind the isolation between two services silver and bronze by using strict priority scheduling in the data plane is to eliminate the so-called knock-on effect observed in load balancing algorithms [32]. The knock on effect refers to the phenomenon where using alternative paths by some sources force other sources whose minhop paths share links with these alternative paths to also use alternative paths. For a given source-destination pair, the primary LSP used by the silver service uses fewer hops than the secondary LSP used by the bronze service because of the way we set up these LSPs. Therefore, DWRR scheduling with appropriate weights is proposed for making sure that the performance of the silver service is not impacted by the load on the bronze queue. DWRR scheduling is preferred over strict priority for avoiding possible starvation of the bronze service.

The MPLS switch, for each of its interfaces, and for each differentiated service runs the ERICA algorithm to calculate the ER for each LSP that is using that differentiated service. The queue capacities are K_LSP for the data LSP queues. For buffer management, we use "drop-from-tail" but since our ER algorithms keep the queue sizes small, the effect of the buffer management policy has been observed to be minimal.

2.3 Establishment of LSPs

There are two main aspects of a multipath routing algorithm: establishment of multiple paths and traffic splitting among these multiple paths. In this study, we mainly focus on the aspect of traffic splitting among multiple LSPs between an ingress node and an egress node. The simple shortest path routing is extended to find multiple disjoint paths.

In our proposed TE architecture, we establish two disjoint LSPs between every source-destination pair of IP routers. We propose that these two LSPs are link disjoint, or equivalently, they do not share a common link since otherwise a single congested link would possibly jeopardize the traffic belonging to a certain source-destination pair. For a particular source-destination pair, the first LSP is the primary one and uses the shortest path (minimum hop path) found using the Dijkstra's algorithm. When there is a tie in the algorithm, we break the tie in a random manner. The second LSP is the secondary LSP and the path for this LSP is found by pruning the links used by the primary LSP and choosing one of the minimum hop paths in the remaining network graph. This choice of two paths ensures link disjointness. If the connectivity is lost after pruning links from the graph, the secondary LSP is not established. Other algorithms can also be used to find two link-disjoint paths but a comparative analysis of these methods and their impact on throughput is left for future research.

2.4 Traffic Splitting

The other main aspect of a multipath routing algorithm is to decide how the traffic is split among multiple paths. We present our traffic splitting approach in Figure 2.2. We assume M destinations and L ports for a given source, and P bins for each destination from this source. We use bins to avoid packet mis-ordering

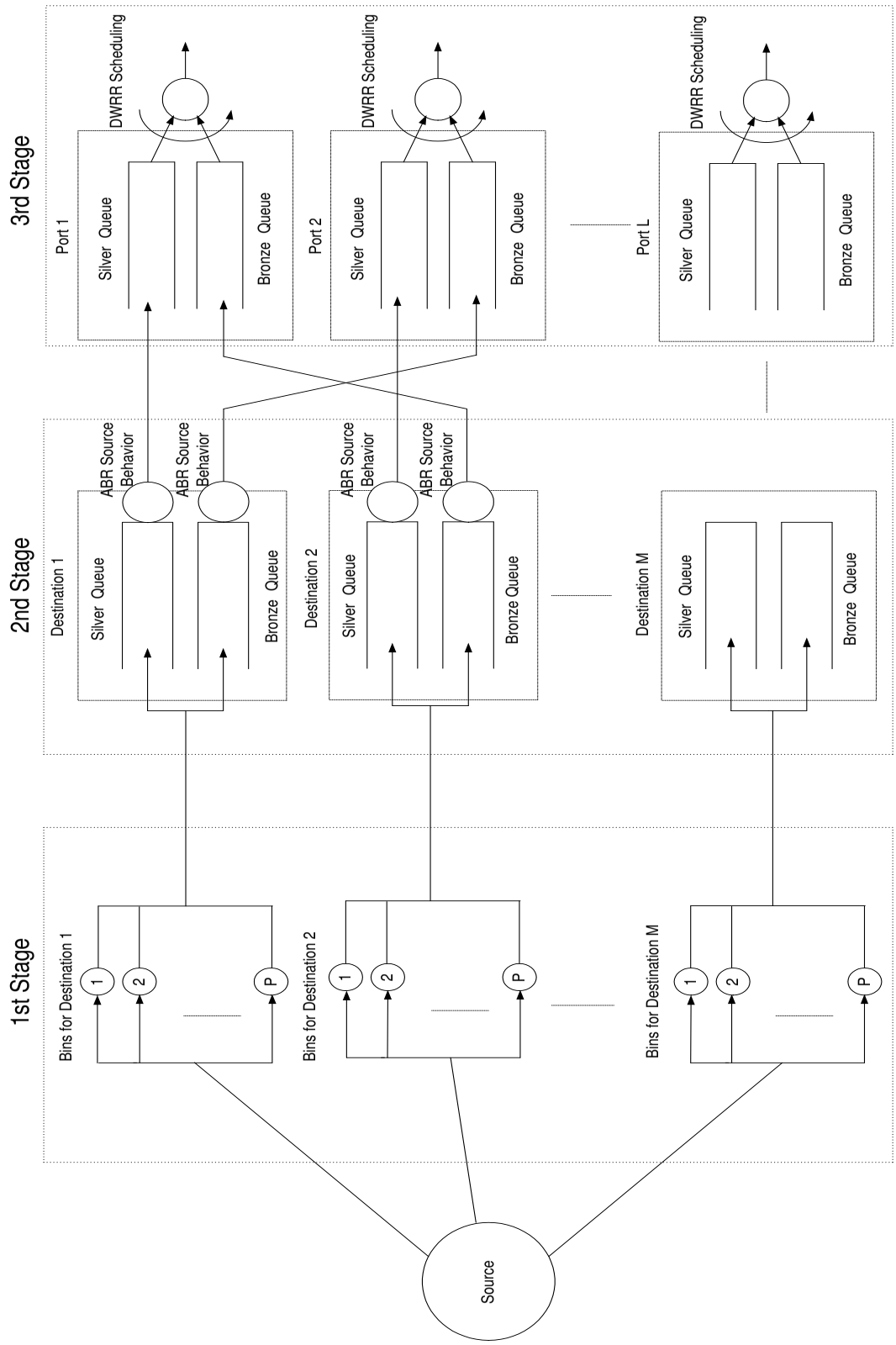

Figure 2.2: MPLS edge switch architecture and traffic splitting mechanisms. We assume that the primary and secondary LSPs for destination 1 use Port 1 and Port 2, respectively and the primary and secondary LSPs for destination 2 use Port 2 and Port 1, respectively. Ports used for destination M are not shown for the sake of simplicity.

within a TCP flow, which can falsely trigger congestion control mechanisms and cause unnecessary throughput degradation [43]. There are three stages in our proposed traffic splitting approach. The following operations will be performed for every new packet generated by a source node in each stage.

In the first stage, we assume that a hash function based on source-destination IP addresses is applied to the packet and the packet is placed to a certain bin (out of the P bins) according to the outcome of the hash function. In this thesis, we do not focus on how P is chosen and how traffic is distributed among the P bins, but there are studies on this topic [23, 43]. We note that traffic carried between source and destination LSRs is an aggregation of traffic generated by many individual sources. A typical traffic source on the Internet does not generate data at constant rate, but is very bursty in nature. We assume that the superposition of the appropriate number of traffic sources is still bursty in nature. Therefore, an appropriate hashing scheme leads to a finer granularity, and the offered traffic from each bin to LSP queues is still bursty, it has active (T_{on}) and idle (T_{off}) periods in transmission. A bin is not a queue, it is an abstract structure used to implement the hashing scheme, and thus to provide these active and idle periods in transmission. The number of bins belonging to a source-destination pair determines the level of granularity, thus the active and idle periods. These periods will be used to decide which service queue a packet should join.

In the second stage, we implement per-destination queuing or to be more precise, we build a silver queue and a bronze queue for every destination at the source. After the packet is mapped into a bin, we decide which service queue it should join. If the packet is to join the silver queue, this is to say that the packet is forwarded via the primary LSP. On the other hand, if the packet joins the bronze queue, then this means that the secondary LSP is used for forwarding this packet. We use the algorithm given in Table 2.1 for these decisions. Each bin

keeps two states: $prev\_pack\_service$ and $prev\_pack\_arriv\_time$. The notation $prev\_pack\_service$ denotes the service queue which the previous packet belonging to this bin has joined and $prev\_pack\_arriv\_time$ denotes the arrival time of the previous packet belonging to this bin. The variables $cur\_pack\_service$ and $cur\_pack\_arriv\_time$ denote the service queue a newly arriving packet will join and the arrival time of the current packet, respectively. T_{int} represents the inter-arrival time of packets belonging to the same bin and found by subtracting $prev\_pack\_arriv\_time$ from $cur\_pack\_arriv\_time$. The variables D_{LSP1} and D_{LSP2} are the delay estimates for the primary and secondary LSPs, respectively. D_{max} denotes the maximum allowed delay for a packet through the MPLS backbone network.

In step 1, the threshold T_{thr} is obtained based on $prev\_pack\_service$. Assume that the previous packet has joined the silver service. Then, the threshold time is calculated by subtracting the delay estimate of the bronze service from the delay estimate of the silver service, and T_{fix} is added to provide a safety margin in case of estimation errors. T_{thr} is actually used to preserve packet ordering when splitting traffic into silver and bronze queues. To be able to transmit this packet over the secondary LSP (bronze queue), T_{thr} should be smaller than the difference between arrival times of the two consecutive packets (previous and current packets). The out of order packet situation arises when a packet created earlier reaches its destination later. For instance, assume that there are two packets belonging to the same bin, namely packet A and packet B. Packet A is created earlier than packet B and joins the silver queue while packet B joins the bronze queue. If packet B reaches earlier than packet A to the destination, it is an out of order situation.

In step 2, the inter-arrival time T_{int} is compared to T_{thr}, to make sure that there is no restriction on which queue a new arriving packet will join. In fact, the comparison of T_{int} and T_{thr} is made to check whether it is possible that the packet

Step1

if $prev\_pack\_service$ *is silver queue*
$T_{thr} = T_{fix} + D_{LSP1} - D_{LSP2}$ % threshold time to assign a packet to bronze queue
else
$T_{thr} = T_{fix} + D_{LSP2} - D_{LSP1}$ % threshold time to assign a packet to silver queue

Step2

if $T_{int} < T_{thr}$
Packet is assigned to $prev\_pack\_service$

else % packet may be assigned to any queue (new burst)
if $D_{LSP1} > D_{max} - D_{tol}$ % silver queue has higher priority
Packet is assigned to bronze queue
else
Packet is assigned to silver queue

Step3

if Packet is assigned to silver queue
if $D_{LSP1} < D_{max}$
Packet joins silver queue % if packet will reach the destination within required time
else
Packet dropped % otherwise it is dropped

else
if $D_{LSP2} < D_{max}$
Packet joins bronze queue % if packet will reach the destination within required time
else
Packet dropped % otherwise it is dropped

Table 2.1: Traffic splitting algorithm in edge node

sent later can reach the destination earlier if they join different service queues. Our aim here is to minimize the probability of mis-ordering by sending the current packet over the different service if the comparison ensures that there will be no out of arrival situation. The case in which T_{int} is smaller than T_{thr} indicates that the arriving packet should be assigned to $prev\_pack\_service$, otherwise there will be a potential mis-ordering. To explain this case more clearly, assume that nth packet has joined the silver queue and A_n represents the arrival time of this packet. The nth packet will reach the destination at the estimated time $A_n + D_{LSP1}$. However, the n+1th packet will reach the destination at the estimated time $A_{n+1} + D_{LSP2}$ if it is sent over the secondary LSP. If $A_{n+1} + D_{LSP2} < A_n + D_{LSP1}$, this is the out of arrival situation since the n+1th packet sent later reaches the destination earlier. Therefore,the n+1th packet should be assigned to the silver queue to prevent this out of arrival situation. $A_{n+1} + D_{LSP2} < A_n + D_{LSP1}$ is equivalent to $A_{n+1} - A_n < D_{LSP1} - D_{LSP2}$ and $T_{int} < T_{thr}$ ($T_{fix} = 0$). When we say a packet is assigned to a certain queue, we mean that this packet is a candidate for joining that queue. Actually, in our proposed system, an assigned packet will either join the associated queue or will simply be rejected based on the policy dictated by step 3 of the algorithm. If T_{int} is not smaller than T_{thr}, it can be assigned to one of the two queues according to delay estimates received from the network and this case corresponds to a new burst which can be routed over the primary or secondary LSP (burst routing). For a new burst, if D_{LSP1} is larger than $D_{max} - D_{tol}$, the packet should be assigned to the bronze queue, otherwise it will not reach the destination within a required time (i.e., D_{max}) over the primary LSP.

The aim of using the delay tolerance D_{tol} is to prevent unnecessary oscillations of D_{LSP1} around D_{max}. If D_{tol} is not used, the first packet belonging to a new burst will be assigned to the silver queue even D_{LSP1} is slightly less than D_{max}. According to our traffic splitting algorithm, all packets belonging to this burst should be assigned to the same queue in which the first packet is assigned. D_{LSP1}

becomes higher than D_{max} after a certain time since more packets are sent over the primary LSP. However, packets are rejected based on the policy dictated by step 3 of the algorithm (if D_{LSP1} is higher than D_{max}) while it may be possible to send through the secondary LSP. Since packets are rejected in the above case, D_{LSP1} will be again smaller than D_{max} after a certain time and new bursts will be assigned to the silver queue. This situation results in oscillation of D_{LSP1} around D_{max} and this causes throughput degradation of the secondary LSP. Therefore, if D_{tol} is not used, packets which are assigned to the silver queue in step 2 of the algorithm may be rejected based on the policy dictated by step 3 of the algorithm while they can be sent over the secondary LSP.

In the final step, we decide whether an arriving packet will join the assigned queue or it will be dropped, by using the delay statistics. D_{max} is used to restrict delay to a certain upper bound, since packets are treated as lost if they do not reach a destination within a certain time for TCP flow. In the case a packet is assigned to the silver queue, this packet joins the silver queue if D_{LSP1} is less than D_{max}, otherwise it is dropped at the edge. The packet is dropped by considering that it will not reach the destination within the maximum allowed time (D_{max}). The same comparison is made for each packet which is assigned to the bronze queue. After determining the appropriate service and joining it, each bin updates its states. The $prev\_pack\_service$ is updated to the $cur\_pack\_service$ and the $prev\_pack\_arriv\_time$ is updated to the $cur\_pack\_arriv\_time$.

Both queues are drained using the standard ABR source behavior using the ER information riding on the probe packets [15]. In the third stage, we employ per-service queuing for every port as in the case of the MPLS core switch. For each source-destination pair, two link-disjoint LSPs are established prior to data transmission, and we decide which queue is connected to which port according to these LSPs. The ABR ER algorithm runs on the third stage queues and gold queues in the third stage are not shown in Figure 2.2 for the sake of simplicity.

2.5 Operation of Our Approach on an Example Topology

In the above sections, our proposed TE architecture with the traffic splitting algorithm was explained in detail. In this section, we will describe how our algorithm works on a simple topology without getting into details. As depicted in Figure 2.3, two link-disjoint LSPs are established prior to data transmission, namely LSP1 and LSP2. LSP1 which passes through the router R1 denotes the primary LSP and is used to transmit packets belonging to the silver service. LSP2 which passes through routers R2 and R3 denotes the secondary LSP and is used for the bronze service. All Data Packets (DP) originating at S and destined to D are mapped into silver and bronze queues according to our traffic splitting algorithms. In this figure, bins are not shown for the sake of simplicity. Packets belonging to the silver queue are transmitted over the primary LSP and join the silver queue at port 1. Since LSP1 is the primary LSP, gold and silver queues are used in the router R1. All data packets departing from the silver queue at port 1 join the silver queue in R1, while PPs belonging to this service join the gold queue. On the other hand, packets belonging to the bronze queue are sent over the secondary LSP and join the bronze queue at port 2. All data packets departing from the bronze queue at port 2 join the bronze queues in the routers R2 and R3, while PPs belonging to this service join the gold queues in R2 and R3.

PPs are sent over both the primary and secondary LSP to gather feedback information and join their corresponding gold queues. In the meantime, R1, R2 and R3 compute the ER and DE information. On the way to the destination, the PP is not modified. When the PP is on its way back from the destination to the source for the primary LSP, R1 sets the ER field to the maximum rate it can support and adds its own DE value to the value written already in DE field. For

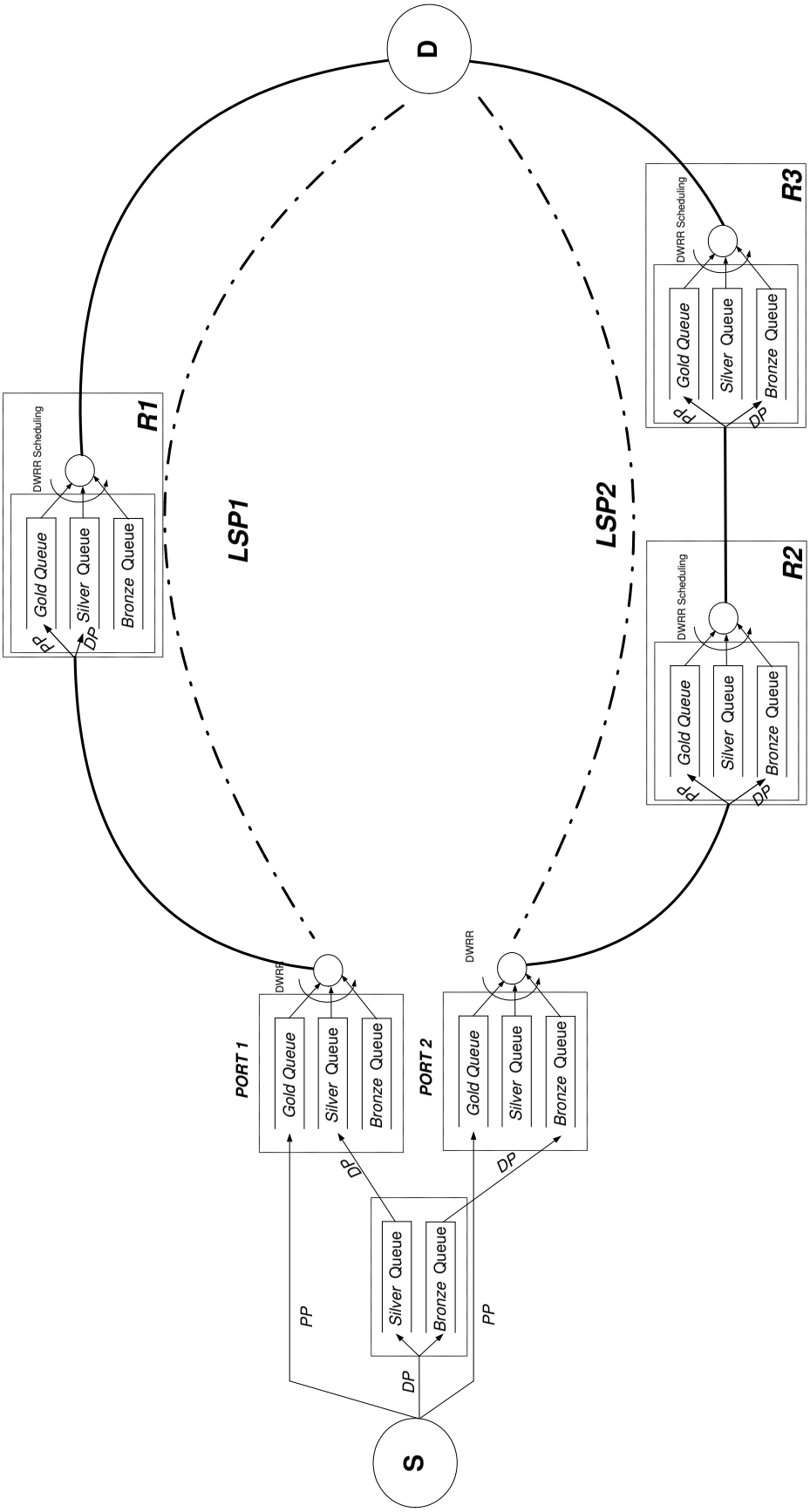

Figure 2.3: Traffic splitting on an example topology.

the secondary LSP, when the PP comes back to the source, the ER field contains the minimum of ERs calculated by R2 and R3 and the DE field contains the sum of DEs computed by R2 and R3. The delay estimate calculated by the source node for a particular service is added to the corresponding DE brought by PPs belonging to this service to find the final delay estimate value which will be used in our traffic splitting algorithm (D_{LSP1} and D_{LSP2}).

Chapter 3

Simulation Results

In this chapter, we will present our simulation results to show the performance of our proposed TE architecture. First, we will describe our event-driven packet-based MPLS simulator, and then present simulation results for a hypothetical US topology and a seven node ring topology. Finally, we will present simulation results which show how the performance of our proposed traffic engineering method is affected when we change the parameters used in our traffic splitting algorithm.

3.1 Simulator

We designed and implemented an event-driven packet-based MPLS simulator using the Java programming language. The simulator allows us to specify the network topology and the traffic demand matrix. The entry *T[i,j]* of the traffic demand matrix gives the long term average traffic rate between the nodes i and j. In our simulation studies, we used the Markov Modulated Poisson Process (MMPP) for generating the actual packet traffic but we note that other distributions for more realistic patterns may also be used. In MMPP, the long term

average traffic rate (T Mbps) is divided by the number of bins (P) belonging to this traffic to find the average traffic rate (R=T/P Mbps) which will be generated by a bin. Then, each bin generates its own bursty traffic with parameters R, T_{onmean}, and $T_{offmean}$. T_{onmean} and $T_{offmean}$ are the mean values of the exponential random variables which are used to generate the active and idle periods of a bin, respectively. When we aggregate the traffic generated by P bins belonging to a source-destination pair, we find the long term average traffic rate (T) which is specified by user before the simulation starts. The simulator reports Current Traffic Rate (CTR) for each LSP along with the Success Rate (SR) and the Overall Success (OS). CTR is the traffic flow of a certain LSP in Mbps to the network at the source. The number of total bytes of a certain LSP, which is sent to the network, is counted throughout an averaging interval and this number is divided by the averaging interval to find the CTR of this LSP at a particular time. SR of a given LSP is measured as the percentage of the number of successfully transmitted bytes of that LSP to the number of total incoming bytes to be carried over that LSP. OS is the measure of the percentage of the total number of successfully transmitted bytes to the number of all incoming bytes to the network during the simulation.

As default simulation parameters, we use 100 bins for each unidirectional traffic between a source-destination pair, 2×10^6 bytes for queue sizes (K_LSP), 10 ms for D_{tol}, and 30 ms for T_{fix}. For every 10 packets belonging to an LSP, a Probe Packet (PP) is emitted to the network to gather the explicit rate and delay estimate information. The extra PPs are sent for every 20 ms in case a PP is not sent within a 20 ms time period.

In our simulation studies, we compare and contrast the methods of single path and multipath (two paths) in terms of their overall throughput and mis-ordering rate. The performance measure used for each method is the Loss Rate (LR), and the Mis-ordering Rate (MOR) for the methods using multiple paths. LR is

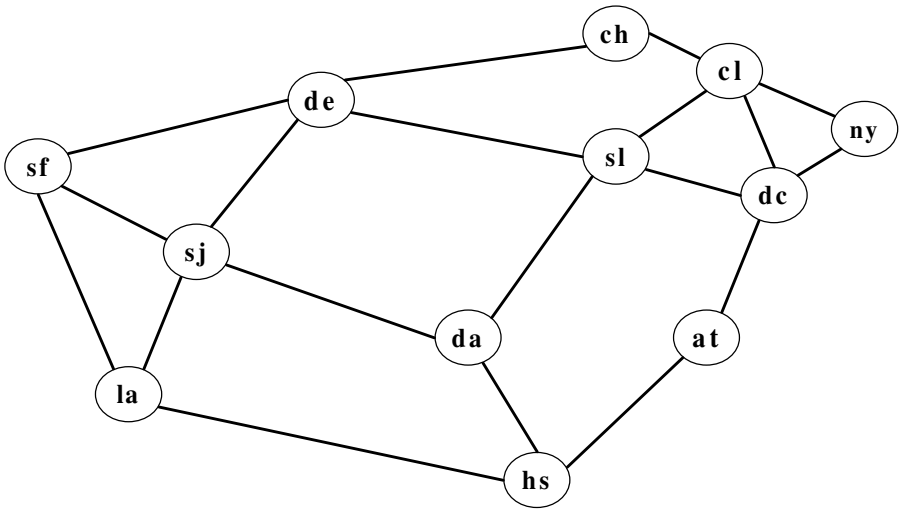

Figure 3.1: The U.S. topology used in our simulation experiments.

calculated by subtracting OS from 100 using each method. MOR for a source-destination pair is defined as the percentage of out of order packets to all packets belonging to this pair. MOR for overall network is calculated as the ratio of all out of order packets to all incoming packets to the network during simulation. When transmitting a packet from a bin, the sequence number is written on each packet. If the sequence number of a new arrived packet to the destination is smaller than the sequence number of the most recently arrived packet, then the new arrived packet is counted as an "out of order packet".

3.2 Mesh Network Topology

As an example mesh topology, we use the US topology depicted in Figure 3.1. This topology and the traffic demand matrix *T[i,j]* are used from the data given in [44]. Each link in this network is bidirectional and has 155 Mbps capacity in both directions except the links between de - ch and ch - cl which have 310 Mbps capacity in both directions.

Traffic Matrix	Method	LR	MOR
T	Single Path	18.44	-
	Multipath	5.29	0.0004
$T \times 0.75$	Single Path	12.51	-
	Multipath	1.37	0

Table 3.1: LRs and MORs for a hypothetical U.S. topology

Table 3.1 presents LR and MOR for two different traffic loads. T in the first column denotes the traffic demand matrix given in [44]. We performed our simulations under two different traffic loads that are T and $T \times 0.75$, the latter one corresponds to the scaled down version of T. In our MMPP traffic generation model, we specify long term average traffic rate, T_{onmean} and $T_{offmean}$ values. T_{onmean} and $T_{offmean}$ are used to model the burst arrivals of the traffic generated by each bin. In these simulations, T_{onmean} and $T_{offmean}$ are set to 200 ms and D_{max} is chosen as 150 ms. We use default parameters 10 ms and 30 ms for D_{tol} and T_{fix}, respectively.

Table 3.1 demonstrates that for traffic matrix T, LR is 18.44 when the traffic is sent only using a single path. The traffic demand matrix is not uniform so that some links are over-utilized while others are not. This results in an inevitably high loss rate. We show that LR is reduced to 5.29 when two paths (multipath) are used for sending traffic. Although there is residual capacity to carry more traffic, a single path is not able to use this residual capacity. The contribution that comes with multipath is the use of this residual capacity to carry more traffic without affecting the shortest-path flows. MOR is 0.0004 in the multipath case. In fact, a bin may contain multiple TCP sessions, therefore this MOR is very pessimistic in the sense that packet ordering should be considered within a single TCP session. With the use of the multiple path method, we observe significant reductions in the loss rate compared to the case of a single path without breaking packet ordering within a flow.

Multiplying T with a scaling factor less than one has an impact of reducing the load on the overall network. LR for the single path case becomes 12.51, whereas for the multipath case it is 1.37. The improvement in the loss rate using multipath is even more significant in this lightly loaded scenario. LR is decreased about tenfold using multipath. For the lightly-loaded scenario, MOR decreases to zero because the residual capacity is more than enough for this traffic demand. Multipath method successfully carries this traffic as observed by the low LR. We conclude from our simulation studies that the more the residual capacity exists, the better our proposed traffic engineering method performs.

A snapshot of the transient behavior of our traffic engineering method is shown in Figure 3.2 by which we present the transient behavior of CTR for the two LSPs from *ny* to *sf* for traffic load T. Total aggregate traffic demand from *ny* to *sf* is 49.579 Mb/s. The Primary LSP (P-LSP) carries 11.628 Mb/s of this traffic and the Secondary LSP (S-LSP) carries 17.213 Mb/s on the average. The primary path from *ny* to *sf* uses the path *ny-dc-sl-de-sf* which is given in Appendix A.3. The total traffic demand of the source-destination pairs whose primary paths use the link between *ny* and *dc* is 200.139 Mb/s. This demand is obtained using the tables given in Appendix A.2 and A.3. However, the capacity of this link is 155 Mb/s and a certain amount of this capacity is used by probe packets. This link is a bottleneck link and not able to carry all demands of the source-destination pairs whose primary paths use it. Therefore, 17.213 Mb/s of the total aggregate traffic from *ny* to *sf* is carried by the S-LSP. As shown in Figure 3.2, the CTR of S-LSP converges slower than that of P-LSP because the S-LSP is used when the estimated delay of a packet over P-LSP is more than 140 ms. At about 200 ms, the source begins to send traffic over the S-LSP,and the CTR of the S-LSP converges to its steady-state value at around 600 ms.

The transient behavior for a scaled traffic matrix is depicted in Figure 3.3. In this figure transient behavior of CTR for the two LSPs from *ny* to *sf* for traffic

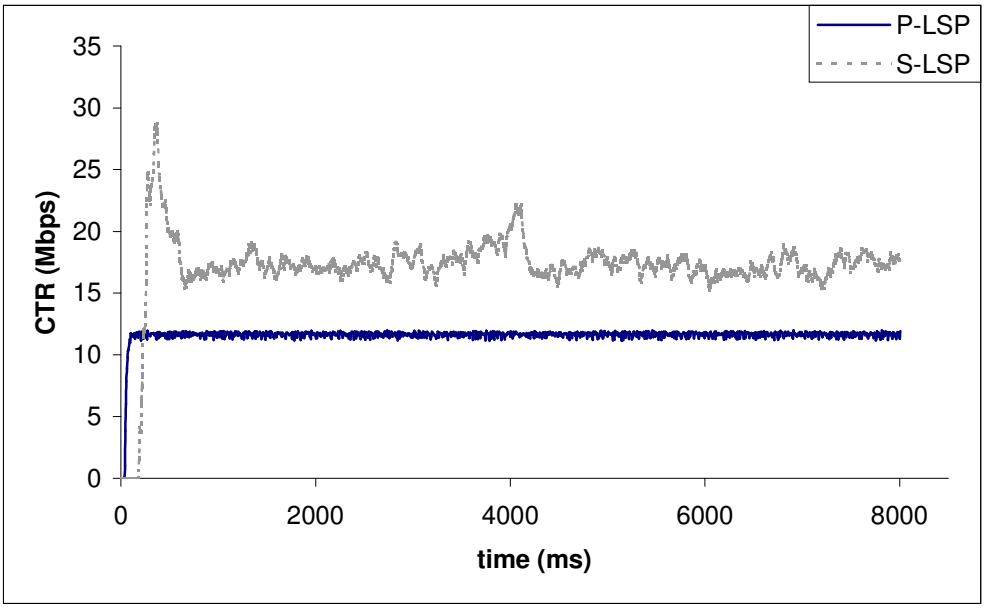

Figure 3.2: Current Traffic Rate graph of traffic flow from *ny* to *sf* for the case traffic load is T.

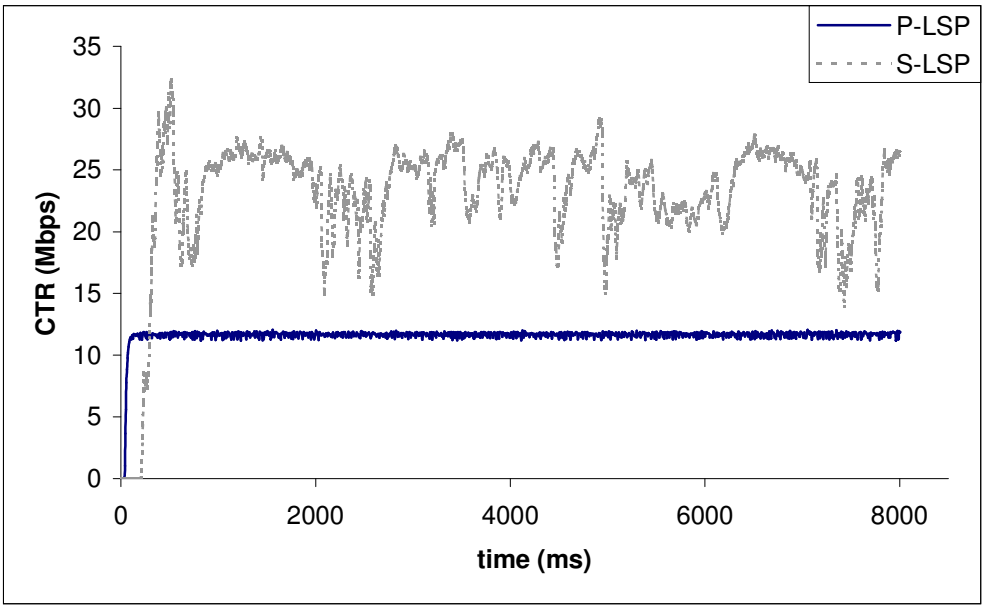

Figure 3.3: Current Traffic Rate graph of traffic flow from *ny* to *sf* for the case traffic load is $T \times 0.75$.

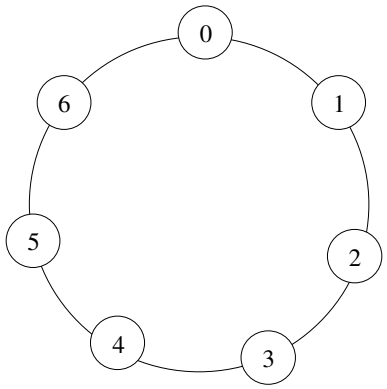

Figure 3.4: 7 node ring topology used in our simulation experiments.

load $T \times 0.75$. Total aggregate traffic demand from *ny* to *sf* is 36.88 Mb/s. The Primary LSP (P-LSP) carries 11.57 Mb/s of this traffic and the Secondary LSP (S-LSP) carries 23.00 Mb/s on the average. When we look at the CTR of the S-LSP in Figure 3.3 we observe more oscillatory behavior than the CTR of the S-LSP in Figure 3.2. The reason is that total aggregate traffic from *ny* to *sf* is almost fully carried, thus the CTR of the S-LSP for scaled down traffic demand follows our generated traffic rate. The nonzero but low LR is mainly due to the transient period of the CTR of the S-LSP.

3.3 Eliminating the Knock-On Effect

In this example, we used seven nodes interconnected to each other using a ring topology depicted in Figure 3.4. Each link is bidirectional and has a capacity of 155 Mbps in both directions. All source-destination pairs have 15 Mbps traffic flow if their shortest paths are in clockwise direction, e.g., source node 0 and destination node 1, otherwise 35 Mbps traffic flow, e.g., source node 0 and destination node 5.

The simulation parameters are chosen as D_{max}=150 ms, T_{fix}=30 ms and $T_{onmean} = T_{offmean}$=200 ms. This topology and traffic matrix are chosen to show the knock on effect if we do not differentiate among the primary and secondary

Method	LR	MOR
Single Path	19.36	-
Equal Priority Multipath	40.82	3.82
Strict Priority Multipath	12.00	0.20

Table 3.2: LRs and MORs for 7 node ring topology

LSPs at the data plane. For source node 0 and destination node 1, the primary LSP uses only one link between node 0 and node 1. On the other hand, the secondary LSP uses six links. If we do not differentiate these LSPs in data plane, ERICA algorithm treats these LSPs equally and gives the same ER value which may be very costly for the overall throughput. This effect is known as the "knock-on" effect in literature. In this case, using multiple paths causes lower throughput value compared to a single path case.

From Table 3.2, LR is 19.36 for "Single Path" method which only uses the shortest path for sending traffic. When we use "Equal Priority Multipath" method which corresponds to making no differentiation among the two services, we found LR as 40.82 and MOR as 3.82. This shows that using multipath results in higher LR compared to the single path case. In addition to a higher LR, using "Equal Priority Multipath" has a drawback of packet mis-ordering. However, in our model which is named as "Strict Priority Multipath" in the table, LR is 12.00 and MOR is 0.20. In this model, P-LSPs have almost strict priority over S-LSPs so that S-LSPs use the residual capacity remaining from P-LSPs and the knock-on effect is eliminated. The detailed explanation for knock-on effect is given below by demonstrating the steady state traffic rates for each source-destination pair in both strict priority and equal priority cases. The CTRs for the "Single Path" case are not shown because they are very similar to CTRs of the P-LSP in the "Strict Priority Multipath" case.

For the seven-node ring topology, there are 6 P-LSPs and 15 S-LSPs passing through each link if we assume the existence of full connectivity between each

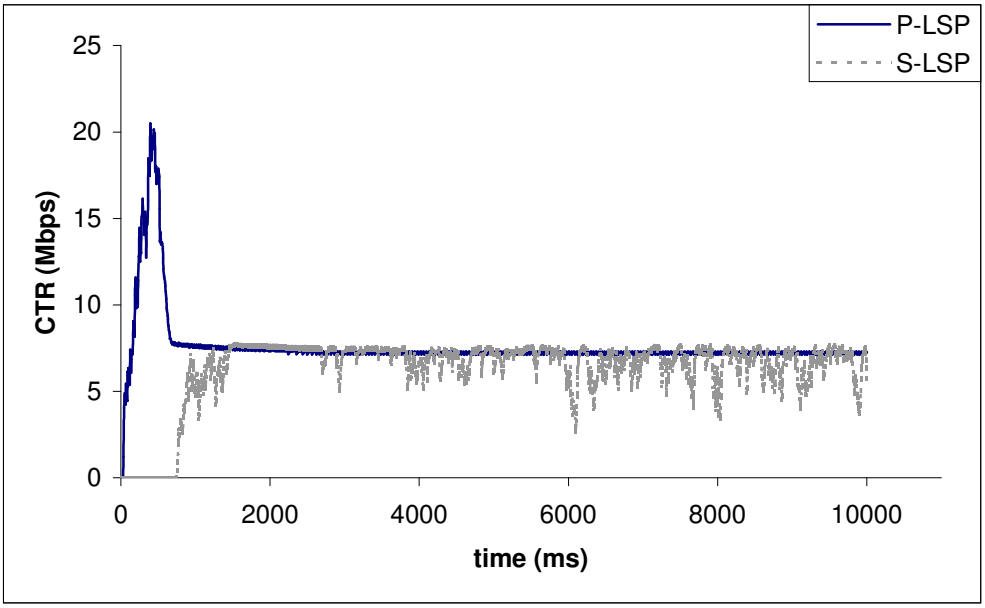

Figure 3.5: Current Traffic Rate graph of the source-destination pair 0-1 in Equal-Priority case.

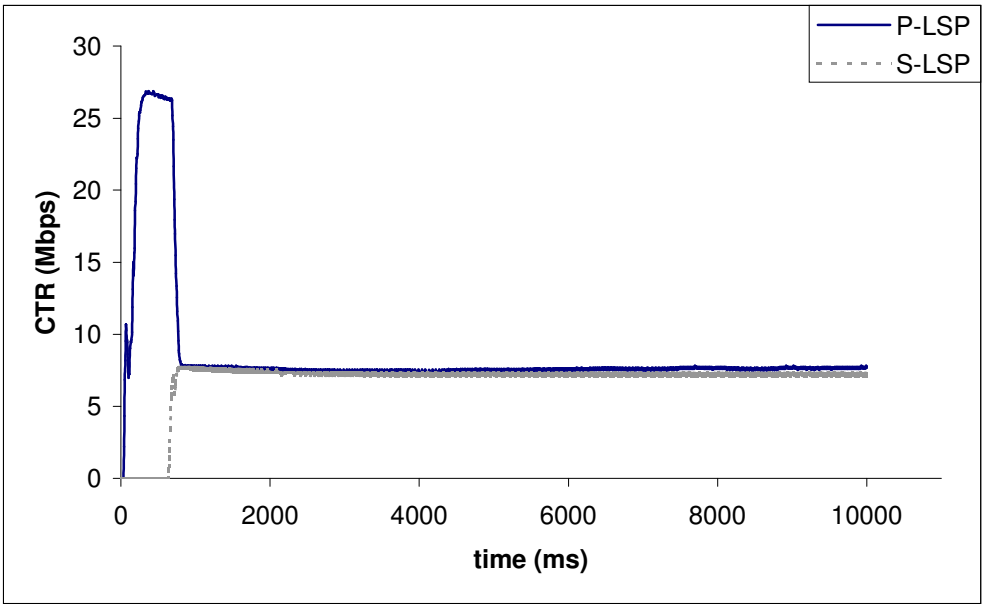

Figure 3.6: Current Traffic Rate graph of the source-destination pair 0-4 in Equal-Priority case.

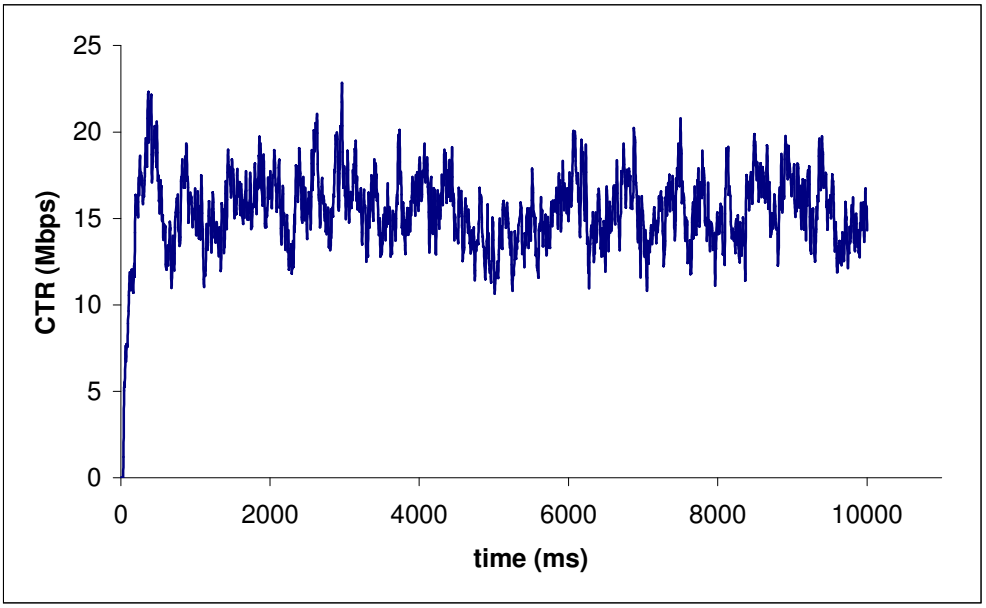

Figure 3.7: Current Traffic Rate graph of the source-destination pair 0-1 in Strict-Priority case for P-LSP.

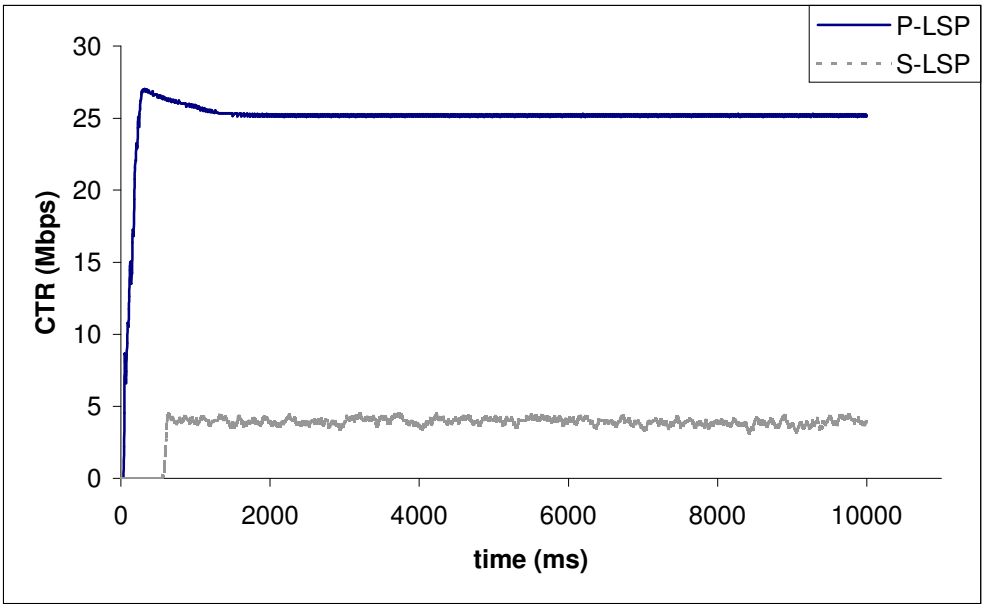

Figure 3.8: Current Traffic Rate graph of the source-destination pair 0-4 in Strict-Priority case.

node. If P-LSPs have no priority over S-LSPs, the CTRs of both P-LSPs and S-LSPs converge to same values. Thus there are 21 LSPs passing through each link and this allows each LSP to have CTR of about (151/21) 7.2 Mb/s. The CTRs of both P-LSP and S-LSP for source-destination pair 0-1 and 0-4 are shown in Figure 3.5 and Figure 3.6 respectively. As seen in figures all CTRs converge to a rate around 7.2 Mb/s for each LSP. The reason for delayed convergence of CTRs for both LSPs is our traffic splitting algorithm which only uses P-LSP until the estimated delay of a packet over P-LSP reaches 140 ms. When traffic is not split during this period, each link capacity is shared by 6 P-LSPs as shown in Figure 3.5 and Figure 3.6.

In the case of P-LSPs having priority over S-LSPs , the CTR of the source-destination pair 0-1 is seen in Figure 3.7. As the shortest-path is in clockwise direction, traffic demand is 15 Mb/s for this pair and link capacity is enough for carrying all generated traffic. Since average CTR is around 15 Mb/s for each P-LSP, the residual capacity for this link will be about 60 Mb/s when we subtract the capacity used by P-LSPs and PPs. There are 15 S-LSPs using this link, the residual capacity remaining from P-LSPs allows 4 Mb/s CTR for each S-LSP as seen in Figure 3.8. The traffic demand for the source-destination pair 0-4 is 35 Mb/s since its shortest-path is in counter-clockwise direction. 25 Mb/s of this traffic is carried using the P-LSP and 4 Mb/s of the remaining 10 Mb/s is carried over the S-LSP.

3.4 Impact of Simulation Parameters

In this section, we will present our simulation results which show how parameters in our traffic splitting algorithm impact the performance measurements, LR and MOR. We focus on five parameters, T_{onmean} and $T_{offmean}$, T_{fix}, D_{max}, and D_{tol}. We used five nodes interconnected to each other using a ring topology depicted

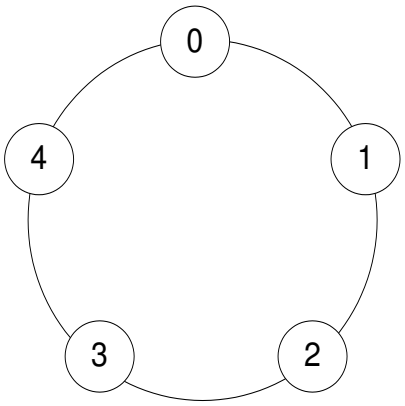

Figure 3.9: 5 node ring topology used in our simulation experiments.

$T_{onmean} = T_{offmean}$	Single Path	Multipath	
	LR	LR	MOR
50	12.13	9.80	0.016
100	12.05	4.11	0.045
200	12.05	1.34	0.051
300	12.15	0.81	0.044
400	12.24	0.57	0.039

Table 3.3: LRs and MORs for different T_{onmean} and $T_{offmean}$ values

in Figure 3.9. Each link is bidirectional and has a capacity of 155 Mbps in both directions. All source-destination pairs have 20 Mbps traffic flow if their shortest paths are in clockwise direction, e.g., source node 0 and destination node 1; otherwise, 60 Mbps traffic flow, e.g., source node 0 and destination node 4. This choice of topology and traffic matrix provides that there exists residual capacity in clockwise direction while capacities in counter-clockwise direction is not enough to carry all traffic demand. Thus, the excessive demand will be carried over the clockwise direction and one can observe the performance of the algorithms using different parameters. We performed our simulations for 15 seconds.

3.4.1 T_{onmean} and $T_{offmean}$

The main purpose of the new multipath TE architecture is to increase throughput while minimizing out of order packets. As a simple traffic splitting algorithm,

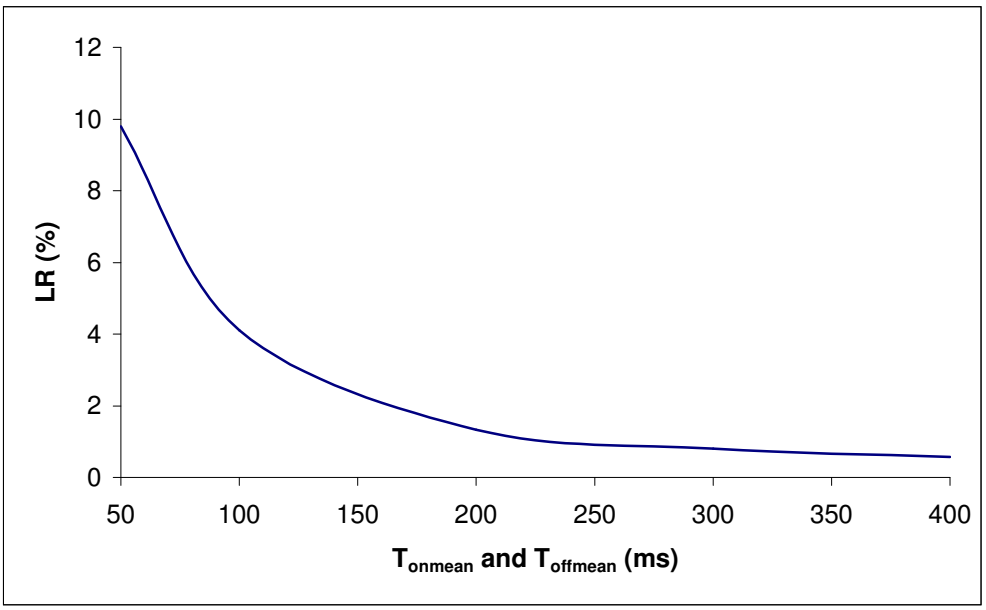

Figure 3.10: The effect of T_{onmean} and $T_{offmean}$ on LR.

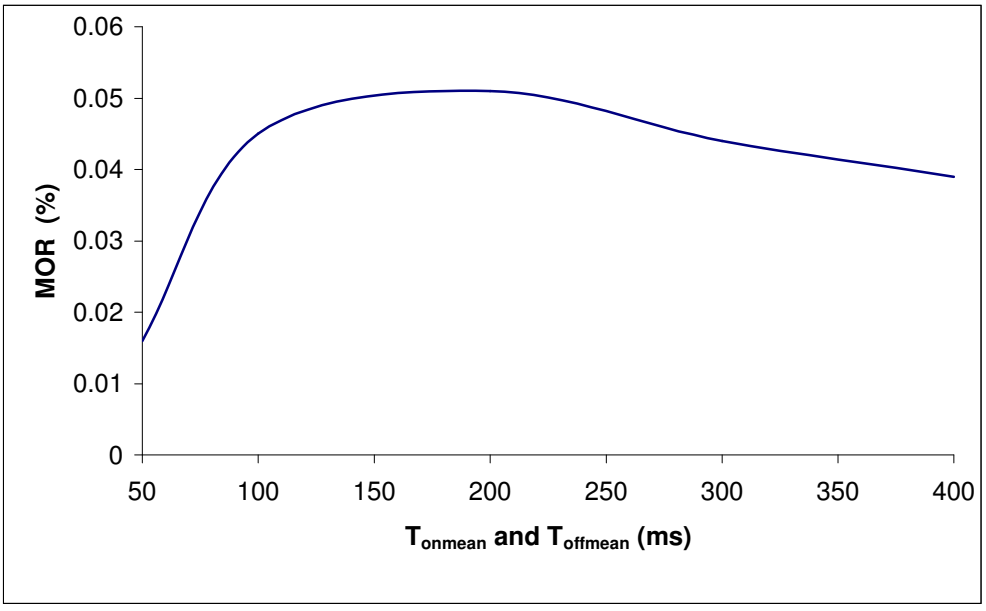

Figure 3.11: The effect of T_{onmean} and $T_{offmean}$ on MOR.

one may check only the occupancy of the silver queue at the edge, and place incoming packets into the bronze queue if the occupancy of the silver queue exceeds a certain threshold. We observe in our simulations that the performance of this scheme is good in terms of LR, but it does not take into account the mis-ordering problem. In order to minimize the out of order packet arrivals, we propose the idea of *burst routing*. We assume that traffic for a given bin is bursty, or equivalently there are gaps between consecutive activity periods. The most important parameters of the burst routing are T_{onmean} and $T_{offmean}$ which represent the mean values of these active and idle periods, respectively. These periods are determined by the number of bins such that the smaller the number of bins, the smaller T_{onmean} and $T_{offmean}$ values are obtained. We assume that T_{onmean} and $T_{offmean}$ are 50 ms in the case the number of bins is 25 and we change T_{onmean} and $T_{offmean}$ values according to the number of bins, i.e. 100 ms if the number of bins is 50. Simulation parameters are selected as follows: D_{max} = 200 ms, T_{fix} = 30 ms and D_{tol}=10 ms.

The impact of T_{onmean} and $T_{offmean}$ on the performance is tabulated in Table 3.3, and shown in Figure 3.10 and Figure 3.11. In fact, $T_{offmean}$ affects directly the performance since it determines the inter-arrival time T_{int} between two consecutive bursts, but we change T_{onmean} simultaneously because the number of bins affects both T_{onmean} and $T_{offmean}$. The higher $T_{offmean}$, the easier a new burst can be identified, and a lower LR results since a burst can be routed independently over one of two LSPs. The amount of traffic carried over the S-LSP affects directly MOR values. If we carry more traffic over the S-LSP, the probability of mis-ordering will increase accordingly. MOR is low for the small values of $T_{offmean}$, since the amount of traffic carried over the S-LSP is small. However, when LRs are close to each other, higher $T_{offmean}$ values result in lower MOR values since the inter-arrival times between two consecutive packets are higher and these two packets can be safely routed over the primary or secondary LSPs. We can conclude that our algorithm performs better for higher $T_{offmean}$ values.

D_{max}	Single Path	Multipath	
	LR	LR	MOR
50	12.69	1.44	0.064
100	12.64	1.23	0.070
150	12.75	1.31	0.063
200	12.05	1.34	0.051
250	12.01	1.82	0.040
300	11.73	3.48	0.032

Table 3.4: LRs and MORs for different D_{max} values

3.4.2 D_{max}

D_{max} denotes the maximum allowed delay for a packet through the MPLS backbone network. The motivation behind using this parameter from our algorithm point of view is to upper-bound D_{LSP1} and D_{LSP2}, and thus make it possible to identify a new burst arrival. From a network point of view, D_{max} is used to restrict delay to a certain upper-bound, since packets are treated as lost if they do not reach the destination within a certain time for TCP flow. We used T_{onmean}=$T_{offmean}$=200 ms, T_{fix}=30 ms and D_{tol}=10 ms as simulation parameters.

The impact of D_{max} on the performance is tabulated in Table 3.4, and shown in Figure 3.12 and Figure 3.13. When D_{max} is set to small values, packets are dropped more easily at the edge while they can be enqueued at edge buffers for higher D_{max} values. In fact, D_{max} value determines the maximum number of bytes which can be enqueued at edge buffers at a certain time. For small D_{max} values, this number is small such that it may not be sometimes possible to send packets at a required rate since there might not be available packets and LR increases. On the other hand, when it is set to large values, it gets difficult to identify a new burst arrival, and this results in higher LRs. The lower MORs seen at the left and right sides of Figure 3.13 are due to the small amount of traffic carried over the S-LSP (high LR). The small amount of traffic carried over the S-LSP decreases the probability of mis-ordering and low MOR values are

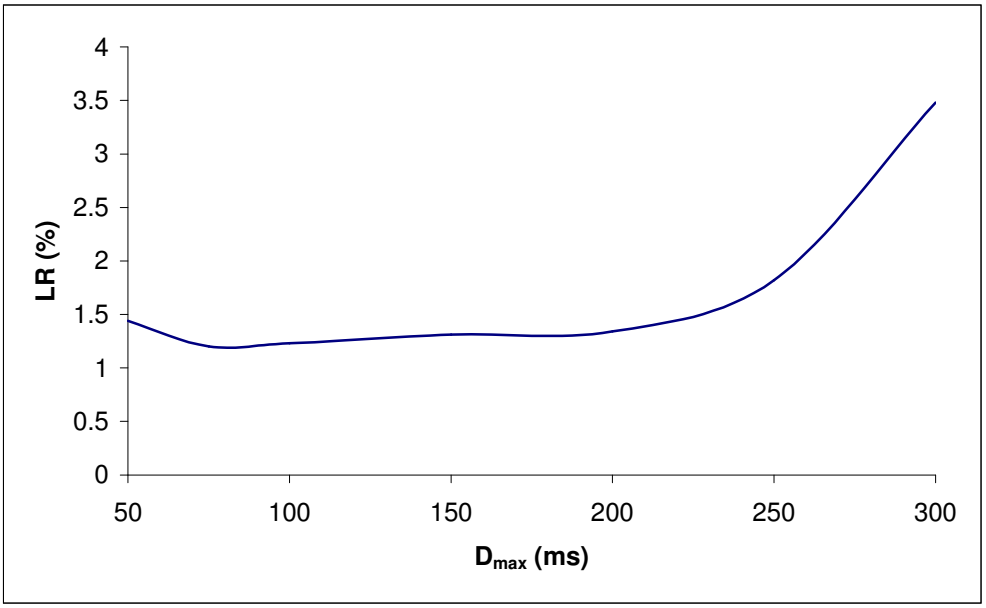

Figure 3.12: The effect of D_{max} on LR.

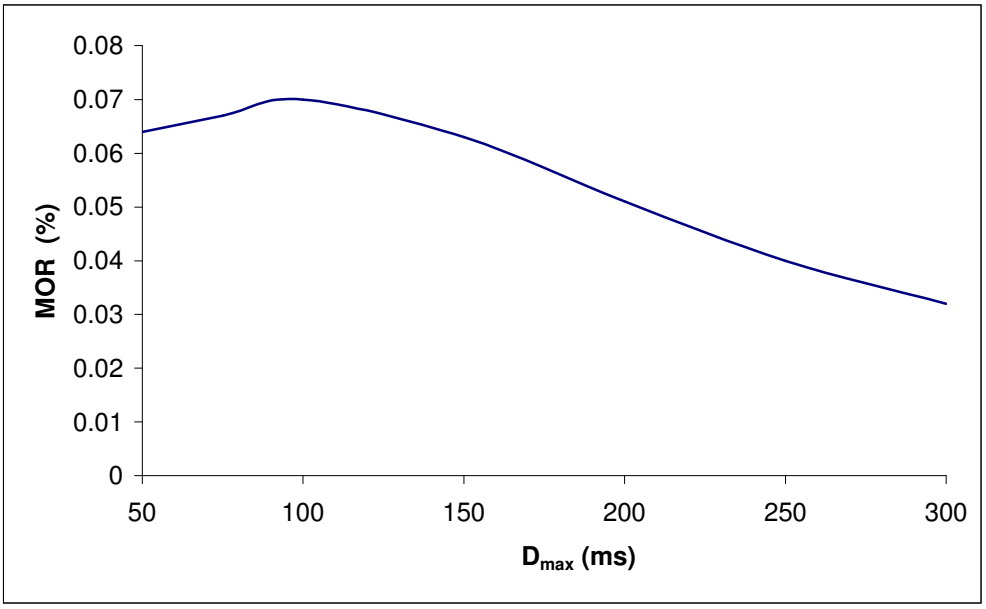

Figure 3.13: The effect of D_{max} on MOR.

T_{fix}	Single Path	Multipath	
	LR	LR	MOR
0	12.99	1.77	0.825
25	13.06	2.06	0.137
50	12.92	2.66	0.003
100	12.71	3.68	0
200	12.76	4.77	0
400	12.90	8.22	0

Table 3.5: LRs and MORs for different T_{fix} values

obtained. When D_{max} is between 100 ms and 200 ms, the LR is relatively flat, but MOR decreases as D_{max} increases. The reason for low MOR values may be that higher D_{max} values make higher D_{LSP1} values while D_{LSP2} values are zero at the beginning of the simulation and this situation decreases the probability of wrong burst identifications. We observe that D_{max} which is chosen slightly less than $T_{offmean}$ provides a good performance.

3.4.3 T_{fix}

T_{thr} in our algorithm gives the difference between the delay estimates of a packet over the primary LSP and the secondary LSP. The delay information is estimated by dividing the number of bytes in a certain service queue to the available capacity of that service found by ERICA algorithm and is brought by PPs after a certain feedback delay. T_{fix} is used to provide a safety margin in case of estimation errors and errors due to feedback delay. We used T_{onmean}=$T_{offmean}$=100 ms, D_{max}=200 ms and D_{tol}=10 ms as simulation parameters.

LR and MOR values for different T_{fix} values are tabulated in Table 3.5, and shown in Figure 3.14 and Figure 3.15. These results clearly show that there is a trade-off between LR and MOR. When we set T_{fix} to 0, we obtained the smallest LR, but the highest MOR. As we increase T_{fix}, LR increases because a higher T_{fix} makes it difficult to identify a new burst and we keep using the primary paths and the excessive traffic is dropped at the edge while it is possible to send

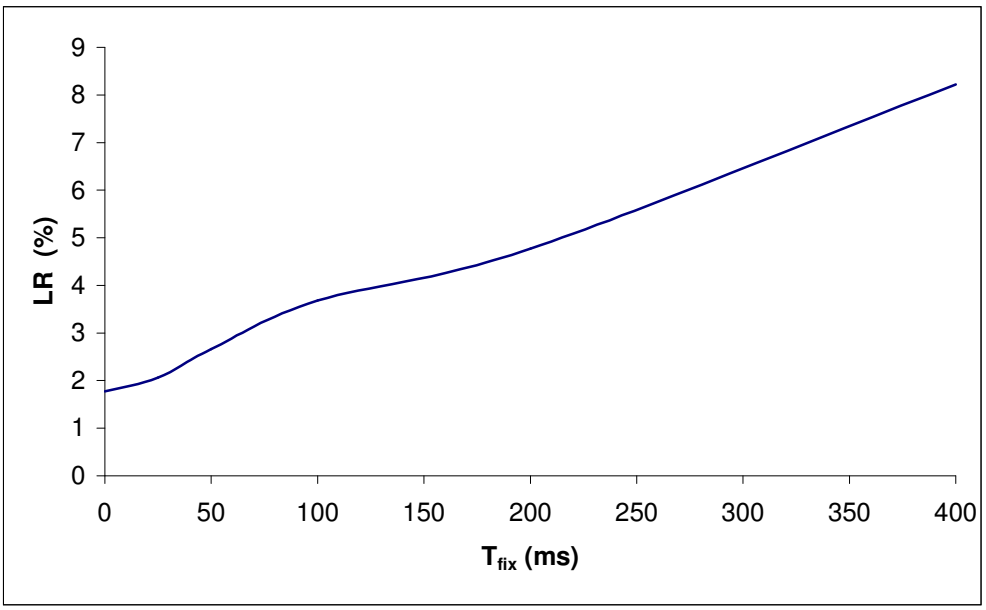

Figure 3.14: The effect of T_{fix} on LR.

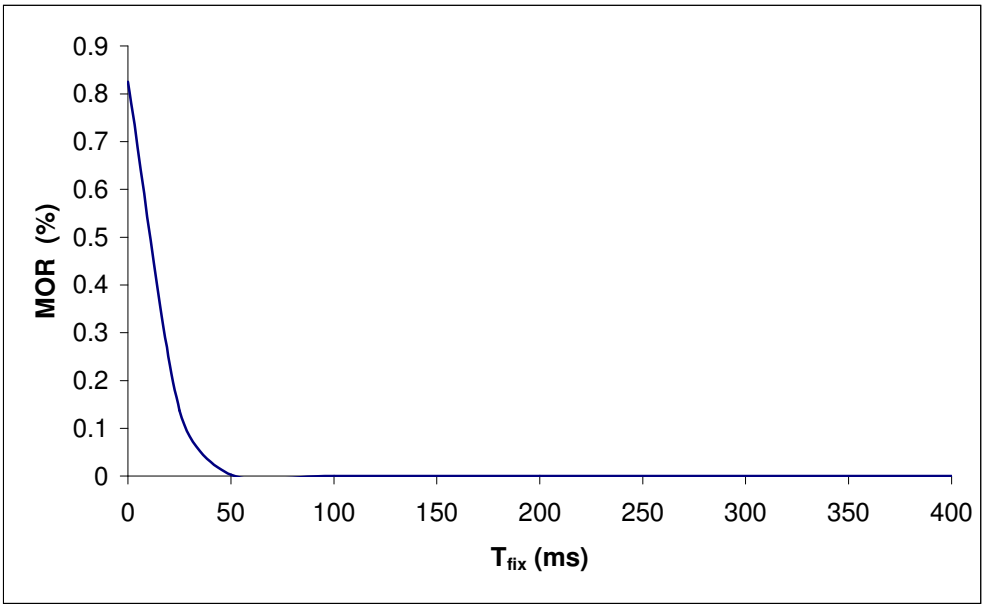

Figure 3.15: The effect of T_{fix} on MOR.

D_{tol}	Single Path	Multipath	
	LR	LR	MOR
0	12.18	12.00	0.008
5	12.70	3.74	0.067
10	12.75	1.31	0.063
20	12.23	0.70	0.077
30	12.64	1.04	0.067

Table 3.6: LRs and MORs for different D_{tol} values

over the secondary paths. However, MOR decreases as T_{fix} increases because we compensate most of the estimation errors by setting T_{fix} conservatively to higher values, and this makes the amount of traffic carried over the S-LSP lower, hence MOR decreases.

3.4.4 D_{tol}

The aim of using the delay tolerance D_{tol} is to prevent unnecessary oscillations of D_{LSP1} around D_{max} and throughput degradation of the S-LSP. We used T_{onmean}=$T_{offmean}$=200 ms, D_{max}=150 ms and T_{fix}=30 ms as simulation parameters.

The impact of D_{tol} on the performance is tabulated in Table 3.6, and shown in Figure 3.16 and Figure 3.17. When the D_{tol} is not used (0 ms), we observe very high LR due to the oscillations of D_{LSP1} around D_{max}. When D_{tol} is not used, the first packet belonging to a new burst will be assigned to the silver queue even D_{LSP1} is slightly less than D_{max}. According to our traffic splitting algorithm, all packets belonging to this burst should be assigned to the same queue in which the first packet is assigned. D_{LSP1} becomes higher than D_{max} after a certain time since more packets are sent over the primary LSP. However, packets are rejected based on the policy dictated by step 3 of the algorithm (if D_{LSP1} is higher than D_{max}) while it may be possible to send through the secondary LSP. Since packets are rejected in the above case, D_{LSP1} will be again

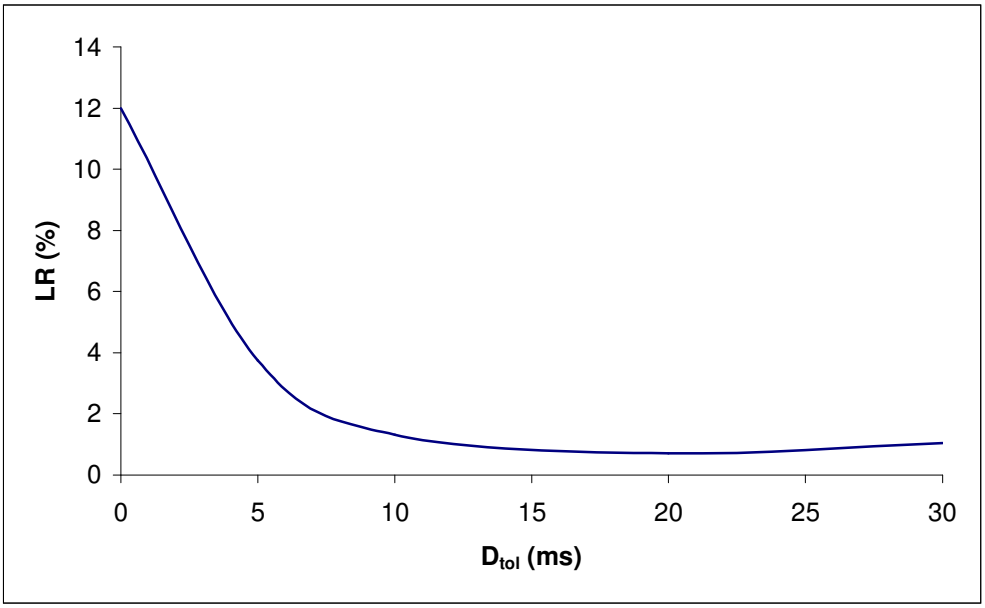

Figure 3.16: The effect of D_{tol} on LR.

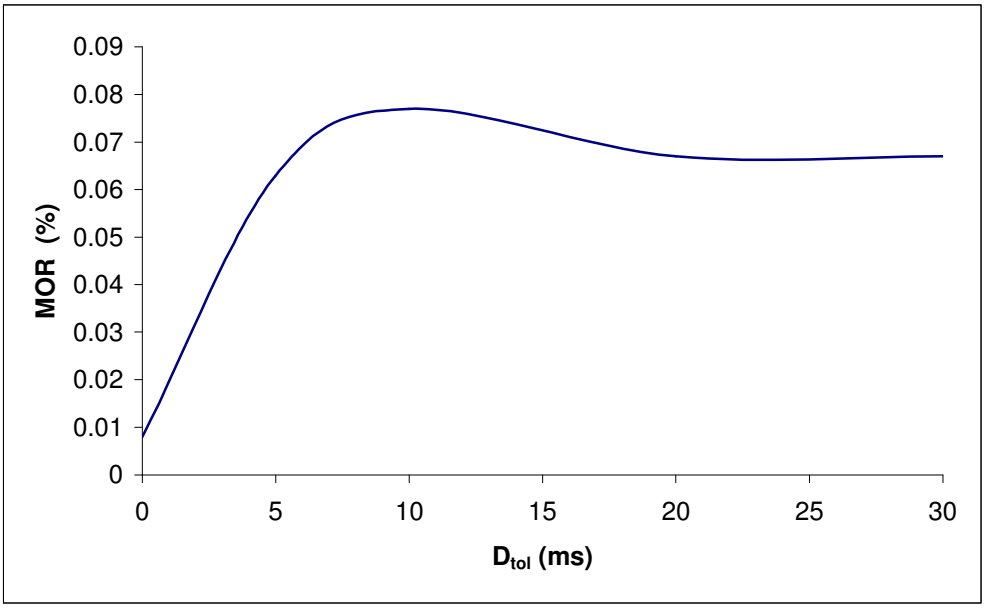

Figure 3.17: The effect of D_{tol} on MOR.

smaller than D_{max} after a certain time and new bursts will be assigned to the silver queue. This situation results in oscillation of D_{LSP1} around D_{max} and this causes throughput degradation of the secondary LSP. As we increase D_{tol} until 10 ms, a significant decrease in LR is observed, but after 10 ms, the effect of D_{tol} is not significant. As D_{tol} increases, higher MOR values are obtained because the amount of traffic carried over the S-LSP increases. From our definition of the MOR value, as the amount of traffic carried over the S-LSP increases, the probability of mis-ordering increases accordingly.

Chapter 4

Conclusions

In this thesis, we proposed a traffic engineering methodology in IP networks using Multiprotocol Label Switching (MPLS) backbones. Available Bit Rate (ABR) service category is generally used for flow control in ATM networks; we here propose to use the Explicit Rate (ER) mode of ABR for MPLS traffic engineering purposes. This methodology is based on a novel concept called the "MPLS-Diffserv" model we introduce in this thesis. Our simulation studies demonstrate that using multiple paths in our proposed architecture provides significantly and consistently better results than the case of a single MPLS LSP in terms of overall throughput in a variety of scenarios. One other challenging issue when multiple paths are used between a source-destination pair is to ensure packet ordering of individual flows in IP networks. We introduce a novel concept, namely the "burst routing" which addresses this problem. By using burst routing, we show that the number of out of order packets is minimized while assuring a significant gain in overall throughput. Future work will consist of further experimentation and some with actual traffic traces.

Appendix A

Appendix

A.1 ERICA Switch Algorithm

The ATM Forum Traffic Management Specification explains in detail the rules for source and destination end system behaviors for the ABR service [15]. The switch behavior is not completely specified to provide the flexibility for switch vendors. Several switch algorithms have been designed [37]-[41]. In our study, we use the Explicit Rate Indication for Congestion Avoidance (ERICA) algorithm [42].

In ERICA algorithm, the switch periodically monitors the load on each link and determines a load factor (z), the available ABR capacity, and the number of currently active virtual connections. An averaging interval is used for this purpose.

$$\textit{Total ABR Capacity} \leftarrow \textit{Link Capacity} - \textit{VBR and CBR Capacity} \qquad \text{(A.1)}$$

$$Target\ ABR\ Capacity \leftarrow Fraction \times Total\ ABR\ Capacity \tag{A.2}$$

$$z \leftarrow \frac{ABR\ Input\ Rate}{Target\ ABR\ Capacity} \tag{A.3}$$

$$FairShare \leftarrow \frac{Target\ ABR\ Capacity}{Number\ of\ Active\ VCs} \tag{A.4}$$

$$VCShare \leftarrow \frac{CCR[VC]}{z} \tag{A.5}$$

$$MaxAllocPrevious \leftarrow MaxAllocCurrent \tag{A.6}$$

$$MaxAllocCurrent \leftarrow FairShare \tag{A.7}$$

The above steps are executed at the end of the switch averaging interval. The available ABR capacity is the capacity left from VBR and CBR bandwidth usage (A.1). The load factor is calculated as the ratio of the measured input rate at the port to the target ABR capacity of the output link (A.3) . The target ABR capacity is a fraction of the total ABR capacity (A.2). *Fraction* can be a constant number such as 0.9, or it can be a function of the queueing delay at that port *f(Q)*. This function allows only a selected fraction of the available capacity to be allocated to the sources. The remaining capacity is used to drain the current queue. In the following equations, Q is the current queue length and $Q0$ represents the target queue length.

$$f(Q) = \frac{a \times Q}{(a-1) \times Q + Q0} \quad for \quad Q > Q0 \tag{A.8}$$

$$f(Q) = \frac{b \times Q0}{(b-1) \times Q + Q0} \quad for \quad 0 \leq Q \leq Q0 \tag{A.9}$$

$$f(Q) = Max(QDLF, \frac{a \times Q}{(a-1) \times Q + Q0}) \quad for \quad Q > Q0 \tag{A.10}$$

f(Q) is a number between 1 and 0 in the range Q0 to infinity (A.8), and between b and 1 in the range 0 to Q0 (A.9). This function is lower bounded by the queue drain limit function (QDLF) (A.10). The aim of using this function instead of a constant number is that the constant number (e.g. 0.9) is not able to utilize the system fully in the steady state. However, this function adapts itself according to the current queue length and tries to keep the system at the target utilization.

The fair share of each VC, *FairShare*, is calculated as in (A.4). Intuitively, *FairShare* is the minimum share which every active source deserves. If the source does not use all of its *FairShare*, then switch fairly allocates the remaining capacity to the sources that can use it. Thus the switch scales the current cell rate (CCR) of the connection by the load factor as in (A.5).

To achieve max-min fairness, ERICA maintains the highest allocation given to any VC on this output port during each averaging interval and ensures that all eligible VCs can also get this high allocation. The variable *MaxAllocPrevious* stores the maximum allocation given in the previous interval, and *MaxAlloc-Current* accumulates the maximum allocation given during the current switch averaging interval.

$$ER \leftarrow Max(FairShare,\ VCShare) \quad for \quad z > 1 + \delta \tag{A.11}$$

$$ER \leftarrow Max(FairShare,\ VCShare,\ MaxAllocPrevious) \quad for \quad z \leq 1+\delta \tag{A.12}$$

$$ER \leftarrow Min(ER,\ Target\ ABR\ Capacity) \tag{A.13}$$

Thus, VCs are given equal allocations during underload (A.12) and the CCRs are divided by the same load factor z during the subsequent overload to bring the sources to their max-min fair shares (A.11). The system is to be considered in a state of overload when its load factor z is greater than $1 + \delta$. The aim of introducing the quantity δ is to force the allocation of equal rates when the overload is fluctuating around unity, thus avoiding unnecessary rate oscillations. Explicit rate (ER) allocation can not be higher than the target ABR capacity as in (A.13).

A.2 The Mean Values of CTRs for the Hypothetical US Topology

In the tables below, we tabulated the demands and the mean values of the CTRs for the P-LSPs and S-LSPs for every source-destination pair. The CTR1 and CTR2 show the mean values of the CTRs for the P-LSP and the S-LSP, respectively. The demand shows the mean of the actual traffic in Mb/s generated by the simulator from a source to a destination.

From-to	**Demand (Mb/s)**	**CTR1 (Mb/s)**	**CTR2 (Mb/s)**
ny-sf	49.579	11.628	17.213
ny-la	38.946	14.777	18.539
ny-dc	38.117	23.122	13.628
ny-sj	37.612	12.505	16.890
ny-ch	33.322	33.322	0
ny-da	23.442	12.501	10.000
ny-hs	16.852	14.732	2.007
ny-at	13.020	13.020	0
ny-cl	11.119	11.119	0
ny-sl	9.715	9.715	0
ny-de	6.013	6.013	0

From-to	Demand (Mb/s)	CTR1 (Mb/s)	CTR2 (Mb/s)
dc-ny	27.820	23.000	4.570
dc-sf	25.560	11.640	12.704
dc-la	20.767	14.772	5.398
dc-sj	19.842	12.493	6.793
dc-ch	18.028	18.028	0
dc-da	12.323	12.250	0.067
dc-hs	9.293	9.293	0
dc-at	7.241	7.241	0
dc-cl	6.059	6.059	0
dc-sl	5.271	5.271	0
dc-de	3.358	3.358	0

From-to	Demand (Mb/s)	CTR1 (Mb/s)	CTR2 (Mb/s)
cl-ny	11.211	11.211	0
cl-sf	10.564	10.564	0
cl-la	8.291	8.291	0
cl-dc	7.984	7.984	0
cl-sj	8.220	8.220	0
cl-ch	7.205	7.205	0
cl-da	5.354	5.354	0
cl-hs	4.040	4.040	0
cl-at	3.210	3.210	0
cl-sl	2.299	2.299	0
cl-de	1.525	1.525	0

From-to	Demand (Mb/s)	CTR1 (Mb/s)	CTR2 (Mb/s)
ch-ny	36.374	36.374	0
ch-sf	34.690	23.173	10.624
ch-la	27.461	27.461	0
ch-dc	27.210	27.210	0
ch-sj	26.206	26.206	0
ch-da	16.633	12.430	3.832
ch-hs	12.112	12.112	0
ch-at	9.534	9.534	0
ch-cl	8.069	8.069	0
ch-sl	6.916	6.916	0
ch-de	4.246	4.246	0

From-to	Demand (Mb/s)	CTR1 (Mb/s)	CTR2 (Mb/s)
de-ny	5.874	5.874	0
de-sf	5.469	5.469	0
de-la	4.487	4.487	0
de-dc	4.428	4.428	0
de-sj	4.351	4.351	0
de-ch	3.750	3.750	0
de-da	2.864	2.864	0
de-hs	2.195	2.195	0
de-at	1.804	1.804	0
de-cl	1.468	1.468	0
de-sl	1.351	1.351	0

From-to	Demand (Mb/s)	CTR1 (Mb/s)	CTR2 (Mb/s)
sf-ny	52.411	11.628	15.616
sf-la	39.134	39.134	0
sf-dc	37.800	11.641	13.658
sf-sj	37.618	37.658	0
sf-ch	33.151	23.138	9.124
sf-da	23.805	22.922	0.842
sf-hs	17.284	17.284	0
sf-at	13.005	13.005	0
sf-cl	11.221	11.221	0
sf-sl	9.328	9.328	0
sf-de	5.864	5.864	0

From-to	Demand (Mb/s)	CTR1 (Mb/s)	CTR2 (Mb/s)
la-ny	41.075	14.756	15.545
la-sf	38.140	38.140	0
la-dc	29.914	14.769	13.567
la-sj	29.869	29.869	0
la-ch	26.960	26.960	0
la-da	18.820	17.949	0.861
la-hs	14.153	14.153	0
la-at	10.438	10.438	0
la-cl	8.806	8.806	0
la-sl	8.048	8.048	0
la-de	4.763	4.763	0

From-to	Demand (Mb/s)	CTR1 (Mb/s)	CTR2 (Mb/s)
hs-ny	18.544	14.697	3.528
hs-sf	17.444	17.444	0
hs-la	14.142	14.142	0
hs-dc	13.699	13.699	0
hs-sj	13.461	13.461	0
hs-ch	12.017	12.017	0
hs-da	8.513	8.513	0
hs-at	5.103	5.103	0
hs-cl	4.233	4.233	0
hs-sl	3.801	3.801	0
hs-de	2.332	2.332	0

From-to	Demand (Mb/s)	CTR1 (Mb/s)	CTR2 (Mb/s)
at-ny	16.114	14.625	1.317
at-sf	14.739	14.739	0
at-la	12.288	12.288	0
at-dc	11.649	11.649	0
at-sj	11.948	11.948	0
at-ch	10.643	10.643	0
at-da	7.548	7.548	0
at-hs	5.755	5.755	0
at-cl	3.728	3.728	0
at-sl	3.197	3.197	0
at-de	2.038	2.038	0

From-to	Demand (Mb/s)	CTR1 (Mb/s)	CTR2 (Mb/s)
sj-ny	31.327	12.492	15.234
sj-sf	29.869	29.869	0
sj-la	23.926	23.926	0
sj-dc	22.793	12.483	9.366
sj-ch	20.711	20.711	0
sj-da	14.599	14.599	0
sj-hs	10.568	10.568	0
sj-at	7.994	7.994	0
sj-cl	6.808	6.808	0
sj-sl	5.963	5.963	0
sj-de	3.743	3.743	0

From-to	Demand (Mb/s)	CTR1 (Mb/s)	CTR2 (Mb/s)
da-ny	27.080	12.504	13.242
da-sf	24.476	23.083	1.375
da-la	19.971	17.966	1.931
da-dc	19.271	12.489	6.006
da-sj	18.494	18.494	0
da-ch	16.739	12.493	3.908
da-hs	8.954	8.954	0
da-at	7.055	7.055	0
da-cl	5.749	5.749	0
da-sl	5.188	5.188	0
da-de	3.101	3.101	0

From-to	Demand (Mb/s)	CTR1 (Mb/s)	CTR2 (Mb/s)
sl-ny	10.946	10.946	0
sl-sf	10.204	10.204	0
sl-la	8.456	8.456	0
sl-dc	8.261	8.261	0
sl-sj	8.164	8.164	0
sl-ch	7.312	7.312	0
sl-da	5.166	5.166	0
sl-hs	3.898	3.898	0
sl-at	3.018	3.018	0
sl-cl	2.727	2.727	0
sl-de	1.519	1.519	0

A.3 Primary and Secondary Paths for the Hypothetical US Topology

In the tables below, we tabulated the primary paths and the secondary paths for every source-destination pair. If the primary and secondary paths from ny to dc are tabulated, the primary and secondary paths from dc to ny are not included in the tables since they use the same nodes used by the two paths from dc to ny but in reverse order. The secondary path from ny to dc is the path ny-cl-dc, then the secondary path from dc to ny is the path dc-cl-ny. Therefore, we tabulate only half of the paths for all source-destination pairs.

From-to	Primary Path	Secondary Path
ny-dc	ny-dc	ny-cl-dc
ny-cl	ny-cl	ny-dc-cl
ny-ch	ny-cl-ch	ny-dc-sl-de-ch
ny-de	ny-dc-sl-de	ny-cl-ch-de
ny-sf	ny-dc-sl-de-sf	ny-cl-ch-de-sj-sf
ny-la	ny-dc-at-hs-la	ny-cl-ch-de-sf-la
ny-hs	ny-dc-at-hs	ny-cl-sl-da-hs
ny-at	ny-dc-at	ny-cl-sl-da-hs-at
ny-sj	ny-dc-sl-da-sj	ny-cl-ch-de-sj
ny-da	ny-cl-sl-da	ny-dc-at-hs-da
ny-sl	ny-cl-sl	ny-dc-sl

From-to	Primary Path	Secondary Path
dc-cl	dc-cl	dc-ny-cl
dc-ch	dc-cl-ch	dc-sl-de-ch
dc-de	dc-sl-de	dc-cl-ch-de
dc-sf	dc-sl-de-sf	dc-at-hs-la-sf
dc-la	dc-at-hs-la	dc-sl-da-sj-la
dc-hs	dc-at-hs	dc-sl-da-hs
dc-at	dc-at	dc-sl-da-hs-at
dc-sj	dc-sl-da-sj	dc-cl-ch-de-sj
dc-da	dc-sl-da	dc-at-hs-da
dc-sl	dc-sl	dc-cl-sl

From-to	Primary Path	Secondary Path
cl-ch	cl-ch	cl-sl-de-ch
cl-de	cl-sl-de	cl-ch-de
cl-sf	cl-sl-de-sf	cl-ch-de-sj-sf
cl-la	cl-dc-at-hs-la	cl-ch-de-sf-la
cl-hs	cl-sl-da-hs	cl-dc-at-hs
cl-at	cl-dc-at	cl-sl-da-hs-at
cl-sj	cl-sl-da-sj	cl-ch-de-sj
cl-da	cl-sl-da	cl-dc-at-hs-da
cl-sl	cl-sl	cl-dc-sl

From-to	Primary Path	Secondary Path
ch-de	ch-de	ch-cl-sl-de
ch-sf	ch-de-sf	ch-cl-sl-de-sj-sf
ch-la	ch-de-sj-la	ch-cl-dc-at-hs-la
ch-hs	ch-cl-sl-da-hs	ch-de-sf-la-hs
ch-at	ch-cl-dc-at	ch-de-sl-da-hs-at
ch-sj	ch-de-sj	ch-cl-sl-da-sj
ch-da	ch-cl-sl-da	ch-de-sj-da
ch-sl	ch-de-sl	ch-cl-sl

From-to	Primary Path	Secondary Path
de-sf	de-sf	de-sj-sf
de-la	de-sj-la	de-sf-la
de-hs	de-sl-da-hs	de-sf-la-hs
de-at	de-sl-dc-at	de-sf-la-hs-at
de-sj	de-sj	de-sf-sj
de-da	de-sl-da	de-sj-da
de-sl	de-sl	de-ch-de-sl

From-to	Primary Path	Secondary Path
sf-la	sf-la	sf-sj-la
sf-hs	sf-la-hs	sf-sj-da-hs
sf-at	sf-la-hs-at	sf-de-sl-dc-at
sf-sj	sf-sj	sf-de-sj
sf-da	sf-sj-da	sf-de-sl-da
sf-sl	sf-de-sl	sf-sj-da-sl

From-to	Primary Path	Secondary Path
la-hs	la-hs	la-sj-da-hs
la-at	la-hs-at	la-sf-de-sl-dc-at
la-sj	la-sj	la-sf-sj
la-da	la-hs-da	la-sj-da
la-sl	la-sf-de-sl	la-hs-da-sl

From-to	Primary Path	Secondary Path
hs-at	hs-at	hs-da-sl-dc-at
hs-sj	hs-da-sj	hs-la-sj
hs-da	hs-da	hs-la-sj-da
hs-sl	hs-da-sl	hs-at-dc-sl

From-to	Primary Path	Secondary Path
at-sj	at-hs-da-sj	at-dc-sl-de-sj
at-da	at-hs-da	at-dc-sl-da
at-sl	at-dc-sl	at-hs-da-sl

From-to	Primary Path	Secondary Path
sj-da	sj-da	sj-de-sl-da
sj-sl	sj-de-sl	sj-da-sl

From-to	Primary Path	Secondary Path
da-sl	da-sl	da-sj-de-sl

Bibliography

[1] D. O. Awduche, "MPLS and Traffic Engineering in IP Networks", IEEE Communications Magazine, vol. 37, pp. 42-47, December 1999.

[2] D. O. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering", Internet Draft <draft-ietf-tewg-principles-01.txt>, work in progress.

[3] J. F. Kurose and K. W. Ross, "Computer Networking: A Top Down Approach Featuring the Internet", Addison Wesley Longman, 2001.

[4] J. Moy, "OSPF Version 2", RFC 2178, 1997.

[5] D. Oran, "OSI IS-IS Intra-domain Routing Protocol", RFC 1142, 1990.

[6] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, 1995.

[7] R. O. Onvural, "Asynchronous Transfer Mode Networks", Artech House, 1995.

[8] U. D. Black, "ATM: Foundation For Broadband Networks", Prentice Hall, 1995.

[9] B. Davie and Y. Rekhter, "MPLS Technology and Application", Morgan Kaufmann Publishers, 2000.

[10] B. Davie, P. Doolan, and Y. Rekhter, "Switching in IP Networks", Morgan Kaufmann Publishers, 1998.

[11] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[12] E. Ayanoglu and N. Akar, "B-ISDN (Broadband Integrated Services Digital Network) ", to appear at Wiley Encyclopedia of Telecommunication, 2003.

[13] ATM Forum, "ATM User-Network Interface (UNI) Specification Version 4.0", ATM Forum specfication af-sig-0061.000, July 1996.

[14] ATM Forum, "Private Network-Network Interface Specification Version 1.0", AF-PNNI-0055, 1996.

[15] ATM Forum, Traffic Management Specification Version 4.0, AF-TM-0056.000, April 1996.

[16] Y. Wang, Z. Wang, and L. Zhang, "Internet traffic engineering without full mesh overlaying", Proceedings of INFOCOM'2001, Anchorage, USA, 2001.

[17] F. Le Faucheur et al, "MPLS Support of Differentiated Services", Internet Draft <draft-ietf-mpls-diff-ext-09.txt>, 2001.

[18] Cisco, "Configuring OSPF", 1997.

[19] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights", Proceedings of INFOCOM'2000, Tel-Aviv, Israel, 2000.

[20] LTM Berry, S. Kohler, D. Staehle, and P. Trangia, "Fast heuristics for optimal routing in IP networks", Universitat Wurzburg Institut fur Informatik Research Report Series, Report No. 262, July 2000.

[21] Y. Wang and Z. Wang, "Explicit routing algorithms for internet traffic engineering", Proceedings of ICCCN'99, September 1999.

[22] C. Villamizar, "OSPF Optimized Multipath (OSPF-OMP)", Internet Draft <draft-ietf-ospf-omp-02.txt>, 1998.

[23] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "MATE: MPLS Adaptive Traffic Engineering", Proceedings of INFOCOM'2001, Anchorage, USA, 2001.

[24] S. Plotkin, "Competetive routing of virtual circuits in ATM networks", IEEE Jour. Selected Areas in Comm., pp. 1128-1136, 1995.

[25] M. Kodialam and T. V. Lakshman, "Minimum interference routing with applications to MPLS traffic engineering", Proceedings of INFOCOM'2000, Tel-Aviv, Israel, March 2000.

[26] Xipeng Xiao, A. Hannan, B. Bailey, S. Carter, and L. M. Ni, "Traffic Engineering with MPLS in the Internet", IEEE Network Magazine, pp. 28-33, March 2000.

[27] C. Villamizar, "MPLS Optimized Multipath (MPLS-OMP)", Internet Draft <draft-ietf-mpls-omp-01.txt>, 1999.

[28] Y. Lee, Y. Seok, Y. Choi, and C. Kim, "A Constrained Multipath Traffic Engineering Scheme for MPLS Networks", IEEE ICC 2002, New York, May 2002.

[29] J. Wang, S. Patek, H. Wang, and J. Liebeherr, "Traffic Engineering with AIMD in MPLS Networks", In Proceedings of Protocols for High Speed Networks, pp. 192-210, 2002.

[30] A. Shaikh, J. Rexford, and K. G. Shin, "Load-Sensitive Routing of Long-Lived IP Flows", SIGCOMM'99, pp. 215-226, 1999.

[31] S. Bahk and M. Zarki, "Dynamic Multipath Routing and How it Compares with other Dynamic Routing Algorithms for High Speed Wide Area Networks", Computer Communications Review, vol. 22, no. 4, Oct 1992.

[32] S. Nelakuditi, Z. L. Zhang, and R. P. Tsang, "Adaptive proportional routing: A localized QoS routing approach", Proceedings of INFOCOM'2000, Anchorage, USA, 2000.

[33] F. P. Kelly, "Routing in circuit switched network: optimization, shadow prices and decentralization", Advances in Applied Probability, vol. 20, pp. 112-144, 1988.

[34] M. Laor and L. Gendel, "The Effect of Packet Reordering in a Backbone Link on Application Throughput", IEEE Network Magazine, vol. 16, no. 5, September 2002.

[35] Juniper Networks, "Supporting Differentiated Service Classes: Queue Scheduling Disciplines", white paper available at www.juniper.net/techcenter/techpapers/200020-06.html, 2002.

[36] NLANR, "WAN packet size distribution", white paper available at http://www.nlanr.net/NA/Learn/packetsizes.html, 2002.

[37] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan, "An efficient rate allocation algorithm for ATM networks providing maxmin fairness", In Proceedings of the 6th IFIP International Conference on High Performance Networking, September 1995.

[38] K. Siu and T. Tzeng, "Intelligent congestion control for ABR service in ATM networks", Computer Communication Review, Volume 24, No. 5, pp. 81-106, October 1995.

[39] D. Tsang and W. Wong, "A new ratebased switch algorithm for ABR traffic to achieve maxmin fairness with analytical approximation and delay adjustment", In Proceedings of the 15th IEEE INFOCOMM, pp. 1174-1181, March 1996.

[40] A. Charny, D. D. Clark, and R. Jain, "Congestion Control with Explicit Rate Indication", In Proceedings of ICC'95, June 1995.

[41] A. Arulambalam, X. Chen, and N. Ansari, "Allocating Fair Rates for Available Bit Rate Service in ATM Networks", IEEE Communications Magazine, vol. 34, pp. 92-100, November 1996.

[42] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, and B. Vandalore, "The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks", IEEE/ACM Transactions on Networking, Vol. 8, No.1, pp. 87-98, February 2000.

[43] Z. Cao, Z. Wang, and E. Zegura, "Performance of Hashing-Based Schemes for Internet Load Balancing", Proceedings of IEEE INFOCOM, March 2000.

[44] "Optimized Multipath", Internet drafts, simulations, examples, and tutorials available at www.fictitious.org/omp, 2002.