Online Traffic Engineering for MPLS Networks

Marléne Botha

Thesis presented in partial fulfilment
of the requirements for the degree of
Master of Science
at the University of Stellenbosch

Prof. A. E. Krzesinski
April 2004

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature: Date:

Abstract

The Internet is fast evolving into a commercial platform that carries a mixture of narrow- and broadband applications such as voice, video, and data. Users expect a certain level of guaranteed service from their service providers and consequently the need exists for efficient Internet traffic engineering to enable better Quality of Service (QoS) capabilities.

Multi-protocol Label Switching (MPLS) is a label switching protocol that has emerged as an enabling technology to achieve efficient traffic engineering for QoS management in IP networks. The ability of the MPLS protocol to create explicit virtual connections called Label Switched Paths (LSPs) to carry network traffic significantly enhances the traffic engineering capabilities of communication networks. The MPLS protocol supports two options for explicit LSP selection: offline LSP computation using an optimization method and dynamic route selection where a single node makes use of current available network state information in order to compute an explicit LSP online.

This thesis investigates various methods for the selection of explicit bandwidth guaranteed LSPs through dynamic route selection. We address the problem of computing a sequence of optimal LSPs where each LSP can carry a specific traffic demand and we assume that no prior information regarding the future traffic demands are available and that the arrival sequence of LSP requests to the network is unknown. Furthermore, we investigate the rerouting abilities of the online LSP selection methods to perform MPLS failure restoration upon link failure.

We propose a new online routing framework known as Least Interference Optimization (LIO) that utilizes the current bandwidth availability and traffic flow distribution to achieve efficient traffic engineering. We present the Least Interference Optimization Algorithm (LIOA) that reduces the interference among competing network flows by balancing the number and quantity of flows carried by a link for the setup of bandwidth guaranteed LSPs in MPLS networks.

The LIOA routing strategy is evaluated and compared against well-known routing strategies such as the

Minimum Hop Algorithm (MHA), Minimum Interference Routing Algorithm (MIRA), Open Shortest Path First (OSPF) and Constraint Shortest Path First (CSPF) by means of simulation.

Simulation results revealed that, for the network topologies under consideration, the routing strategies that employed dynamic network state information in their routing decisions (LIOA, CSPF and MIRA) generally outperformed the routing strategies that only rely on static network information (OSPF and MHA). In most simulation experiments the best performance was achieved by the LIOA routing strategy while the MHA performed the worse. Furthermore we observed that the computational complexity of the MIRA routing strategy does not translate into equivalent performance gains.

We employed the online routing strategies for MPLS failure recovery upon link failure. In particular we investigated two aspects to determine the efficiency of the routing strategies for MPLS rerouting: the suitability of the LSP configuration that results due to the establishment of LSPs prior to link failure and the ability of the online routing strategy to reroute failed LSPs upon link failure. Simulation results revealed similar rerouting performance for all online routing strategies under investigation, but a LSP configuration most suitable for online rerouting was observed for the LIOA routing strategy.

Opsomming

Die Internet is voordurend besig om te evoleer in 'n medium wat 'n wye reeks moderne kommunikasietegnologieë ondersteun, insluitende telefoon, video en data. Internet gebruikers verwag gewaarborgde diens van hul diensverskaffers en daar bestaan dus 'n vraag na doeltreffende televerkeerbeheer vir gewaarborgde Internet diensgehalte.

Multiprotokol Etiketskakeling (MPLS) is 'n etiketskakeling protokol wat doeltreffende televerkeerbeheer en diensgehalte moontlik maak deur die eksplisiete seleksie van virtuele konneksies vir die transmissie van netwerkverkeer in Internetprotokol (IP) netwerke. Hierdie virtuele konneksies staan bekend as etiketgeskakelde paaie. Die MPLS protokol ondersteun tans twee moontlikhede vir eksplisiete seleksie van etiketgeskakelde paaie: aflyn padberekening met behulp van optimeringsmetodes en dinamiese aanlyn padseleksie waar 'n gekose node 'n eksplisiete pad bereken deur die huidige stand van die netwerk in ag te neem.

In hierdie tesis word verskeie padseleksiemetodes vir die seleksie van eksplisiete bandwydte-gewaarborgde etiketgeskakelde paaie deur middel van dinamiese padseleksie ondersoek. Die probleem om 'n reeks optimale etiketgeskakelde paaie te bereken wat elk 'n gespesifeerde verkeersaanvraag kan akkommodeer word aangespreek. Daar word aanvaar dat geen informasie in verband met die toekomstige verkeersaanvraag bekend is nie en dat die aankomsvolgorde van etiketgeskakelde padversoeke onbekend is. Ons ondersoek verder die heroeteringsmoontlikhede van die aanlyn padseleksiemetodes vir MPLS foutrestorasie in die geval van skakelonderbreking.

Vir hierdie doel word 'n nuwe aanlyn roeteringsraamwerk naamlik Laagste Inwerking Optimering (LIO) voorgestel. LIO benut die huidige beskikbare bandwydte en verkeersvloeidistribusie van die netwerk om doeltreffende televerkeerbeheer moontlik te maak. Ons beskryf 'n Laagste Inwerking Optimering Algoritme (LIOA) wat die inwerking tussen kompeterende verkeersvloei verminder deur 'n balans te handhaaf tussen die aantal en kwantiteit van die verkeersvloeistrome wat gedra word deur elke netwerkskakel.

Die LIOA roeteringstrategie word geëvalueer met behulp van simulasie en die resultate word vergelyk met ander bekende roeteringstrategieë insluitende die Minimum Node Algorithme (MHA), die Minimum Inwerking Algoritme (MIRA), die Wydste Kortste Pad Eerste Algoritme (OSPF) en die Beperkte Kortste Pad Eerste Algoritme (CSPF).

Die resultate van die simulasie-eksperimente toon dat, vir die netwerk topologieë onder eksperimentasie, die roeteringstratgieë wat roeteringsbesluite op dinamiese netwerk informasie baseer (LIOA, MIRA, CSPF) oor

die algemeen beter vaar as die wat slegs staatmaak op statiese netwerkinformasie (MHA, OSPF). In die meeste simulasie-eksperimente vaar die LIOA roeteringstrategie die beste en die MHA roeteringstrategie die slegste. Daar word verder waargeneem dat die komputasiekomplesiteit van die MIRA roeteringstrategie nie noodwendig weerspieël word in die sukses van roeteringsuitkoms nie.

In die geval waar die aanlyn roeteringstrategieë aangewend word vir MPLS foutrestorasie, toon die resultate van simulasie-eksperimente dat al die roeteringstrategieë min of meer dieselfde uitkoms lewer ten opsigte van herroetering van onderbreekte verkeersvloei. Die konfigurasie van etiketgeskakelde paaie deur die LIOA roeteringstrategie voor skakelonderbreking is egter die geskikste vir televerkeer herroetering na skakelonderbreking.

Acknowledgements

This work was performed within the *Siemens-Telkom Centre of Excellence for ATM & Broadband Networks and their Applications* and is supported by grants from the South African National Research Foundation, Telkom SA Limited, Siemens Telecommunications and the University of Stellenbosch.

I hereby acknowledge the financial assistance of the South African Department of Labour (DoL) towards this research. Opinions expressed and conclusions arrived at are my own and not necessarily to be attributed to the DoL.

I would also like to thank the International Office at the University of Stellenbosch and the Stockholm Royal Institute of Technology (KTH) for their support during the exchange period I have spent at KTH.

Special thanks to my supervisor, Prof. A. E. Krzesinksi, for his supervision and involvement in this thesis and to Antoine Bagula for his constant motivation and research contributions.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The Internet is fast evolving into a commercial platform that carries a mixture of narrow- and broadband applications such as voice, video and data. This evolution together with the rapid increase of Internet users presents several challenges to Internet Service Providers (ISPs). Various newly evolved network applications require services that are deterministic in nature while users expect a certain level of guaranteed service from their service providers. Consequently the need exists for efficient Internet traffic engineering to enable better Quality of Service (QoS) capabilities.

Neither the current IPv4 protocols deployed in the Internet nor the emerging IPv6 protocols provide sufficient capabilities for traffic engineering to enable the necessary QoS control required by the network users and applications. Current generation IPv4 protocols were developed on the basis of a connectionless model where routing decisions are based on simple performance metrics such as hop-count and transmission delay. Despite its ability to scale to very large networks, IPv4 only provides support for rudimentary levels of QoS control.

The emerging IPv6 protocols extend current IPv4 protocols with flow label fields and extended headers and thus provide carrier networks with an improved priority system. However, these improvements still do not meet the required QoS levels for many real-time or bandwidth intensive applications. The success of the Internet will thus greatly depend on the design of new QoS driven routing algorithms or the efficient extension of currently employed routing protocols to achieve better QoS levels in communication networks.

1.1 Multi-protocol Label Switching

Multi-protocol Label Switching (MPLS) is a label switching protocol that has emerged as a potential enabling technology to achieve traffic engineering and QoS management in IP networks. It was introduced by the Internet Engineering Task Force (IETF) and has since been implemented by vendors such as Cisco, Toshiba, Ipsilon and IBM [1, 2, 3, 4]. As the name indicates, MPLS was designed to be applicable to any network layer protocol, but in this thesis we will focus on the use of IP as the network layer protocol.

1.1.1 MPLS Architecture

The MPLS architecture is specified in [5, 10, 38]. In this section we give a brief overview of the key concepts of MPLS as applicable to this thesis. The importance of traffic engineering and the suitability of MPLS for traffic engineering will be discussed in the next section.

An MPLS backbone network consists of a number of routers known as *label switching routers (LSRs)*. The LSRs are connected by physical links. An LSR is called an ingress or an egress router depending on whether it is handling traffic that respectively enters or leaves the MPLS network domain. The routers that connect the MPLS backbone to other network domains are known as *edge label switching routers (ELRs)*.

A primary component of MPLS is the Label Switching Mechanism. In traditional IP networks IP header analysis is performed at every hop of the packet's path. In the MPLS forwarding paradigm the IP header analysis is only performed at the ingress LSR where the IP packet enters the MPLS domain.

During the header analysis phase all packets are grouped into *Forwarding Equivalence Classes (FECs)* based on parameters such as the address prefix, the host address and QoS requirements. The FEC to which a packet belongs is encoded as a short fixed length value known as a *label* which is sent with the packet when the packet is forwarded to its next hop. At each subsequent hop the label is used as an index into a routing table that specifies the next hop in the packet's path and a new label. Depending on the label operation specified by the current label of the packet, the old label can be replaced by the new label (*label swapping*), a new label can be added to the packet's label stack (*label pushing*) or the current label can be removed from the packet's label stack (*label popping*) before the packet is forwarded to the next hop.

All packets belonging to the same FEC follow the same path through the MPLS network. This path is known as a *Label Switched Path (LSP)*. LSPs are assigned virtual capacity and MPLS has LSP management capabilities in order to add or remove LSPs and to assign and reassign virtual capacity to the LSPs as required. MPLS also provides mechanisms to perform LSP overload protection by means of admission control and packet policing.

Path selection refers to the process of determining an LSP for a particular FEC. Currently the MPLS architecture supports two methods for path selection: *hop by hop routing* and *explicit routing*. Hop by hop routing allows each LSR to independently choose the next hop to which the packets of a particular FEC will be forwarded. This is comparable to the forwarding mode currently used in IP networks. In an explicitly routed LSP each LSR does not independently choose the next hop, but the entire LSP is specified by a single LSR, usually the ingress LSR. Explicit routing is useful for purposes such as policy routing and traffic engineering. The significance of explicit routing for traffic engineering purposes will be discussed in the next section.

To establish LSPs, various *signaling protocols* are used to distribute label information and to update the *Label Information Base (LIB)* located at every LSR. The MPLS architecture does not impose a single signaling protocol and currently a number of different label distribution protocols are being standardized. These include extensions to existing protocols such as the Border Gateway Protocol (BGP) and the ReSerVation Protocol (RSVP)[6] and also newly proposed protocols such as the Label Distribution Protocol (LDP) [23, 42]. Some of these signaling protocols have also been extended to provide the capability for the explicit routing of LSPs for traffic engineering purposes (see for example RSVP-TE [16, 18] and CR-LDP [21]).

Although a relatively new protocol, MPLS is likely to be a technology of choice in future IP-based transport networks. Some of the benefits of MPLS include its capability for multi-protocol support and the fact that it is intended to support any type of link layer medium for example ATM, Frame Relay, Ethernet, FDDI and Token Ring. Due to its simplified packet forwarding and switching mechanisms and the possibility to create hardware implementations for the look-up and forwarding operations, MPLS enables improved routing flexibility.

1.3 Thesis Outline

The thesis is structured as follows:

Chapter 2 discusses the online routing optimization problem and its applicability to traffic engineered LSPs in MPLS networks. The limitations inherent to online routing are addressed and a brief overview of some of the most popular online routing schemes currently deployed in communication networks is given. Furthermore, recently proposed online routing algorithms for online traffic engineering of LSPs are discussed and special attention is given to the *Minimum Interference Routing Algorithm (MIRA)* and its extensions [31].

Chapter 3 proposes a new online routing framework known as Least Interference Optimization (LIO) that utilizes the current bandwidth availability and traffic flow distribution to achieve efficient traffic engineering. We present the Least Interference Optimization Algorithm (LIOA) that reduces the interference among competing network flows by balancing the number and quantity of flows carried by a link for the setup of bandwidth guaranteed LSPs in MPLS networks.

In Chapter 4 the performance of the LIOA routing strategy is evaluated and compared against other popular online routing strategies by means of simulation. The performance measures under consideration include LSP request rejection, bandwidth utilization, flow distribution and path quality.

Chapter 5 addresses MPLS failure restoration and the possibility of utilizing the online routing strategies for LSP rerouting upon link failure. Two aspects of the online routing strategies are investigated: the suitability of the LSP configuration that results due to the establishment of LSPs prior to link failure and the ability of the online routing strategy to reroute failed LSPs upon link failure.

Chapter 2

An Overview of Online Traffic Engineering

This chapter gives a brief introduction to online routing and explains the limitations inherent to online routing algorithms. We state the requirements for online traffic engineering in MPLS networks and we introduce a mathematical model for the online routing problem addressed in this thesis. Section 2.2 gives a brief overview of well-known online routing strategies as well as a detailed discussion of online routing algorithms that will be evaluated in this thesis. In section 2.3 we present an online routing approach that combines concepts of both online routing and offline optimization as a solution to the online routing problem.

2.1 The Online Routing Problem

Online routing is an allocation process that aims to assign bandwidth and other network resources to network applications in an on demand manner. Online routing algorithms are essentially *myopic*: routing decisions are topologically driven and based on the current network state without taking the long term implications on the network performance into account. Although widely deployed, online routing algorithms are often criticized for their myopic nature and their tendency to cause network congestion [15, 41]. Since we aim to utilize online routing methods for the explicit setup of LSPs, we start our discussion by explaining the limitations inherent to online routing methods.

To explain these limitations we consider the network topology presented in figure 2.1. The network consists of 11 nodes and 12 links with fixed link capacities as indicated in figure 2.1(a). Assume that requests to establish bandwidth guaranteed LSPs arrive independently

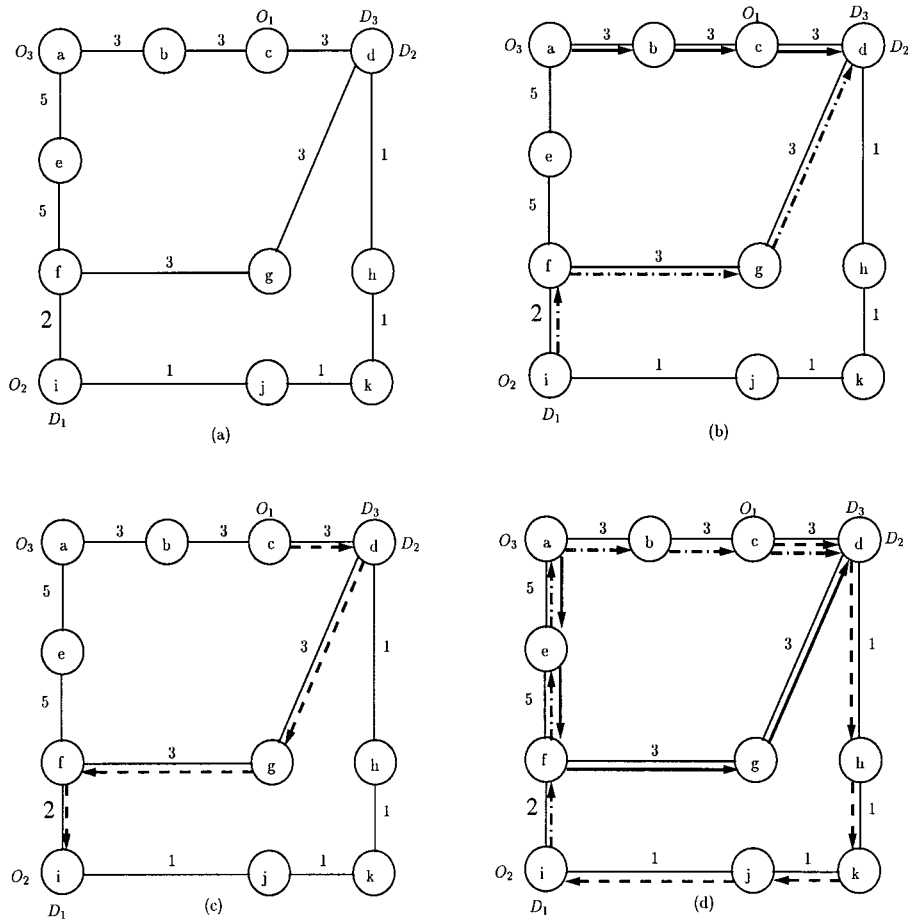

Figure 2.1: The Online Routing Limitations

to the network and are accepted and routed in order of arrival only if there is enough available bandwidth to carry the required demand. Each LSP setup request is denoted by a triple (O, D, d) where O denotes the ingress node, D denotes the egress node and d denotes the bandwidth demand to be routed between O and D. Since most routing algorithms currently deployed in communication networks are based on a *shortest path first* principle, we use the classical Minimum Hop Algorithm (MHA) algorithm [8] to select a route along which an LSP will be set up. Hence, every demand will always be routed along the shortest available path in terms of hop count. We assume no prior knowledge of the arrival sequence or demand sizes of future LSP setup requests.

Consider the arrival of three LSP setup requests in the following order: $(O_3, D_3, 3)$, $(O_2, D_2, 2)$, $(O_1, D_1, 1)$. If we select the minimum hop route, the first LSP will be set up along the route $a-b-c-d$ and the second LSP will be set up along the route $i-f-g-d$. This configuration

will make it impossible to accept the last request due to insufficient bandwidth availability. Hence the request $(O_1, D_1, 1)$ will be rejected. This situation is depicted in figure 2.1(b). The same situation will occur if request $(O_2, D_2, 2)$ arrives before request $(O_3, D_3, 3)$ – the MHA algorithm will accept at most two LSP requests.

Now consider the case where $(O_1, D_1, 1)$ arrives first. By choosing the minimum hop path, the LSP for $(O_1, D_1, 1)$ will be set up along route $c - d - g - f - i$ making it impossible to accept either request $(O_2, D_2, 2)$ or $(O_3, D_3, 3)$. The situation is depicted in figure 2.1(c).

From the above discussion it is possible to make two observations:

- The sequence of the arriving LSP requests plays an important role in the overall performance of the routing algorithm.
- Since each arriving LSP request is oblivious to future requests, each LSP is set up in a myopic manner and an "optimal" routing decision is made by only taking the current available network state information into account. This strategy might not necessarily result in an optimal LSP configuration.

The second observation is stressed by the fact that there does indeed exist an optimal LSP configuration that allows all three LSP setup requests to be accepted. The optimal solution is depicted in figure 2.1(d) and is obtained if the LSP for the request $(O_1, D_1, 1)$ is set up along route $c - d - h - k - j - i$, the LSP for request $(O_2, D_2, 2)$ is set up along route $i - f - e - a - b - c - d$ and the LSP for request $(O_3, D_3, 3)$ is set up along route $a - e - f - g - d$. Note however that none of these LSPs are set up along the shortest routes between their corresponding ingress-egress pairs. In fact, if any of the requested LSPs are set up along the shortest routes between their respective ingress-egress pairs, it will be impossible to obtain the optimal solution. This proves that making an "optimal" decision for each LSP request does not necessarily lead to the optimal LSP configuration for the entire network.

So if making an "optimal" decision (in this case the minimum hop path determined by the MHA algorithm) for each LSP request does not result in an optimal network configuration, what can be done in order to obtain an optimal or near optimal network configuration? Three possible solutions include:

1. Making use of an offline optimization method to calculate an optimal LSP configuration based on prior knowledge of LSP setup requests and bandwidth demand sizes.

2. Splitting the bandwidth demand in smaller demand portions in an attempt to route the single bandwidth demand over multiple LSPs.

3. Adjusting the routing objective of the online routing algorithm in such a way that it not only considers *static* network information (such as topological information) but also exploits *dynamic* network information (for example residual link capacities) in an attempt to be more responsive to the current network state.

The first solution requires prior knowledge of all LSP setup requests and bandwidth demand sizes. In the event where this information is available in advance, the most effective traffic engineering can be achieved by utilizing an offline optimization method. However, in practice the accurate awareness of future LSP demands is unlikely and the static behaviour of offline traffic engineering is ineffective in cases where existing LSP demands change continually or where new LSP demands are frequently added to an already configured network. In this scenario the dynamic adaptability of online routing algorithms and the fact that they do not depend on prior knowledge of traffic demands provide a more flexible routing solution.

For the network example illustrated in 2.1, the second solution namely traffic splitting does provide a mechanism to find an optimal network configuration. Consider for example the situation depicted in figure 2.1(c) where the demand $(O_1, D_1, 1)$ arrives first. If we split the demands $(O_2, D_2, 2)$ and $(O_3, D_3, 3)$ it is possible to accept all three LSP requests by routing 1 unit of demand $(O_2, D_2, 2)$ along route $i - j - k - h - d$ and 1 unit along route $i - f - g - d$ while routing 2 units of demand $(O_3, D_3, 3)$ along route $a - b - c - d$ and 1 unit along route $a - e - f - g - d$.

In practice, traffic splitting is often used for load balancing purposes, but it might not be possible for the routing algorithm to split demands in an arbitrary manner since the traffic being routed might be inherently unsplittable. Hence, even though traffic splitting might enable better resource usage, it is desirable for a routing algorithm to be able to route a fixed demand of bandwidth between an ingress-egress pair in an optimal way without relying on the possibility of traffic splitting.

It is therefore useful to experiment with different online routing strategies and to evaluate the effect that different routing objectives have on the prevention of network congestion and the optimality of the resulting LSP configuration.

2.1.1 Requirements for MPLS Routing Algorithms

To be useful in practice, an online LSP selection algorithm for MPLS network should satisfy certain performance and implemental requirements. These requirements have been identified in [31]:

1. *Feasibility of distributed implementation*: Although a central route server could be used for LSP computations, it is generally preferred that the online algorithm is suited to a distributed implementation where each LSP's explicit route can be computed locally at the ingress router. Furthermore it is desirable for the algorithm to restrict its use of dynamic information to information derivable from current routing protocols.

2. *Computational requirements*: The success of online routing algorithms relies not only on their ability to perform effective resource assignment, but also on their computational efficiency. It must be possible to implement the routing algorithm on LSRs and route servers and the algorithm must execute within a reasonable time frame to be useful in practice.

3. *Re-optimization:* Although frequent rerouting is undesirable, the algorithm must be capable of rerouting existing LSPs in the event of occasional network re-optimization. This optimization process cannot be done efficiently using an offline method since there is no knowledge of future LSP setup arrivals.

4. *Good rerouting capability upon failure*: The ability of the routing algorithm to successfully reroute failed LSPs after a network component failure is regarded as an important performance metric. A robust routing algorithm should therefore take the possibility of network failure into account when making routing decisions and also provide rerouting capabilities for failed LSPs.

5. *Policy constraints:* The algorithm must be able to incorporate common policy constraints such as policy restrictions on the type of links or routers that are permissible for routing a LSP.

2.1.2 Mathematical Formulation of the Online Routing Problem

We now present a mathematical model based on optimization principles for the specific online routing problem that will be addressed in this thesis.

Consider a network represented by a directed graph $G(\mathcal{N}, \mathcal{L})$ where $\mathcal{N}$ represents the set of nodes and $\mathcal{L}$ represents the set of links. Let c_ℓ denote the capacity of link $\ell \in \mathcal{L}$ and let $\mathcal{P}_{i,e}$ denote the set of feasible paths (LSPs) connecting ingress-egress pair (i, e). Assume that a request to establish an LSP of $d_{i,e}$ bandwidth units between ingress-egress pair (i, e) arrives to the network and that future LSP demands are unknown.

The routing problem consists of finding the best feasible path $p^* \in \mathcal{P}_{i,e}$ along which to set up the LSP such that

$$0 \leq r_\ell < c_\ell \tag{2.1}$$

$$d_{i,e} \leq \min_{\ell \in p^*}(c_\ell - r_\ell) \tag{2.2}$$

$$L_p = \sum_{\ell \in p} L_\ell \tag{2.3}$$

$$L_{p^*} = \min_{p \in \mathcal{P}_{i,e}} L_p \tag{2.4}$$

where r_ℓ is the total bandwidth reserved by all LSPs traversing link ℓ, L_ℓ is the cost of link ℓ and L_p is the cost of path p.

Equation (2.1) expresses the feasibility of the total traffic flow carried by a link – the flow on a link must be less than the capacity of the link. Equation (2.2) expresses the feasibility of the path and states that each link of the feasible path p^* must have enough unreserved bandwidth to carry the demand $d_{i,e}$. Equation (2.3) defines the additive path cost function while equation (2.4) expresses the optimality of the routing process.

Equation (2.3) deserves closer examination since it is essentially this formulation that determines the behaviour of the routing process. The path cost function L_p is defined as a linear additive function that consists of the individual link costs (or weights) L_ℓ of each link on the path. The distinct behaviour of different routing strategies will therefore be determined by the routing objective expressed by the link cost L_ℓ assigned to each link.

This formulation of the routing problem lends itself to a simple and efficient solution. The problem of finding the optimal path can be reduced to finding the minimum cost path that can easily be solved with classical graph algorithms such as Dijkstra's Algorithm or the Bellman-Ford Algorithm [8], using L_ℓ as the weight on link ℓ. The predominant complexity of solving the routing problem will be determined by the complexity of calculating the link costs.

2.2 A Survey of Online Routing Algorithms

2.2.1 Shortest Path First Algorithms

Shortest Path First (SPF) algorithms refer to a group of path selection algorithms with the objective of selecting the "shortest" path, not necessarily based on physical distance or link count, but according to weights assigned to each of the links of the path.

SPF algorithms are undoubtedly the most popular routing algorithms in current networks and are widely deployed routing protocols. This can mainly be attributed to their simplicity, ease of implementation and computational efficiency. They exist in many forms, route according to various objectives and differ mainly by their link cost functions (or link weights). We give an overview of the most popular SPF routing algorithms, paying special attention to the algorithms that will be used in this thesis. In particular, the SPF algorithms considered in this thesis will be the Minimum Hop Algorithm (MHA), the Open Shortest Path First algorithm (OSPF) and the Constraint Shortest Path First algorithm (CSPF). For each of these algorithms we specify the respective link cost function.

The Minimum Hop Algorithm

The objective of the *Minimum Hop Algorithm (MHA)* is to select shortest paths based on hop count (node count). Although popular due to its simplicity, the MHA algorithm often performs poorly due to its ignorance of resource availability (for example bandwidth distribution in the network topology) so that all routing decisions are based on static topological information. The routing objective is reflected by link cost function 2.1:

Link Cost Function 2.1 (MHA) *The MHA link cost function for link ℓ is given by*

$$L_\ell = 1.$$

The Open Shortest Path First Algorithm

Open Shortest Path First (OSPF) [37] is currently the most commonly used intra-domain Internet routing protocol. Link weights can be set proportional to their physical distances, but a standard default heuristic, proposed by Cisco [1], is to set the cost of every link inversely proportional to its capacity in an attempt to relieve network congestion. Similar to the MHA algorithm, the static nature of the cost function may lead to unbalanced load distribution by

assigning bandwidth requests to overloaded parts of the network while leaving other parts of the network under-utilized. Successful efforts have been made to solve the load balancing problem of OSPF in an attempt to make it a viable option for traffic engineering (see for example [28, 33]). In this thesis we will consider the default OSPF link cost defined in link cost function 2.2:

Link Cost Function 2.2 (OSPF) *The OSPF link cost function for link ℓ is given by*

$$L_\ell = 1/c_\ell$$

where c_ℓ is the total capacity on link ℓ.

Constraint Shortest Path Routing

Constraint Shortest Path First (CSPF) [9] was proposed as a solution to the load balancing problem inherent to OSPF by using link costs that reflect the current resource availability in the network. These include costs metrics that aim to utilize dynamic network state information such as path delay and residual link capacities, and path computations are often done on a pruned network topology. We will consider the following link cost setting in this thesis:

Link Cost Function 2.3 (CSPF) *The CSPF link cost function for link ℓ is given by*

$$L_\ell = \frac{1}{c_\ell - r_\ell}$$

where c_ℓ is the total capacity and r_ℓ is the total reserved capacity on link ℓ.

Note however that the use of dynamic network state information incurs a higher routing cost since the state of all dynamic cost metrics must be propagated through the network at frequent intervals in order to update the information stored in routers and route servers.

Other Popular Shortest Path First Algorithms

Although not used in this thesis, we also mention three other popular Shortest Path First strategies since they are comparable with the work done in this thesis and often used as benchmark algorithms for the performance evaluation of competing algorithms.

The *Widest Shortest Path (WSP)* algorithm was proposed by [29] for primary path routing of bandwidth guaranteed paths. WSP is similar to the MHA algorithm, but achieves

better load balancing by selecting the shortest path in terms of hop count with maximum residual capacity on the links of the path. The *Shortest Widest Path (SWP)* algorithm [45] is essentially the same as WSP, but it first selects a path with maximum residual capacity and if there exists more than one such path, the shorter is preferred. The *K Shortest Path (KSP)* algorithm [12] modifies the WSP algorithm by only considering K path options in the routing decision in an attempt to reduce computational complexity.

2.2.2 Minimum Interference Routing

The concept of minimum interference routing was introduced in [31] and an online routing algorithm known as the *Minimum Interference Routing Algorithm (MIRA)* was proposed to address the problem of setting up explicit bandwidth guaranteed LSPs in MPLS networks. The algorithm assumes that that LSP set-up requests arrive one by one to the network and that there is no prior knowledge regarding future setup requests. By exploiting the knowledge of the residual link capacities and the location of all possible ingress-egress pairs, the number of LSP requests that are rejected due to insufficient network capacity is reduced.

The main idea behind minimum interference routing is to set up each LSP in such a way that it does not interfere too much with potential future LSP set-up requests between other ingress-egress pairs. The amount of interference for ingress-egress pair (i, e) due to the establishment an LSP between another ingress-egress pair is defined as the decrease in the maximum flow value (maxflow)[1] between (i, e). The minimum interference LSP for an ingress-egress pair will thus be the explicit route that maximizes the minimum maxflow value between all other ingress-egress pairs.

The concept of maximum flow can be explained as follows. Consider the network topology consisting of 4 nodes and 5 links with fixed capacities as illustrated in figure 2.2(a). The maximum amount of flow that can be routed between ingress-egress pair (O_1, D_1) is 8 bandwidth units – 3 units along route $a - b - d$, 3 units along route $a - c - d$ and 2 units along route $a - c - b - d$. Now consider the situation illustrated in figure 2.2(b) where a flow of 2 bandwidth units is routed between ingress-egress pair (O_2, D_2) along route $b - d - c$. The maximum flow value that can now be routed between ingress-egress pair (O_1, D_1) is 4 bandwidth units – 1 unit along route $a - c - d$ and 3 units along route $a - b - d$.

[1] The maxflow problem is a well known problem in the context of flow networks and has widely been covered in the literature. See for example [8] for a detailed discussion on maxflow computations and algorithms.

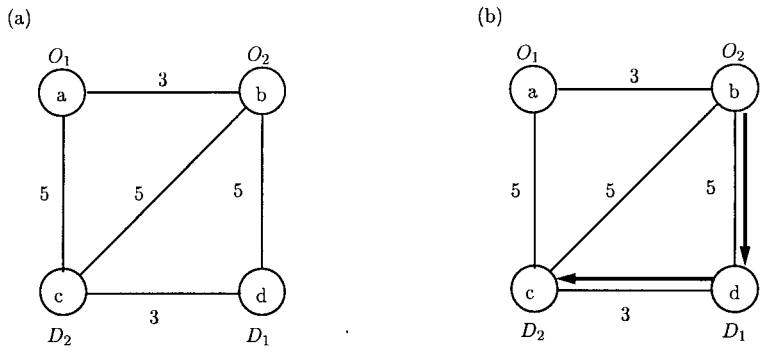

Figure 2.2: The Maxflow Problem

The minimum interference routing problem was formulated as an integer programming problem, but since solving this problem is NP-hard, the problem is solved by a heuristic approach that assigns appropriate weights to all the links and then utilizes a shortest path algorithm to compute a minimum interference LSP. We will not discuss the derivation of the link weights (see [31, 35]) but describe the implementation of the Minimum Interference Routing Algorithm as applicable to this thesis.

Critical Link Set

In order to compute the link weight for every link, the notion of a *critical link* is used. Critical links are links that can prevent the setup of future LSPs between certain ingress-egress pairs if they are heavily loaded.

A link is defined to be critical for a given ingress-egress pair if the link belongs to any minimum-cut set of the ingress-egress pair. The minimum-cut sets are computed using the current residual link capacities. Furthermore it is shown in [31] that the complete set of critical links $\mathcal{C}_{i,e}$ for ingress-egress pair (i, e) can be derived from the resulting flow residual graph after execution of a maxflow algorithm between ingress-egress pair (i, e) (see [8] for more information regarding flow residual graphs).

Let i denote the ingress node, let e denote the egress node and assume that the maximum flow between i and e has been computed. Let $\mathcal{T}$ represent the nodes that can be reached from i and let $\mathcal{S}$ denote the set of nodes that can reach e in the resulting flow residual graph. Then link $(u, v) \in \mathcal{C}_{i,e}$ if

- link (u, v) is filled to capacity,

- $v \notin \mathcal{S}$ and $u \notin \mathcal{T}$,
- there is no path between u and v in the flow residual graph.

After determining the critical link sets for all ingress-egress pairs, it is possible to determine the link weights which are defined in [31] to be

$$L_\ell = \sum_{(i,e):\ell \in \mathcal{C}_{i,e}} \alpha_{ie}$$

where $\alpha_{i,e}$ is a value expressing the importance of ingress-egress pair (i, e).

The Minimum Interference Routing Algorithm

In [31] an algorithm is presented for the establishment of bandwidth guaranteed LSPs in MPLS networks. For every LSP setup request, the link weights are computed and the minimum interference route is calculated by executing a shortest path first algorithm. The MIRA algorithm is given below.

MINIMUM INTERFERENCE ROUTING ALGORITHM (MIRA)

Input:
A Graph $G(\mathcal{N}, \mathcal{L})$, the set of ingress-egress pairs $\mathcal{I}$ and an ingress-egress pair (i, e) between which a flow of $d_{i,e}$ units has to be routed.

Output:
A route between i and e having a capacity of $d_{i,e}$ units of bandwidth.

Algorithm:

1. Compute the maximum flow values for all ingress-egress pairs $(a, b) \in \mathcal{I} \backslash (i, e)$.
2. Compute the set of critical links $\mathcal{C}_{a,b}$ for all $(a, b) \in \mathcal{I} \backslash (i, e)$.
3. Compute the weights $L_\ell := \sum_{(a,b):\ell \in \mathcal{C}_{a,b}} \alpha_{ab}$ for all $\ell \in \mathcal{L}$.
4. Eliminate all the links that have less than $d_{i,e}$ units of residual bandwidth to form a reduced network.

5. Using Dijkstra's algorithm, compute the shortest path in the reduced network using L_ℓ as the weight on link ℓ.

6. Route the demand of $d_{i,e}$ units between i and e along this shortest path and update the residual link capacities.

With regards to the implementation of the MIRA algorithm, we make the following remarks:

- If k is the number of ingress-egress pairs, then determining all the critical link sets has a computational complexity of $O(k)$ in addition to the complexity of the maxflow computation. Therefore the execution time of the MIRA algorithm is determined by the execution time of the maxflow computation.
- Several methods for efficient computation of the maxflow values are discussed in [35]. The most popular maxflow algorithm is the Ford-Fulkerson algorithm, while the Goldberg-Tarjan highest level push algorithm is regarded as one of the fastest maxflow algorithms [8] with a complexity of $O(n^2\sqrt{m})$ where n represents the number of nodes and m represents the number of links.
- The α value in step 3 of the MIRA algorithm determines the behavior of the shortest path algorithm and can be set to any of the values proposed in [35]. For our implementation we chose the $\alpha = 1$, thus giving equal importance to all ingress-egress pairs so that L_ℓ expresses the number of ingress-egress pairs for which link ℓ is critical.

We now define the link cost function that will be used for the MIRA algorithm throughout this thesis:

Link Cost Function 2.4 (MIRA) *The MIRA link cost function for link ℓ is given by*

$$L_\ell = \sum_{(i,e):\ell \in \mathcal{C}_{i,e}} 1$$

where $\mathcal{C}_{i,e}$ is the critical link set of ingress-egress pair (i,e).

Performance Improvements

Although recently proposed, the MIRA algorithm has received a lot of attention and has been widely criticized for its high computational complexity (see for example [11, 41, 46] and [47]).

In light of the criticism, improvements were made to both the computational complexity and the overall performance of the MIRA algorithm. These improvements include the following (see [31] and [32] for detailed discussions):

- *Lexicographic MIRA*: The original MIRA algorithm aims to find a route that maximizes the minimum maxflow value between all ingress-egress pairs. By fine-tuning this process it is possible to improve the overall performance of the MIRA algorithm. The main objective remains to find the solution that maximizes the minimum maxflow value, but among all these solutions with the same minimum maxflow value, a secondary objective is to choose the solution that maximizes the second highest maxflow value and so on.
- *Computation Frequency of Critical Links*: In an attempt to lower the high computational complexity, the frequency of computing the critical link sets has been investigated. Instead of calculating the critical link sets for every newly arrived LSP request, the critical link computation is only done after the arrival of a fixed number of LSP requests and all link weights are derived from these critical link sets for a fixed number of subsequent LSP request arrivals. Experimentation proved that by choosing an effective computation frequency, this method substantially decreases the computational complexity of the algorithm without severe performance degradation.
- Δ-*Critical Links*: A second enhancement made to the overall performance of the MIRA algorithm involves a more accurate computation of the critical link sets. This enhancement not only computes the critical links, but also identifies links that are near to being critical. The Δ-critical links computation is especially useful in the case where critical link sets are not computed after every LSP setup request arrival but rather at more infrequent intervals.

2.2.3 Related Work

Since the introduction of MPLS as an enabling technology for improved QoS management, much attention has been given to the path selection process and the setup of explicit bandwidth guaranteed LSPs. We already discussed the MIRA algorithm, and although not examined in this thesis, we mention other work closely related to online traffic engineering for the establishment of bandwidth guaranteed LSPs.

- In [43] a new online algorithm for dynamic routing of bandwidth guaranteed LSPs is presented. The algorithm assumes no prior knowledge regarding future LSP setup

requests or their characteristics. The algorithm is closely related to the MIRA algorithm and also considers the notion of critical links, but unlike the MIRA algorithm the new algorithm assigns an importance factor to all network links by characterizing their bandwidth contribution for routing future LSP demands. The performance of the new algorithm was tested against the performance of routing algorithms such as MIRA, WSP, SWP and MHA. Simulation results show that the newly proposed algorithm achieves a lower LSP rejection rate compared to the competing algorithms, but since the routing process involves the calculation of critical links, this algorithm has a prohibitively large computational complexity similar to the MIRA algorithm.

- The *Stochastic Performance Comparison Routing Algorithm (SPeCRA)* was proposed in [11]. SPeCRA also requires no prior information regarding future LSP setup requests and their stochastic nature. The SPeCRA algorithm considers a set of possible routing schemes (such as SPF routing strategies) and then periodically selects an optimal routing algorithm from the algorithm set based on LSP rejection probabilities. The SPeCRA strategy was compared with the MIRA and the KSP algorithm and outperformed these algorithms with regard to the LSP rejection ratio. The computational complexity of the SPeCRA algorithm will mainly depend on the algorithms chosen for the set of routing schemes.

- In [47] an attempt is made to compare the scalability and performance tradeoffs in MPLS and IP routing by comparing some of the recently proposed algorithms for the selection of bandwidth guaranteed LSPs with traditional IP routing strategies. The algorithms under consideration include the online algorithms WSP, MIRA and a dynamic per-packet IP routing strategy as well as a routing strategy known as Profile Based Routing[2]. Results show that the per-packet routing strategy outperforms the other algorithms on performance measures such as network throughput and resource utilization, while the PBR routing approach performs poorly compared to the competing algorithms.

2.3 Advanced Online Routing

It is clear from the discussion in section 2.1 that efficient traffic engineering can be enforced by using either online or offline strategies, but both strategies come with certain disadvantages. While offline optimization strategies can compute an optimal or near optimal network configuration, they have the disadvantage that their LSP selection remains static after net-

[2]This routing strategy will be discussed in section 2.3

work configuration and they might not be useful in practice because they strongly rely on the accuracy of their input data, most importantly on an accurate traffic profile. The dynamic and robust nature of online routing add to the attractiveness of online routing strategies but, due to their myopic nature, online routing strategies very rarely achieve the same optimality as their offline rivals.

Recently a group of advanced online routing strategies has been proposed by the network community. This advanced online routing approach aims to enhance traffic engineering capabilities by combining the most attractive concepts of both online routing strategies and offline optimization methods. We briefly discuss two recently proposed advanced routing strategies.

The first approach is referred to as *Profile Based Routing (PBR)* [41]. The PBR algorithm uses quasi-static information about a traffic profile, measured or inferred from service level agreements, in a preprocessing step and determines fixed bandwidth allocations on the network links for each of the ingress-egress pairs. The online phase of the routing algorithm then routes LSP setup requests using a shortest path first approach but with the additional information given by the preprocessing phase. The preprocessing phase involves solving the well-known multi-commodity network flow problem and the output is used to guide the online routing process as well as to impose admission control.

The second approach is known as *Design Based Routing (DBR)* [46]. The idea behind DBR is similar to the idea exploited by PBR: the DBR algorithm also uses a traffic profile to compute an optimal LSP configuration offline, but instead of allocating portions of bandwidth on the network links to the ingress-egress pairs, the pre-computed LSPs are stored in a central database and deployed on request from an ingress node that receives a LSP setup request. DBR obtains a traffic profile form customer prescriptions, traffic projections and historical measurements, but does not rely much on the accuracy of the traffic profile. The traffic profile is only used in the initial computations of optimal paths, but the centralized database server can re-compute and re-optimize these paths periodically as information regarding the current state of the network becomes available.

Chapter 3

Least Interference Optimization

This chapter introduces a new online routing framework for the establishment of bandwidth guaranteed LSPs in MPLS networks. The *Least Interference Optimization (LIO)* framework is discussed in section 3.1 and the *Least Interference Optimization Algorithm (LIOA)* is presented in section 3.2.

3.1 Least Interference Optimization Routing

Route optimization is a core concept upon which the efficiency of traffic engineering schemes depends. In the previous chapter we discussed some of the popular approaches to online traffic engineering and we presented the link cost functions for the MHA, OSPF, CSPF and MIRA routing strategies (refer to the link cost functions 2.1–2.4). However due to the myopic nature of these algorithms, and online routing algorithms in general, it is difficult to achieve an optimal path configuration for efficient traffic engineering.

Intuitively one expects CSPF and MIRA to achieve better resource utilization than MHA and OSPF since the former employ dynamic network state information and thus, unlike MHA and OSPF, base their routing decisions on information that reflects the current resource availability of the network. The MIRA routing strategy also introduces the concept of *interference* and, although still myopic, attempts to take the requirements of potential future LSP demands into account before bandwidth allocation. Unfortunately this strategy incurs additional complexity that does not necessarily translate into equivalent performance gains.

Similar to the MIRA routing strategy, we propose a *Least Interference Optimization (LIO)*

routing strategy for the establishment of bandwidth guaranteed LSPs in MPLS networks that aims to minimize the interference between competing network flows to achieve efficient route optimization. The LIO routing strategy is essentially a minimum cost path routing strategy that, like the online routing strategies previously discussed[1], computes a minimum cost path using a link cost (weight) assigned to each of the network links.

3.1.1 A Multiplicative Penalty Function

Link costs are calculated with a multiplicative link penalty function defined as follows:

Link Cost Function 3.1 (LIOA(a)) *The multiplicative LIO link cost function for link ℓ with calibration parameter $0 \leq \alpha \leq 1$ is given by*

$$L_\ell = \frac{I_\ell^\alpha}{(c_\ell - r_\ell)^{1-\alpha}}$$

where $I_\ell > 0$ is the value of the interference parameter[2], c_ℓ is the total capacity and r_ℓ is the total reserved capacity of link ℓ.

The link cost function consists of two dynamic link cost metrics and is proportional to the product of the link interference I_ℓ and the inverse of the residual link capacity, balanced by a calibration parameter α. We will discuss the effect of each of these parameters in turn.

3.1.2 The Interference Parameter: I_ℓ

The interference parameter I_ℓ is a measure that indicates the extent to which the selection of link ℓ for the set up of an LSP might interfere with the setup of future LSPs. The LIO framework does not restrict the choice of interference parameter, and we will consider two possibilities for the calculation of I_ℓ in this thesis.

The MIRA routing strategy defines the interference for an ingress-egress pair (i, e) to be the decrease in the maximum flow value between (i, e) due to the set up of a LSP between

[1] Refer to the mathematical formulation of the online routing problem and the online routing strategies discussed in chapter 2.

[2] Note that if $I_\ell = 0$, L_ℓ is undefined.

another ingress-egress pair. It might therefore be possible to set the interference parameter I_ℓ equal to the link interference value calculated by the MIRA routing strategy. Thus

$$I_\ell = \sum_{(i,e):\ell \in \mathcal{C}_{i,e}} 1 \tag{3.1}$$

where $\mathcal{C}_{i,e}$ is the critical link set[3] for ingress-egress pair (i, e). Unfortunately calculating the interference value as defined in the MIRA strategy has a prohibitively high computational cost and complexity.

As an alternative we propose an interference measure that estimates the popularity of a link by setting the interference parameter for link ℓ equal to the number of traffic flows (or the number of LSPs) carried by link ℓ. Let $\mathcal{A}_\ell$ denote the set of LSPs that traverses link ℓ. Then

$$I_\ell = \mid \mathcal{A}_\ell \mid . \tag{3.2}$$

Although not as predictive in nature as the interference measure proposed by the MIRA routing strategy, setting I_ℓ equal to the number of traffic flows on link ℓ has the advantage of simplicity and incurs almost no additional computational cost.

Note the restriction placed on I_ℓ by the link cost function 3.1: $I_\ell > 0$. This is to insure that the link cost $L_\ell > 0$ since a link cost $L_\ell = 0$ will lead to inconsistent path selection. The nature of the interference parameter should therefore be such that $I_\ell > 0$, or I_ℓ should be initialized to a suitable value on every link ℓ before computation of the link costs.

Note too that the interference measure proposed by the MIRA routing strategy has certain limitations that might lead to performance degradation. To illustrate this and to evaluate the effectiveness of the two interference measures proposed in equations 3.1 and 3.2, we adopt a routing strategy referred to as *Least Interference(LI)* routing that bases all routing decisions on the link interference. This is equivalent to setting $\alpha = 1$ in link cost function 3.1, resulting in the link cost $L_\ell = I_\ell$.

Consider the network topology presented in figure 3.1(a). The topology is known as the Concentrator Network. Node C acts as a feeder node for the $2n$ ingress nodes $S_1, \ldots, S_{2n}$ and

[3]Refer to chapter 2 for a detailed discussion of critical link sets and the computation of the MIRA link interference.

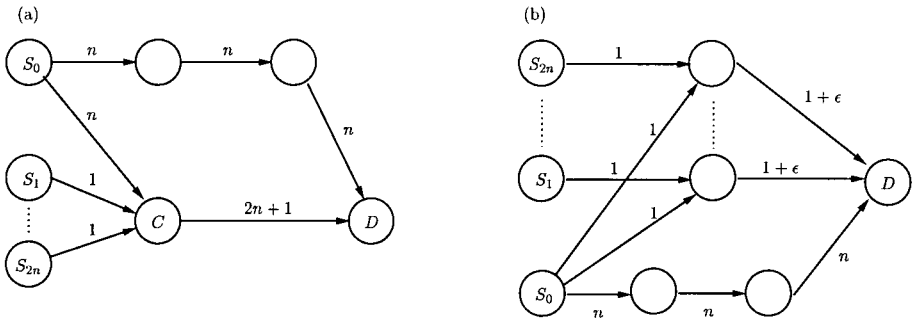

Figure 3.1: (a) The Concentrator Network (b) The Distributor Network.

a high capacity link, the *fat link* $C - D$, connects the feeder node C to the egress node D. Furthermore the ingress node S_0 is connected to the egress node D via two paths: a 2-hop path which traverses the fat link and an alternative 3-hop path.

Consider the random arrival sequence of n LSP setup requests between ingress-egress pair (S_0, D) and one LSP setup request between each of the ingress-egress pairs $(S_1, D), \ldots, (S_{2n}, D)$. All LSP requests are for one bandwidth unit. We employ the LI routing scheme to select paths along which to set up the requested LSPs. We consider two cases, the first case employing the interference measure proposed in equation 3.1 and the second case employing the interference measure proposed in equation 3.2:

- *Case 1:* First consider the interference parameter proposed in equation 3.1 that has the same effect as routing with the MIRA algorithm. As a consequence of its interference calculation, the MIRA algorithm will achieve the same routing performance as the MHA algorithm: all LSPs between (S_0, D) will be routed along the 2-hop path, thus saturating the fat link and making it impossible to route all the $2n$ LSP requests between $(S_1, D), \ldots, (S_{2n}, D)$. This is because the MIRA algorithm will never consider the fat link $C - D$ to be a critical link since it is not in the minimum cut for any of the ingress-egress pairs[4].

- *Case 2:* Now consider the case where the interference parameter is set equal to the interference measure proposed in equation 3.2. In this case the LI routing strategy will detect the popularity of the fat link $C - D$ and a many of the LSP requests between (S_0, D) will be routed along the alternative 3-hop, leaving more available bandwidth on the popular fat link for the requests between $(S_1, D), \ldots, (S_{2n}, D)$.

[4]The reader can confirm the behaviour of the MIRA algorithm by referring to the critical link conditions presented in chapter 2.

For the scenario described above, the interference measure proposed in equation 3.2 achieves a better bandwidth assignment than the interference measure proposed in equation 3.1. However, routing only according to the number of flows on the network links also has limitations. As an example, refer to the network topology known as the Distributor Network presented in figure 3.1(b). We consider the same LSP request arrival scenario as before: n LSP requests between ingress-egress pair (S_0, D) and one LSP request between each of the ingress-egress pairs $(S_1, D), \ldots, (S_{2n}, D)$, all LSP requests being for a single bandwidth unit.

In this case assignments to the interference parameter I_ℓ as defined in both equation 3.1 and equation 3.2 will result in the same inefficient bandwidth utilization – each of the LSPs between (S_0, D) will be routed along one of the 2-hop paths, thus each time choking off the $1 + \epsilon$ link, and consequently making it impossible to route all of the LSP requests between $(S_1, D), \ldots, (S_{2n}, D)$. In the case of the MIRA interference parameter calculation this is because none of the $1 + \epsilon$ links will be identified as being critical. Furthermore, since each of the $1 + \epsilon$ links will either be empty or carry one LSP (in which case the link is infeasible for further path selection), these links will never appear to be popular links to the LI routing strategy with interference parameter as proposed in equation 3.2.

For the remainder of this thesis we will use the interference measure proposed in equation 3.2 as the value of I_ℓ due to its simplicity, computational efficiency and the fact that it may be more easily deployed in communication networks. We will justify this choice by means of performance evaluation in chapter 4.

3.1.3 The Residual Link Capacity: $c_\ell - r_\ell$

Routing decisions based only on the value of the interference parameter are subject to certain limitations that do not always lead to an efficient LSP configuration. It is therefore necessary that the LIO routing framework not only bases its routing decisions on the interference parameter, but also be aware of the bandwidth distribution in the network.

Routing algorithms such as OSPF and CSPF base their routing decisions on the bandwidth distribution in the network, CSPF being more dynamic since it not only considers the static link capacities, but also the effect that its bandwidth assignment has on the overall bandwidth distribution in the network. The LIO routing strategy uses the same mechanism as the CSPF routing strategy to keep track of the current bandwidth distribution in the network by considering the residual capacities of the links.

Consider once again the Concentrator and Distributor network topologies depicted in figure 3.1 and the same LSP request arrival scenario as discussed in the previous section. Adding awareness of the bandwidth distribution to the two interference measures defined in equation 3.1 and 3.2 has the following effect:

- For both the Concentrator and the Distributor networks, the LIO routing strategy with the interference parameter I_ℓ as defined in equation 3.1 (the MIRA interference parameter) will perform similar to the CSPF routing strategy. This is because the interference calculation will still not identify any critical links, hence $I_\ell = 1$ for all links, yielding a link cost function similar to the CSPF link cost function.

- If I_ℓ is defined as in equation 3.2, the LIO strategy will also perform similar to the CSPF routing strategy for the Distributor network topology. However, for the Concentrator network topology, LIO will perform better than the CSPF routing strategy since the LIO strategy will be more sensitive to saturation of the fat link $C - D$ than CSPF. Thus the LIO strategy will route more of the LSP requests between (S_0, D) along the alternative 3-hop path than CSPF. (Since the fat link has more than double the amount of capacity than the links of the alternative 3-hop path, the CSPF routing strategy will regard the 2 hop path as the minimum cost path until at least $n + 1$ LSPs traverse the fat link. The LIO strategy will consider the 3-hop path earlier.)

3.1.4 The Calibration Parameter: α

It is important to realize that, in the case where I_ℓ is defined as in equation 3.2, there is a correlation between the number of flows and the amount of reserved bandwidth on a link: $r_\ell \propto I_\ell$. One could therefore conclude that since the two cost metrics I_ℓ and r_ℓ both respond linearly to network state changes, the use of both might be cumbersome. However, each of these cost metrics represents a distinct routing objective and the assignment of different weights to each of these metrics will result in specialized routing outcomes as might be required for networks or network traffic with distinctive characteristics or to satisfy particular network policies: consider for example networks with a big bandwidth demand size variation or a network policy that provisions for network failure scenarios. We therefore introduce the calibration parameter α.

The calibration parameter $0 \leq \alpha \leq 1$ expresses a trade-off between the link interference

and the residual link capacity. By varying the value of α, it is possible to control the relative importance of the respective cost metrics. The variation of the α parameter will thus have an influence on the behaviour of the routing strategy.

Note the resulting effect when the calibration parameter takes on one of its extreme values:

$$L_\ell = \begin{cases} 1/(c_\ell - r_\ell) & \alpha = 0 & \text{CSPF routing} \\ 1/c_\ell & \alpha = r_\ell = 0 & \text{OSPF routing} \\ I_\ell & \alpha = 1 & \text{LI routing} \end{cases}$$

If the calibration parameter is set to 0, we obtain a link cost function identical to the link cost function of CSPF. Furthermore, if link ℓ is unloaded ($r_\ell = 0$), the link cost function imitates the link cost function of OSPF. Setting the calibration parameter to 1 will base all routing decisions exclusively on the link interference value to obtain a least interference path.

When $0 < \alpha < 1$ the link cost function yields a mix of LI routing and CSPF routing. The influence of the calibration parameter α on the behaviour of the routing strategy is explored in chapter 4.

3.1.5 An Additive Penalty Function

The link cost expressed in the link cost function 3.1 is a multiplicative composition rule of the link interference and the inverse of the residual link capacity and these two metrics are balanced by the power parameter α. However, in the case where queuing and propagation delays are involved in the routing cost, a more complex routing objective may require the incorporation of additional cost metrics expressed by additive composition rules. It may therefore be useful to consider a logarithmic transform of the multiplicative penalty function expressed in link cost function 3.1.

Link Cost Function 3.2 (LIOA(b)) *Let $c_m \geq \max_{\ell \in \mathcal{L}} c_\ell$. The additive LIO link cost function for link ℓ with calibration parameter $0 \leq \alpha \leq 1$ is given by*

$$L_\ell = \alpha \log I_\ell + (1 - \alpha) \log \frac{c_m}{c_\ell - r_\ell} + F_\ell$$

where $I_\ell > 0$ is the interference parameter, c_ℓ is the total capacity, r_ℓ is the total reserved capacity and F_ℓ is a parameter expressing additional cost metrics for link ℓ .

We make the following observations regarding link cost function 3.2:

- By considering the logarithmic transform of the link cost function presented in 3.1, we obtain an additive composition rule expressing a similar trade-off between link interference and residual link capacity in the form of a weighted sum. This additive composition rule enables the simple extension of the LIO model to include other additive cost metrics expressed by parameter F_ℓ in link cost function 3.2.
- The logarithmic transform requires some additional mathematical manipulation due to the mathematical behaviour of the logarithmic function. Since $\log(\frac{1}{c_\ell - r_\ell}) < 0$, we introduce the parameter c_m to avoid the occurrence of negative link costs. The parameter c_m can be set to any positive value, but should have at least the value of the biggest link capacity of all the links in the network.

Initial experiments showed that the link cost function 3.2 and link cost function 3.1 are equivalent (for the case where $F_\ell = 0$ and consequently no additional cost metrics are taken into account) and since the investigation of the use of additional cost metrics is beyond the scope of this thesis, we will use the multiplicative link cost function 3.1 for the remainder of this thesis.

3.2 The Least Interference Optimization Algorithm

In this section we present the *Least Interference Optimization Algorithm (LIOA)*. LIOA finds a least cost path along which to set up a bandwidth guaranteed LSP by assigning weights to the links of the network according to link cost function 3.1. The algorithm assumes that LSP setup requests arrive one by one and are routed in order of arrival. Future LSP demands are unknown. Note that although the LIO model does not restrict the choice of the interference parameter I_ℓ, the algorithm uses the interference parameter under investigation in this thesis, proposed in section 3.1.2. The algorithm can however be extended to include any choice of interference parameter.

LEAST INTERFERENCE OPTIMIZATION ALGORITHM (LIOA)

Input:

A graph $G(\mathcal{N}, \mathcal{L})$ and an ingress-egress pair (i, e) between which a flow of $d_{i,e}$ units has to be

routed.

Output:
A least cost path p^* between i and e having a capacity of $d_{i,e}$ units of bandwidth.

Algorithm:

1. Form a pruned network topology by setting $L_\ell := \infty$ if $c_\ell - r_\ell < d_{i,e}$.
2. Compute the link weights of the pruned topology by setting
$$L_\ell := \frac{I_\ell^\alpha}{(c_\ell - r_\ell)^{1-\alpha}}$$
3. Using Dijkstra's algorithm, compute the least cost path p^* in the pruned network topology using L_ℓ as the weight of link ℓ.
4. Route the demand of $d_{i,e}$ units between i and e along the shortest path p^* and for every $\ell \in p^*$ set
$$\begin{aligned} I_\ell &:= I_\ell + 1 \\ r_\ell &:= r_\ell + d_{i,e}. \end{aligned}$$

3.2.1 Complexity Analysis

The complexity of the LIOA algorithm is determined by the shortest path computation which has complexity $O(N^2)$ when shortest paths are computed with Dijkstra's algorithm. The computational complexity of the LIOA algorithm is thus similar as that of popular routing algorithms such as MHA, OSPF and CSPF. Note however that if an interference measure other than the interference measure proposed in this thesis is used, the computational complexity might be determined by the computational complexity of the interference parameter.

3.3 Concluding Remarks

This chapter introduced the Least Interference Optimization framework and presented the LIOA algorithm for online establishment of bandwidth guaranteed LSPs in MPLS networks.

The online method uses a link cost function that reduces the interference among competing flows by reducing the number and magnitudes of flows carried on the links. The link cost is presented as both a multiplicative and additive link cost function and contains a calibration parameter that expresses the trade-off between competing routing objectives.

The LIOA algorithm is simple in terms of computational complexity and implementation. LIOA achieves the same time complexity as classical intra-domain routing algorithms and restricts its use of dynamic information to information derivable from current routing protocols [29, 33, 40], thus making it a viable option for deployment by ISPs.

Chapter 4

Performance Evaluation of Online Routing Strategies

This chapter evaluates and compares the performance of several online routing strategies for the setup of bandwidth guaranteed LSPs in MPLS networks. The routing strategies under consideration are the Minimum Hop Algorithm (MHA), the Minimum Interference Routing Algorithm (MIRA), Open Shortest Path First (OSPF) routing, Constraint Shortest Path First (CSPF) routing and the newly proposed Least Interference Optimization Routing Algorithm (LIOA). The experimental environment is presented in section 4.1 while all simulation experiments are discussed in sections 4.2 and 4.3. A short conclusion is offered in section 4.4.

4.1 Experimental Environment

The performance of the online routing schemes are measured and evaluated by means of simulation and compared with regards to various performance metrics. In this section we describe the experimental environment and conditions under which all performance evaluation experiments take place. We present the network topologies, the simulation method, the assumptions and the relevant performance metrics used to evaluate the online routing strategies.

4.1.1 The Network Topologies

Three network topologies are considered for the performance evaluation of the online routing methods: a 12-node network, a 15-node network and a 23-node network. The relevant network topology statistics are given in table 4.1 and graphical representations of the networks can be obtained from Appendix A. All network topologies consist of two link types: OC-48 and

	12-node	15-node	23-node
Nodes	12	15	23
Ingress-egress pairs	132	210	506
OC-48 links	4	12	12
OC-12 links	32	44	64
Total bandwidth	576	1104	1344
Average node degree	3	3.8	3.8
Max node degree	4	6	6
Min node degree	2	2	2

Table 4.1: Topological Network Information

OC-12 links with 48 and 12 units of bandwidth respectively. All links are uni-directional and to enable the establishment of a large number of LSPs, link capacities are scaled by a constant factor ω as indicated in each experiment. Each node represents a possible ingress node or egress node.

The 15-node network is identical to the network topology presented in [31] and was chosen due to its popularity for online routing performance studies (see for example [11, 41, 43]). The 12-node network was obtained from [11] and represents a commercial backbone topology of the USA. The 23-node network is a fictitious representation of a backbone topology for the USA.

4.1.2 The Simulation Model

The simulation model used for performance evaluation is based on a Markov chain simulation model that only recognizes connection level events. A description of the simulation model is given in Appendix B.

In the context of this thesis, a simulation *event* refers either to the arrival of an LSP setup request to one of the ingress-egress pairs of the network or to an LSP tear-down operation. The total number of events E consists of LSP setup request arrivals A and LSP tear down operations D, thus $E = A + D$. A simulation *trial* (or simulation run) refers to a fixed number of sequentially simulated events and each independent trial is seeded with a unique seed value to obtain a distinct random sequence of simulation events.

The simulation process considers a network topology as directed graph $G(\mathcal{N}, \mathcal{L})$ where $\mathcal{N}$ denotes the set of nodes and $\mathcal{L}$ denotes the set of network links. The event generation process

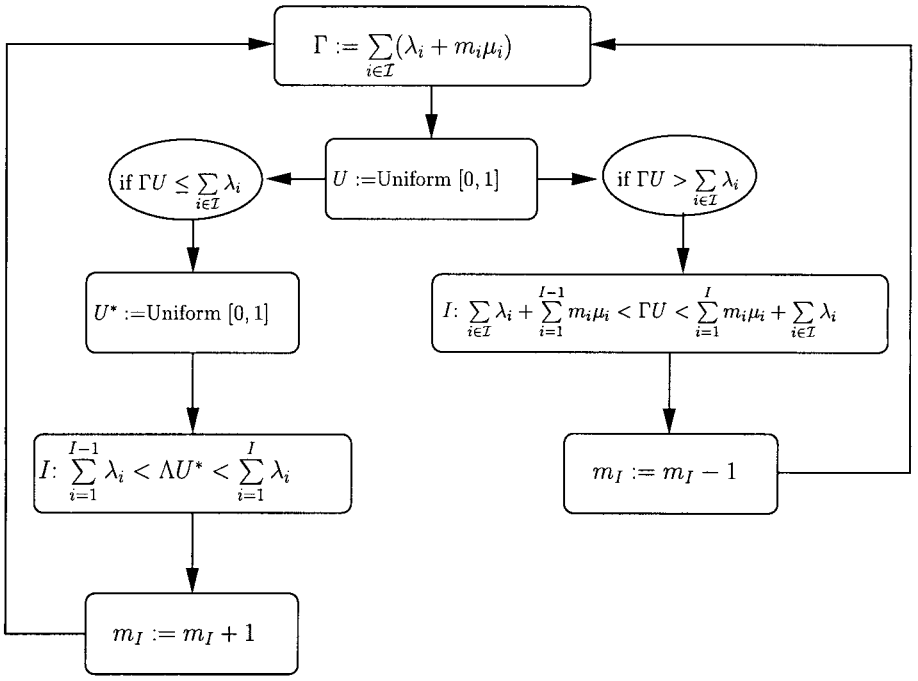

Figure 4.1: Generation of Simulation Events.

can briefly be described as follows: let $\mathcal{I}$ denote the set of ingress-egress pairs in the network. An LSP setup request arrives to ingress-egress pair $i \in \mathcal{I}$ according to a Poisson process with an average arrival rate λ_i and the total arrival rate to the network is denoted by $\Lambda = \sum_{i\in\mathcal{I}} \lambda_i$. Each LSP setup request is for d_i units of bandwidth. When an LSP setup request arrives to ingress-egress pair i, an online routing strategy is employed to compute a feasible path of d_i units of bandwidth along which to establish the LSP. If such a path exists the LSP is established, otherwise the LSP setup request is rejected. LSP holding times are exponentially distributed with parameter $1/\mu_i$.

The generation of a single simulation event is depicted in the diagram in figure 4.1 where U and U^* are independent uniformly distributed random variables[1], m_i is the number of established LSPs connecting ingress-egress pair i and $I \in \mathcal{I}$ indicates the ingress-egress pair to which an LSP setup request arrive or for which an LSP is torn down.

The simulation input is the number of simulation trials T, the number of simulation events E for each simulation trial, a file containing the network description and the online routing strategy used to determine the paths along which the LSPs will be routed.

[1]The use of two independent random variables in the event generation process is explained in Appendix B.

The simulation output is the relevant performance measures that are discussed in section 4.1.4. Performance measures are calculated from simulation data that was recorded at frequent intervals of the simulation process and which reflects the state of the network at various intervals during the progress of the simulation.

Note that the simulation model is event-driven rather than time-driven although the notion of time is present in the simulation model (mainly expressed by the LSP holding times). We will consider the advance of the simulation process in terms of simulation events, and in particular the event of LSP setup request arrivals to the network.

4.1.3 Simulation Modes

We consider two distinctive simulation modes: a *dynamic* simulation mode and a *static* simulation mode. In dynamic mode the LSP setup requests arrive to each of the ingress-egress pairs of the network according to a Poisson process with an average arrival rate λ_i. If an LSP request is accepted it remains in the network for an exponentially distributed time with average μ_i, after which it is torn down and the bandwidth reserved by the LSP is returned to the network.

In the static simulation mode LSP setup requests also arrive according to a Poisson process with an average arrival rate λ_i, but after an LSP has been accepted, it remains in the network for the full duration of the simulation. Although not representative of realistic network operation, static simulation gives us the opportunity to experiment with the online routing strategies when routing long-lived connections. Unlike in the case of dynamic simulation where bandwidth is periodically allocated and released, bandwidth allocation to LSPs in static mode has an effect on all future network states since once bandwidth has been allocated it is lost to the network indefinitely. Static simulation therefore accentuates the routing decisions made by the various routing strategies and, unlike in dynamic simulation mode, each inefficient routing decision will be reflected in the resulting network performance.

We also experiment with a combination of dynamic and static simulation – a portion of the accepted LSP setup requests remain in the network for an infinite time period while other accepted LSP requests remain in the network for an exponentially distributed time period. This situation is more representative of realistic network behaviour.

In general an LSP is not set up for a single network connection, but rather once an LSP

has been set up it is used by a large number of network connections. One will therefore not expect LSPs to be set up and torn down frequently, and consequently it is reasonable to assume that LSPs will have a long lifetime (relative to the holding time of a network connection for example).

4.1.4 Performance Measures

To compare the competing routing strategies, the following performance measures are considered:

LSP Rejection

The primary performance measure considered in this thesis is the LSP request rejection. LSP setup requests are rejected when a feasible path along which to route the LSP cannot be established due to insufficient bandwidth availability. The availability of bandwidth is influenced by bandwidth allocation decisions when LSPs are being set up, and thus the LSP rejection gives an overall reflection of the routing decisions taken by the respective routing strategies. The LSP rejection is expressed by the following metrics:

- The number of rejected LSP setup requests for trial t: R_t.
- The LSP acceptance ratio for trial t after A_t LSP request arrivals:

$$a_t = 100(1 - \frac{R_t}{A_t}). \tag{4.1}$$

- The average number of rejected LSP setup requests for T trials:

$$R = \frac{1}{T}\sum_{t=1}^{T} R_t. \tag{4.2}$$

  (The average acceptance ratio a for T trials is defined similarly.)

- The variance of the number of rejected LSP requests for T trials:

$$R_\sigma^2 = \frac{1}{(T-1)}\sum_{t=1}^{T}(R_t - R)^2 \tag{4.3}$$

  and the standard deviation of the number of rejected LSP requests for T trials:

$$R_\sigma = \sqrt{R_\sigma^2}. \tag{4.4}$$

- The improvement index IMP_z which indicates the percentage of improvement in the average LSP rejection R of a routing strategy over strategy z where z indicates the routing strategy achieving the worst performance:

$$IMP_z = 100\frac{R_z - R}{R_z}. \tag{4.5}$$

Note that all averages are calculated for a confidence interval with 95% certainty. Refer to Appendix C for a discussion on acquiring confidence intervals.

Link Utilization

The link utilization for link $\ell \in \mathcal{L}$ expresses the percentage of the total link capacity c_ℓ that is reserved by all LSPs traversing link ℓ. Keeping the link utilization as low as possible on each link is a primary objective for all the routing strategies under investigation and plays an important role in congestion avoidance. We will define the link utilization performance measure in section 4.3.2.

Flow Analysis

The available flow for an individual ingress-egress pair gives an indication of the amount of unreserved bandwidth on all feasible paths that connect the ingress-egress pair along which future LSPs requests can be set up. Although the flow analysis is closely related to the link utilization, there is a fundamental difference between these measures: even though most of the network links may be under-utilized this does not necessarily imply efficient bandwidth allocation by the routing strategy, since the existence of bottleneck links may hinder the setup of feasible LSPs. The flow analysis thus provides a better measure for the efficiency of bandwidth allocation.

We will consider performance measures such as the total available network flow between all ingress-egress pairs in the network and smallest available flow value between the individual ingress-egress pairs. These performance measures are defined in section 4.3.3.

Path Quality

The online routing strategies are evaluated with regards to the quality and characteristics of the paths that have been calculated for the respective LSPs. The analysis includes measures

such as the path length, path length frequency, path multiplicity and path preference. These measures are further discussed in section 4.3.4.

4.1.5 Assumptions

The following assumptions are made and apply to all online routing experiments:

- *No traffic splitting:* Although traffic splitting is often used for load balancing purposes, it is not permissible for a routing algorithm to split traffic demands in an arbitrary manner. We assume the setup of LSPs for a fixed bandwidth demand without taking the possibility of traffic splitting into account.
- *No admission control:* We assume that no admission control is imposed on arriving LSP requests. If there is enough available bandwidth and there exists a feasible path for the LSP request, the request will be accepted even if setting up the LSP might lead to network congestion. A LSP setup request will only be rejected if a feasible path does not exist.
- *No queuing:* We do not make provisioning for LSP setup requests that cannot be accepted on arrival and thus do not allow request queuing at any of the ingress nodes. A LSP request that cannot be accepted on arrival due to insufficient bandwidth availability will be rejected and lost to the network forever.
- *No propagation or signalling delays:* We do not take propagation or signalling delays into account. Hence, we assume that once an LSP has been accepted the effect of its establishment on the network resources is immediate and all LSRs are instantly aware of the state change in the network resources (all routing tables are updated instantaneously). The same conditions hold for an LSP tear-down operation.
- *Constant LSP request arrival rates:* We assume that all arrival rates of LSP setup requests to the respective ingress-egress pairs remain constant for the full duration of the simulation.
- *First-come-first-served:* We assume that LSP setup requests arrive one at a time and are served in order of arrival.

	12-node Network		15-node Network		23-node Network	
	Dynamic	Static	Dynamic	Static	Dynamic	Static
ω	10	100	10	100	10	100
λ	4	4	5	5	2	2
μ	1	∞	1	∞	1	∞
b	[1,4]	[1,4]	[1,4]	[1,4]	[1,4]	[1,4]
T	20	20	20	20	20	20
A	25000	10000	25000	13000	25000	13000

Table 4.2: (Experiment 4.1) Simulation Settings

4.2 Parameter Evaluation: LIO Routing

In chapter 3 we defined a multiplicative link cost function employed by the LIO routing strategy to calculate the respective link weights (refer to link cost function 3.1). The link cost function consists of three cost metrics: an interference parameter I_ℓ, the residual link capacity $c_\ell - r_\ell$ and a calibration parameter α. Before comparing of the LIO routing strategy proposed in chapter 3 to other online routing schemes, we evaluate the behaviour of the cost metrics expressed by the parameters of the LIOA cost function.

4.2.1 The Interference Parameter: I_ℓ

In chapter 3 two interference measures were proposed for the interference parameter I_ℓ in the LIOA cost function (see equations 3.1 and 3.2): the interference measure as calculated by the MIRA routing strategy and an interference measure indicating the number of LSPs traversing a link. We used illustrative examples to show that both the proposed interference measures have certain limitations, and in this section the performance of the two performance measures are evaluated by means of simulation.

Experiment 4.1 *A comparative evaluation of the LSP rejection achieved by the LIOA routing strategy in both static and dynamic simulation mode with different settings of the interference parameter I_ℓ.*

Experiment 4.1 is conducted using each of the three network topologies introduced in section 4.1.1. We assume that the LSP arrival rate λ is the same to each ingress-egress pair and the holding time μ is the same for each established LSP. The values of the link scaling factor ω, the LSP request arrival rate λ, the LSP holding time μ, the number of simulation trials T and the arrival events A per simulation trial for each of the three network topologies are

given in table 4.2. The LSP bandwidth demand sizes are uniformly distributed in the interval indicated by b in table 4.2. Note that in table 4.2 $\mu = \infty$ refers to a holding time longer than the duration of the simulation. For the LIOA routing strategy we set the calibration parameter $\alpha = 0.5$ to express an equal balance between the interference parameter I_ℓ and the residual link capacity $c_\ell - r_\ell$. The experiment is repeated for each of the proposed settings of the interference measure I_ℓ.

Figures 4.2, 4.3 and 4.4 plot the total number of rejected LSP requests at each simulation trial for each of the settings of the interference parameters in both dynamic and static simulation mode. It is clear from these results that for the three network topologies under consideration the performance measure proposed in equation 3.2 achieves lower LSP request rejection in both static and dynamic simulation mode (except for the static mode simulation of the 15-node network where the performance is almost identical). The performance measure proposed in equation 3.2 also has the advantage of a substantially lower computational complexity compared to the performance measure proposed in equation 3.1 since the latter includes numerous maxflow computations. For these reasons, we set the interference parameter I_ℓ equal to the number of LSPs traversing link ℓ for the remainder of our experiments.

4.2.2 The Calibration Parameter: α

The LIOA cost function includes a calibration parameter $0 \leq \alpha \leq 1$ that expresses the balance between the interference parameter I_ℓ and the residual link capacity $c_\ell - r_\ell$. It is mentioned in chapter 4 that there exists a relation between the number of LSPs traversing a link and the total reserved capacity of the link. In this section we investigate this relation by experimenting with different settings of the calibration parameter α under varying bandwidth demand size conditions. We also obtain an optimal setting for the α parameter for each bandwidth demand size variation.

Experiment 4.2 (α-analysis) *A comparative evaluation of the LSP rejection achieved by the LIOA routing strategy in both static and dynamic simulation mode for different values of the calibration parameter α under varying bandwidth demand conditions.*

Experiment 4.2 is conducted using each of the three network topologies introduced in section 4.1.1 and we assume that the LSP arrival rate λ is identical to each ingress-egress pair and the holding time μ is the same for each established LSP. The experimentation conditions are identical to those presented in table 4.2, but we repeat the experiment in both dynamic and static simulation mode for bandwidth demand sizes uniformly distributed between $[1, 4], [1, 7], [1, 10]$

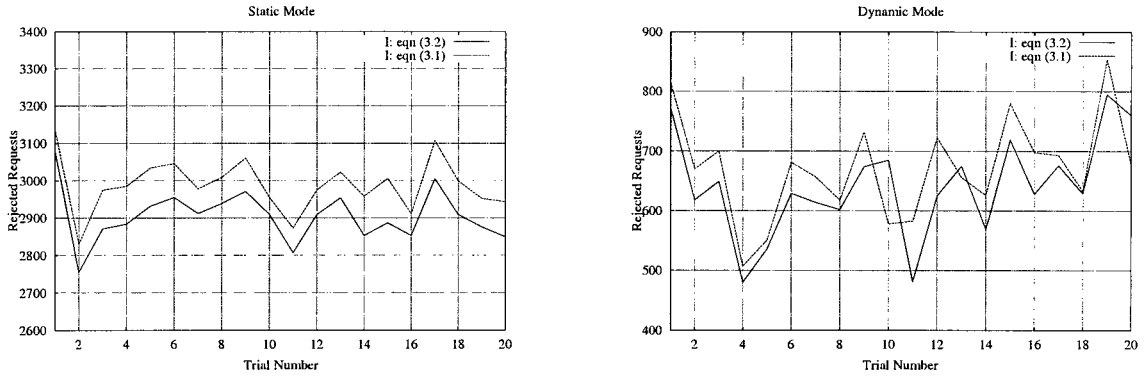

Figure 4.2: (Experiment 4.1) LSP Rejection: 12-node Network

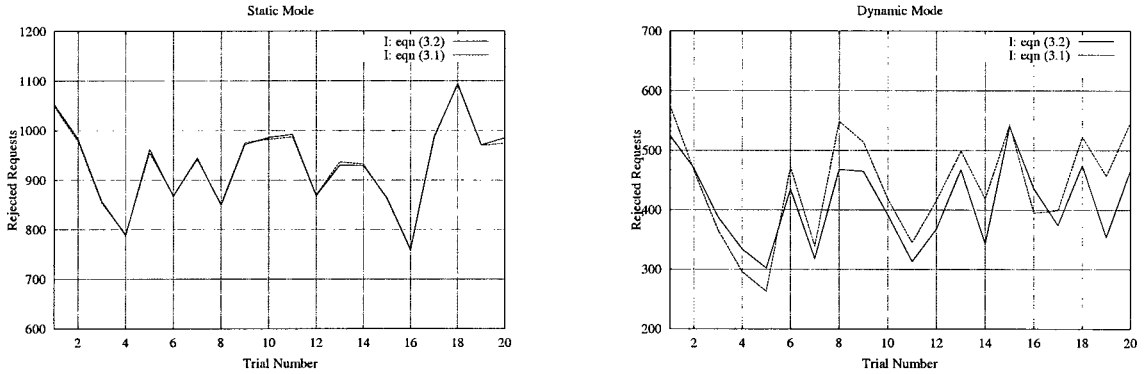

Figure 4.3: (Experiment 4.1) LSP Rejection: 15-node Network

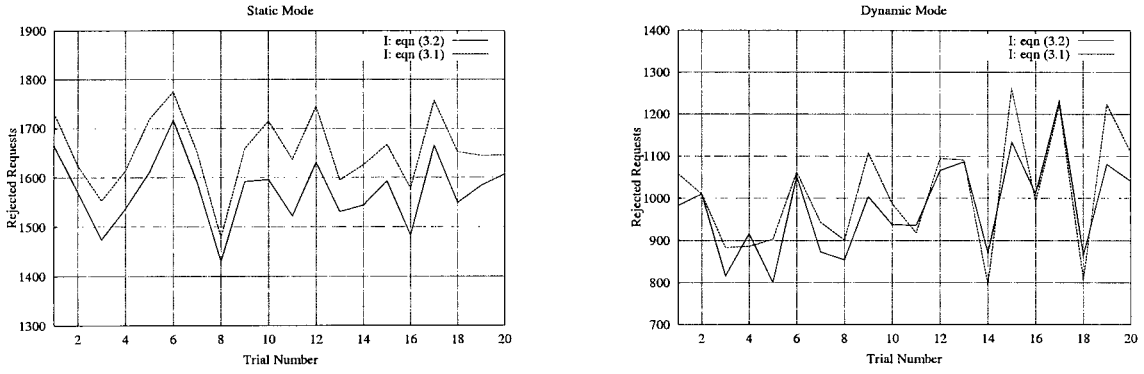

Figure 4.4: (Experiment 4.1) LSP Rejection: 23-node Network

	Dynamic Mode				Static Mode			
	$U[1,4]$	$U[1,7]$	$U[1,10]$	$U[1,15]$	$U[1,4]$	$U[1,7]$	$U[1,10]$	$U[1,15]$
$\alpha = 0.0$	704	5419	8230	10873	2953	4940	5994	6935
$\alpha = 0.3$	672	5359	8193	10875	2920	**4930**	**5983**	**6916**
$\alpha = 0.5$	**635**	**5351**	**8166**	**10859**	**2906**	4932	5988	6918
$\alpha = 0.7$	661	5410	8191	10928	2907	4955	6017	6937
$\alpha = 1.0$	747	5572	8360	11046	3053	5171	6168	7018

Table 4.3: (Experiment 4.2) α-analysis: 12-Node Network.

	Dynamic Mode				Static Mode			
	$U[1,4]$	$U[1,7]$	$U[1,10]$	$U[1,15]$	$U[1,4]$	$U[1,7]$	$U[1,10]$	$U[1,15]$
$\alpha = 0.0$	572	5954	8727	11278	959	4479	6318	7953
$\alpha = 0.3$	474	5915	8674	11266	936	4469	**6313**	**7920**
$\alpha = 0.5$	427	**5847**	**8649**	**11265**	932	**4466**	6314	7926
$\alpha = 0.7$	**426**	5850	8683	11304	**928**	4485	6343	7976
$\alpha = 1.0$	728	6303	8999	11545	1021	4802	6634	8189

Table 4.4: (Experiment 4.2) α-analysis: 15-Node Network.

	Dynamic Mode				Static Mode			
	$U[1,4]$	$U[1,7]$	$U[1,10]$	$U[1,15]$	$U[1,4]$	$U[1,7]$	$U[1,10]$	$U[1,15]$
$\alpha = 0.0$	1083	5194	7985	10966	1649	4030	5764	7673
$\alpha = 0.3$	1003	**5127**	**7878**	**10922**	1597	3985	**5638**	**7579**
$\alpha = 0.5$	**971**	5147	7913	10933	**1574**	**3974**	5652	7621
$\alpha = 0.7$	1027	5191	8073	11012	1600	4004	5827	7718
$\alpha = 1.0$	1413	5736	8544	11344	1864	4449	6214	7933

Table 4.5: (Experiment 4.2) α-analysis: 23-Node Network.

and $[1, 15]$ units of bandwidth respectively while varying the α parameter between 0 and 1.

The results for each of the network topologies are denoted in tables 4.3, 4.4, and 4.5. Each table entry corresponds to the average of the total number of rejected LSP setup requests R over all simulation trials and for each bandwidth demand variation the lowest number of rejected LSP requests are indicated. We thus obtain the optimal α value for each demand setting. We make the following observations:

- For smaller demand size variation ($U[1, 4]$, $U[1, 7]$) a setting of $\alpha \geq 0.5$ generally results in lower LSP request rejection and hence LIOA obtains better bandwidth allocation if more weight is given to the interference parameter than to the residual link capacity.
- For larger demand size variation ($U[1, 10]$, $U[1, 15]$) a setting of $\alpha \leq 0.5$ generally results in lower LSP request rejection. Consequently it seems to be a good idea to assign more weight to the residual link capacity in the case where bandwidth demand sizes have a large variation. This is especially noticeable in static simulation mode.
- Note that $\alpha = 0$ corresponds to CSPF routing and $\alpha = 1$ corresponds to LI routing. However in none of the cases simulated CSPF or LI routing resulted in the lowest LSP rejection. This shows that basing the routing decision on a combination of the two cost metrics I_ℓ and $c_\ell - r_\ell$ as expressed by the LIOA cost function achieves better results than only considering one of the two cost metrics.

4.3 Performance Evaluation

In this section we conduct various simulation experiments to evaluate and compare the performance of the Minimum Hop Algorithm (MHA), the Minimum Interference Routing Algorithm (MIRA), Open Shortest Path First (OSPF) routing, Constraint Shortest Path First (CSPF) routing and the Least Interference Optimization Algorithm (LIOA) with regards to LSP request rejection, network flow availability, link utilization and path quality.

4.3.1 LSP Rejection

LSP rejection occurs when an LSP setup request arrives to the network, but cannot be established due to insufficient bandwidth availability. The LSP rejection gives an indication of the efficiency of the bandwidth allocation made by each of the routing strategies under

	12-node Network	15-node Network	23-node Network
ω	100	100	100
λ	4	5	2
μ	∞	∞	∞
b	[1,4]	[1,4]	[1,4]
T	40	40	40
A	10000	13000	13000

Table 4.6: (Experiment 4.3) Simulation Settings

investigation.

As mentioned before, simulation in static mode is not realistic but gives a good idea of the worst case performance of the routing strategies – each bandwidth allocation decision will be reflected in the final performance of the routing strategy.

Experiment 4.3 *A comparative evaluation of the LSP rejection achieved by the MHA, MIRA, OSPF, CSPF and LIOA routing strategies in static simulation mode.*

Experiment 4.3 is conducted using each of the three network topologies introduced in section 4.1.1. We assume that the LSP arrival rate λ is identical to each ingress-egress pair and once an LSP is accepted it remains in the network for an infinite time period. Table 4.6 gives the values of the simulation variables. In the case of the LIOA routing strategy the calibration parameter α is set equal to the optimal values obtained for each of the network topologies in experiment 4.2.

The results for each of the network topologies are presented in tables 4.7, 4.8 and 4.9. For each routing strategy we state the average number R rejected LSP requests over 40 trials, the standard deviation R_σ of the rejected LSP requests, the acceptance ration a and the improvement percentage IMP_{MHA} over the MHA algorithm.

The results show that MHA achieves the worst performance in terms of LSP rejection while the LIOA and CSPF routing strategies perform the best. In general the LIOA strategy performs slightly better than the CSPF strategy while obtaining improvements between 15%-41% over MHA. Note that for the 12-node and the 23-node network topologies the routing strategies LIOA, CSPF and MIRA that employ dynamic network state information outperform the routing strategies MHA and OSPF that rely only on static network state information.

Method	R	R_σ	a	IMP_{MHA}
MHA	3431	89	65.7	0
OSPF	3254	57	67.5	5
MIRA	3139	92	68.6	9
CSPF	2953	68	70.5	14
LIOA	2906	69	70.9	15

Table 4.7: (Experiment 4.3) 12-node Network: Static Mode

Method	R	R_σ	a	IMP_{MHA}
MHA	1570	60	87.9	0
MIRA	1380	65	89.4	12
OSPF	1171	69	90.0	25
CSPF	959	75	92.6	39
LIOA	928	72	92.7	41

Table 4.8: (Experiment 4.3) 15-node Network: Static Mode

Method	R	R_σ	a	IMP_{MHA}
MHA	2239	60	82.8	0
OSPF	1906	67	85.3	15
MIRA	1771	67	86.4	21
CSPF	1649	66	87.3	26
LIOA	1574	63	87.9	30

Table 4.9: (Experiment 4.3) 23-node Network: Static Mode

	12-node Network	15-node Network	23-node Network
ω	10	10	10
λ	4	5	2
μ	1	1	1
T	40	40	40
A	25000	25000	25000

Table 4.10: (Experiment 4.4) Simulation Settings

To evaluate the performance of the routing strategies in a dynamic network environment, we conduct two experiments in dynamic simulation mode exploring the behaviour of the routing strategies under varying network load conditions and different demand size variations.

Experiment 4.4 (Load Variation) *A comparative evaluation of the LSP rejection achieved by the MHA, MIRA, OSPF, CSPF and LIOA routing strategies in dynamic simulation mode under varying load conditions.*

Experiment 4.5 (Demand Size Variation) *A comparative evaluation of the LSP rejection achieved by the MHA, MIRA, OSPF, CSPF and LIOA routing strategies in dynamic simulation mode with different demand size variations.*

Experiments 4.4 and 4.5 are conducted using each of the three network topologies introduced in section 4.1.1. We assume that the average holding times for all established LSPs are identical. In experiment 4.4 all bandwidth demands are uniformly distributed between 1 and 4 units of bandwidth and the arrival rate λ is perturbed as explained below to obtain different network loads. In experiment 4.5 the arrival rates to the ingress-egress pairs are identical, but the bandwidth demands are uniformly distributed between $[1, 4], [1, 7], [1, 10]$ and $[1, 15]$ units of bandwidth respectively. Table 4.10 indicates the values of the simulation variables. In the case of the LIOA routing strategy the calibration parameter α is set equal to the optimal values obtained for each of the network topologies in experiment 4.2.

The results for experiment 4.4 are presented in tables 4.11, 4.12 and 4.13. The total arrival rate to the network is indicated by $\Lambda = \sum_{i \in \mathcal{I}} \lambda_i$. The network loads were chosen to model the following scenarios:

- *case 1:* The total network arrival rate when the arrival rates to all ingress-egress pairs are identical and set equal to the values presented in table 4.10.

- *cases 2 and 3:* The arrival rates presented in table 4.10 are slightly perturbed. In this case the arrival rate to each of the ingress-egress pairs differs, but the total network arrival rate is more or less the same.
- *case 4:* A large perturbation is applied to the arrival rates at each ingress-egress pair presented in table 4.10 resulting in heavy load conditions.

From tables 4.11, 4.12 and 4.13 we observe that in all cases the LIOA strategy outperforms the other routing strategies under consideration with regard to LSP request rejection, achieving up to 52% performance improvement over the MHA strategy. Note that this applies in the three situations where the arrival rates to the individual ingress- egress pairs are identical and different but with a similar network load (cases 1,2 and 3). Under heavy load conditions the performance difference between the various routing strategies is less.

The standard deviation R_σ indicates the consistency of the routing strategy and its sensitivity to different LSP request arrival sequences – a smaller standard deviation indicates a more consist response to different arrival sequences. In most cases the consistencies of the static routing strategies MHA and OSPF are similar while the consistencies of the dynamic routing strategies CSPF and MIRA are similar. In general however, the LIOA algorithm is the most consistent of all the routing strategies.

The results for experiment 4.5 are presented in tables 4.14, 4.15 and 4.16. The routing strategies behave in a manner similar to the observations made in experiment 4.4. However, as the demand size variation increases the performance difference indicated by IMP_{MHA} becomes less evident. This is due to the fact that the available bandwidth was not enough to demonstrate the different behaviour of the routing strategies. We observe once again that the LIOA routing strategy performs better than the other routing strategies under investigation and in most cases proves to be the most consistent.

We next conduct a single simulation experiment to evaluate the performance of the online routing strategies in a network environment where established LSPs have different holding times.

Experiment 4.6 *A comparative evaluation of the LSP rejection achieved by the MHA, MIRA, OSPF, CSPF and LIOA routing strategies in dynamic simulation mode with varying LSP holding times.*

Load	Method	R	R_σ	a	IMP_{MHA}
$\Lambda = 528$	MHA	987	105	96.1	0
	OSPF	775	113	96.9	21
	CSPF	704	101	97.2	29
	MIRA	701	99	97.2	29
	LIOA	635	94	97.5	36
$\Lambda = 529$	MHA	924	94	96.3	0
	OSPF	721	93	97.1	22
	MIRA	633	86	97.5	31
	CSPF	602	87	97.6	35
	LIOA	572	84	97.7	38
$\Lambda = 546$	MHA	1279	98	94.9	0
	OSPF	1051	107	95.8	18
	MIRA	978	103	96.1	24
	CSPF	932	103	96.3	27
	LIOA	874	85	96.5	32
$\Lambda = 661$	MHA	2872	133	88.5	0
	OSPF	2791	149	88.8	3
	MIRA	2650	158	89.4	8
	CSPF	2583	152	89.7	10
	LIOA	2473	132	90.1	14

Table 4.11: (Experiment 4.4) 12-node Network: Load variation

Load	Method	R	R_σ	a	IMP_{MHA}
$\Lambda = 1050$	MHA	895	93	96.4	0
	CSPF	572	101	97.7	36
	MIRA	554	104	97.8	38
	OSPF	500	89	98.0	44
	LIOA	426	71	98.3	52
$\Lambda = 1053$	MHA	981	93	96.1	0
	MIRA	668	118	97.3	32
	CSPF	667	115	97.3	32
	OSPF	639	91	97.4	35
	LIOA	565	91	97.4	42
$\Lambda = 1084$	MHA	1318	125	94.7	0
	CSPF	994	111	96.1	25
	MIRA	981	101	96.1	26
	OSPF	907	128	96.4	31
	LIOA	809	100	96.8	39
$\Lambda = 1473$	MHA	4707	127	81.2	0
	MIRA	4521	128	81.1	4
	CSPF	4459	124	82.3	5
	OSPF	4394	134	82.4	7
	LIOA	4266	124	82.9	9

Table 4.12: (Experiment 4.4) 15-node Network: Load variation

Load	Method	R	R_σ	a	IMP_{MHA}
$\Lambda = 1012$	MHA	1447	121	94.2	0
	OSPF	1268	147	94.9	12
	MIRA	1080	156	95.6	25
	CSPF	1083	129	95.6	25
	LIOA	971	108	96.1	33
$\Lambda = 1018$	MHA	1443	114	94.2	0
	OSPF	1268	124	94.9	12
	MIRA	1086	143	95.7	25
	CSPF	1087	132	95.7	25
	LIOA	978	124	96.1	32
$\Lambda = 1020$	MHA	1590	129	93.6	0
	OSPF	1440	155	94.2	9
	MIRA	1239	156	95.0	22
	CSPF	1252	121	95.0	21
	LIOA	1147	122	95.4	28
$\Lambda = 1530$	MHA	4806	118	80.8	0
	OSPF	4701	108	81.2	2
	MIRA	4613	112	81.6	4
	CSPF	4569	122	81.7	5
	LIOA	4505	112	82.0	6

Table 4.13: (Experiment 4.4) 23-node Network: Load variation

Load	Method	R	R_σ	a	IMP_{MHA}
$1 \leq b \leq 4$	MHA	987	105	96.1	0
	OSPF	775	113	96.9	21
	CSPF	704	101	97.2	29
	MIRA	701	89	97.2	29
	LIOA	635	94	97.5	36
$1 \leq b \leq 7$	MHA	5619	143	77.5	0
	OSPF	5539	134	77.8	1
	MIRA	5527	148	77.9	2
	CSPF	5419	139	77.3	4
	LIOA	5351	137	78.6	5
$1 \leq b \leq 10$	MHA	8379	115	66.5	0
	OSPF	8318	126	66.6	1
	MIRA	8316	130	66.7	1
	CSPF	8230	142	66.1	2
	LIOA	8166	94	66.2	3
$1 \leq b \leq 15$	MHA	11033	100	55.9	0
	OSPF	10960	115	56.2	1
	MIRA	10973	94	56.1	1
	CSPF	10873	74	56.5	1
	LIOA	10859	91	56.6	2

Table 4.14: (Experiment 4.5) 12-node Network: Request Size Variation

Load	Method	R	R_σ	a	IMP_{MHA}
$1 \leq b \leq 4$	MHA	895	93	96.4	0
	CSPF	572	101	97.7	36
	MIRA	554	104	97.8	38
	OSPF	500	89	98.0	44
	LIOA	426	71	98.3	52
$1 \leq b \leq 7$	MHA	6208	111	75.2	0
	MIRA	6078	100	75.7	2
	CSPF	5954	107	76.2	4
	OSPF	5968	82	76.1	4
	LIOA	5847	103	76.6	6
$1 \leq b \leq 10$	MHA	8887	86	64.5	0
	MIRA	8787	94	64.9	1
	CSPF	8727	94	65.1	2
	OSPF	8724	77	65.1	2
	LIOA	8649	74	65.4	3
$1 \leq b \leq 15$	MHA	11437	71	54.3	0
	MIRA	11379	93	54.5	1
	CSPF	11278	94	54.9	1
	OSPF	11339	80	54.7	1
	LIOA	11265	88	54.9	2

Table 4.15: (Experiment 4.5) 15-node Network: Request Size Variation

Load	Method	R	R_σ	a	IMP_{MHA}
$1 \leq b \leq 4$	MHA	1447	121	94.2	0
	OSPF	1268	149	94.9	12
	MIRA	1080	156	95.6	25
	CSPF	1083	129	95.6	25
	LIOA	971	108	96.1	33
$1 \leq b \leq 7$	MHA	5380	125	79.5	0
	MIRA	5245	112	79.0	2
	CSPF	5194	114	79.2	3
	OSPF	5284	110	78.8	2
	LIOA	5127	99	79.4	5
$1 \leq b \leq 10$	MHA	8129	106	67.5	0
	MIRA	8069	156	67.7	1
	CSPF	7985	113	68.1	2
	OSPF	8093	109	67.6	1
	LIOA	7878	110	68.4	3
$1 \leq b \leq 15$	MHA	11061	84	55.8	0
	MIRA	11013	107	56.0	1
	CSPF	10966	91	56.1	1
	OSPF	11048	107	55.8	0
	LIOA	10922	87	56.3	1

Table 4.16: (Experiment 4.5) 23-node Network: Request Size Variation

	12-node Network	15-node Network	23-node Network
ω	20	20	20
λ	4	5	2
b	[1,4]	[1,4]	[1,4]
T	40	40	40
A	15000	15000	15000

Table 4.17: (Experiment 4.6) Simulation Settings

Experiment 4.6 is conducted using each of the three network topologies introduced in section 4.1.1. We assume that the LSP arrival rate λ is identical to each ingress-egress pair. Table 4.17 gives the values of the simulation variables. In the case of the LIOA routing strategy we set the calibration parameter $\alpha = 0.5$. We simulate two types of LSPs: 80% of the arriving LSPs are assumed to have exponentially distributed holding times with average μ while the remaining 20% are assumed to remain in the network for an infinite time period once they are accepted.

The results for experiment 4.6 are given in tables 4.18, 4.19 and 4.20 for each of the three network topologies. The MHA achieves the highest LSP rejection while the LIOA routing strategy improves on the performance of the other routing strategies under investigation. Note that the a ratio is low, especially in the case of the 12-node network topology. This can be attributed to the limited bandwidth availability and the fact that 20% of the LSP connections are never torn down.

4.3.2 Link Utilization

The link utilization u_ℓ for link $\ell \in \mathcal{L}$ expresses the percentage of the total link capacity c_ℓ that is reserved by all LSPs traversing link ℓ and is defined as:

$$u_\ell = 100\frac{r_\ell}{c_\ell}. \tag{4.6}$$

In this section we conduct simulation experiments to explore the behaviour of the routing strategies under investigation with regards to link utilization.

Experiment 4.7 *A comparative evaluation of the performance of the MHA, MIRA, OSPF, CSPF and LIOA routing strategies in both dynamic and static simulation mode with regard to the link utilization.*

Method	R	R_σ	a	IMP_{MHA}
MHA	7329	252	51.1	0
MIRA	7305	243	51.3	1
OSPF	7274	237	51.5	1
CSPF	7244	264	51.8	1
LIOA	7186	220	52.1	2

Table 4.18: (Experiment 4.6) 12-node Network

Method	R	R_σ	a	IMP_{MHA}
MHA	3499	189	76.7	0
MIRA	3386	202	77.4	3
OSPF	3251	198	78.2	7
CSPF	3125	199	79.2	11
LIOA	3087	207	79.4	12

Table 4.19: (Experiment 4.6) 15-node Network

Method	R	R_σ	a	IMP_{MHA}
MHA	5404	147	64.0	0
OSPF	5395	172	64.0	0
MIRA	5302	177	64.7	2
CSPF	5125	162	65.8	5
LIOA	5029	143	66.5	7

Table 4.20: (Experiment 4.6) 23-node Network

Experiment 4.7 is conducted using the 12-node network, the 15-node network and the 23-node network topologies. We assume that the LSP arrival rate λ is identical to each ingress-egress pair and the holding time μ is the same for each established LSP. The simulation environment is identical to the environment presented in table 4.2 for static and dynamic mode but only a single simulation trial is considered. In the case of the LIOA routing strategy the calibration parameter α is set equal to the optimal values obtained in experiment 4.2.

The results of experiment 4.7 are depicted in the graphs presented in figures 4.5, 4.6 and 4.7. The graphs indicate the number of links with a link utilization factor $u_\ell > 95\%$ at various intervals of the simulation for both static and dynamic simulation mode.

The highest number of links with $u_\ell > 95\%$ is observed for the MHA routing strategy, mainly because MHA will continue to choose highly utilized links for its shortest paths until the link bandwidth is exhausted before taking other links into account. Slightly better performance is observed for the OSPF routing strategy since the latter considers the link capacities, thus choosing bigger links first. Both the CSPF and LIOA algorithm improve on the performance of MIRA, MHA and OSPF since their link cost functions take the current link bandwidth availability into account and compute a least cost path that minimizes the total link utilization of the links of the path. In general we observe that the LIOA routing strategy performs better than the CSPF routing strategy.

4.3.3 Flow Availability

The available flow for an ingress-egress pair gives an indication of the amount of unreserved bandwidth on all feasible paths that connect the ingress-egress pair along which future LSPs can be set up. The available flow[2] between an ingress-egress pair can be calculated using a Maximum Flow algorithm (see for example the Bellman-Ford algorithm [8]). We define the following cost metrics that express the network flow availability:

- F_i: the available flow between ingress-egress pair $i \in \mathcal{I}$.
- F_{tot}: the total network flow defined as the sum of the available flow values between each of the ingress-egress pairs

$$F_{tot} = \sum_{i \in \mathcal{I}} F_i. \tag{4.7}$$

[2]Refer to chapter 2 for a discussion on maxflow values

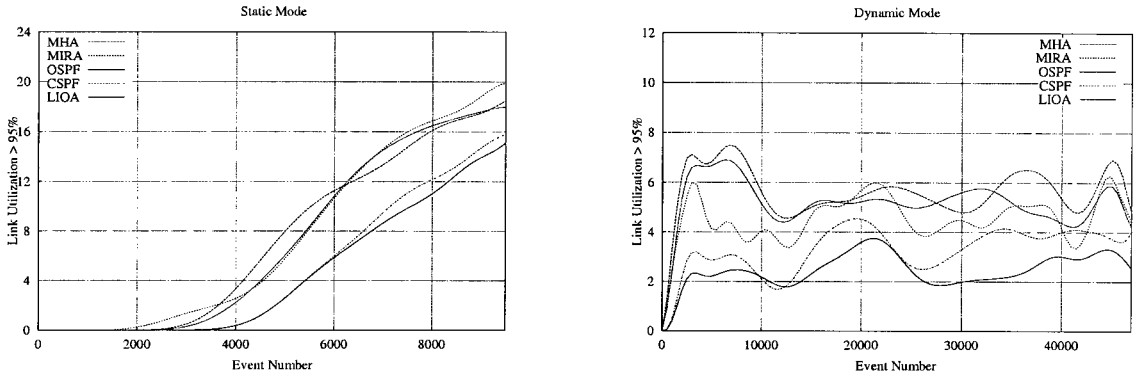

Figure 4.5: (Experiment 4.7) Link Utilization: 12-node Network

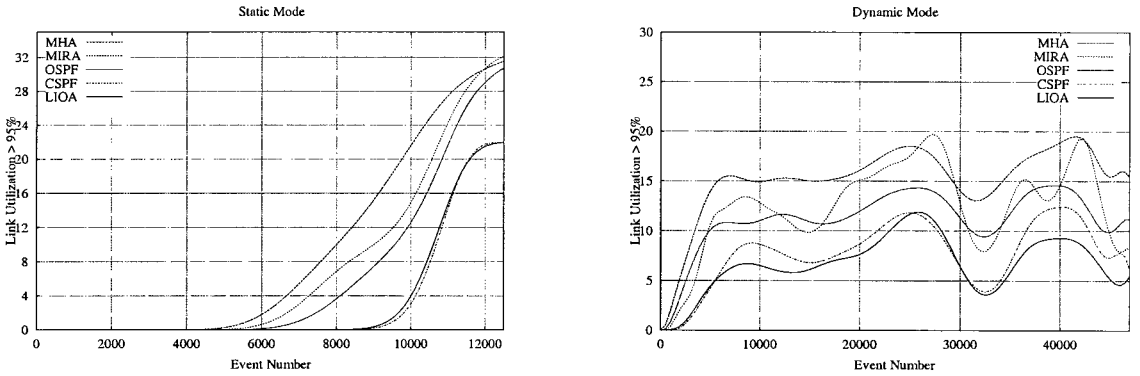

Figure 4.6: (Experiment 4.7) Link Utilization: 15-node Network

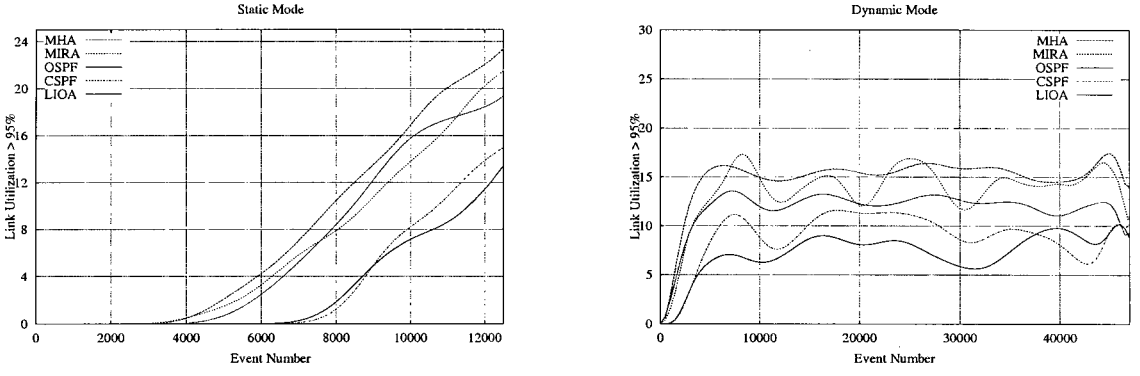

Figure 4.7: (Experiment 4.7) Link Utilization: 23-node Network

- F_{min}: the minimum flow value defined as the flow value of the ingress-egress pair with the lowest available flow

$$F_{min} = \min_{i \in \mathcal{I}} F_i. \tag{4.8}$$

We conduct two experiments recording the flow availability in the network at various simulation intervals to reveal the behaviour of the routing strategies under investigation:

Experiment 4.8 (Total Available Flow) *A comparative evaluation of the performance of the MHA, MIRA, OSPF, CSPF and LIOA routing strategies in both dynamic and static simulation mode with regards to the total network flow availability F_{tot} at various instants of the simulation.*

Experiment 4.9 (Minimum Available Flow) *A comparative evaluation of the performance of the MHA, MIRA, OSPF, CSPF and LIOA routing strategies in both dynamic and static simulation mode with regards to the minimum available flow F_{min} at various instants of the simulation.*

Experiments 4.8 and 4.9 are conducted using the same simulation environment as described in experiment 4.7. The simulation is performed in both dynamic and static simulation mode using the 12-node network, the 15-node network and the 23-node network topologies. For both experiments a single simulation trial is considered.

The results of experiments 4.8 and 4.9 are depicted in the graphs presented in figures 4.8 – 4.13. The graphs in figures 4.8–4.10 indicate the total network flow F_{tot} at different simulation times for each of the routing strategies. Except for the MHA routing strategy, all the routing strategies achieve more or less the same level of performance in the case of static simulation. However, under dynamic simulation conditions we observe that the LIOA routing strategy achieves the best performance and consequently achieves the most efficient bandwidth allocation. These results are consistent with the results obtained in experiments 4.3–4.6 and partially explain the fact that the LIOA routing strategy achieves the lowest LSP request rejection.

The graphs in figures 4.11–4.13 indicate the minimum flow value F_{min} between any of the ingress-egress pairs in the network at different simulation times as measured in experiment 4.9. The results are similar to those obtained in experiment 4.8 and the best performance is achieved by the LIOA routing strategy.

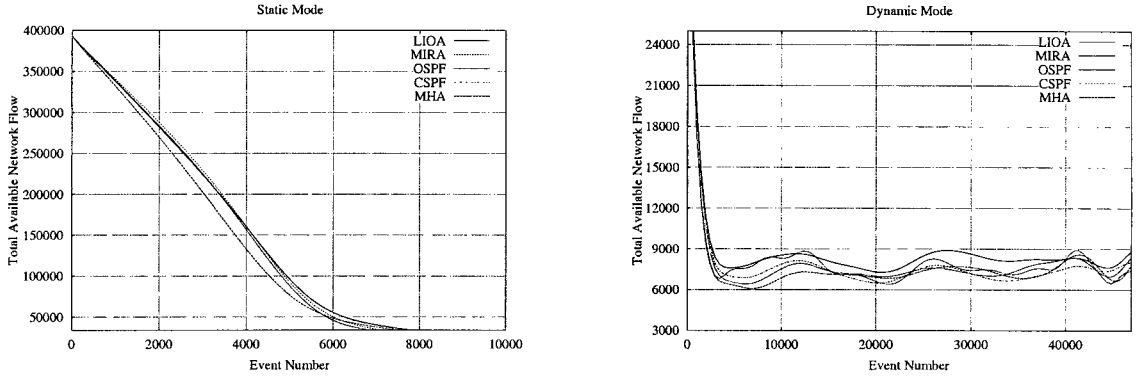

Figure 4.8: (Experiment 4.8) Total Network Flow: 12-node Network

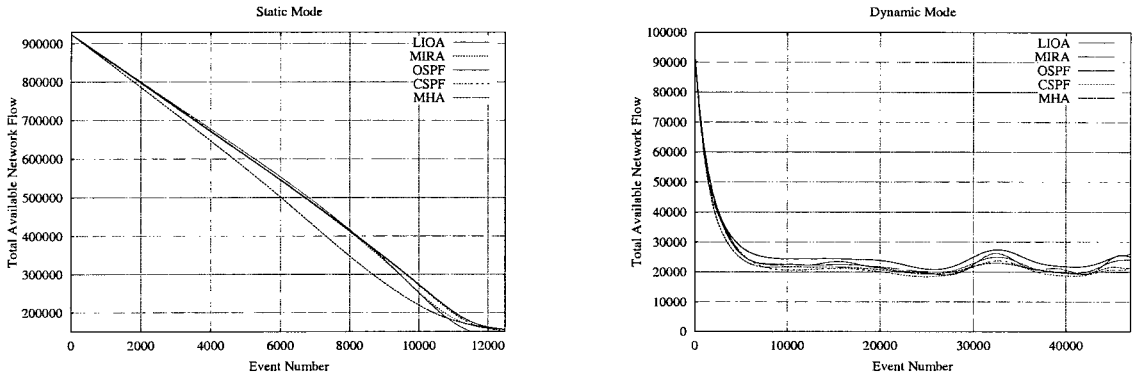

Figure 4.9: (Experiment 4.8) Total Network Flow: 15-node Network

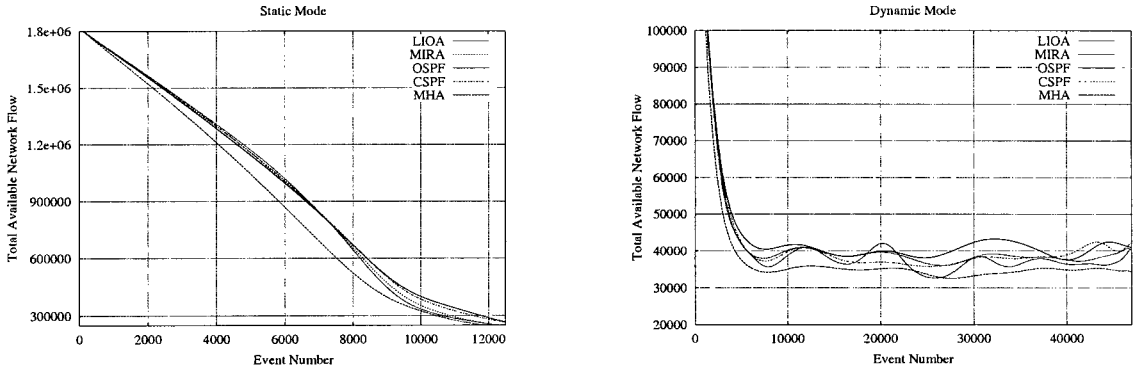

Figure 4.10: (Experiment 4.8) Total Network Flow: 23-node Network

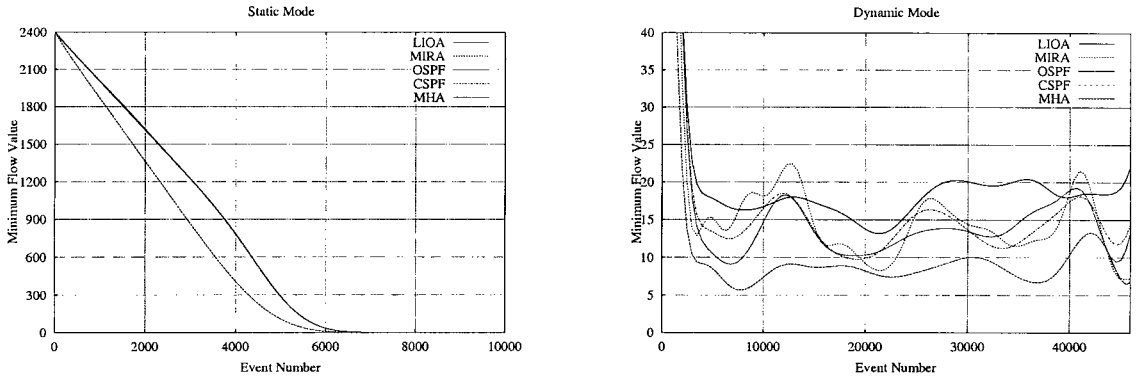

Figure 4.11: (Experiment 4.9) Minimum Flow: 12-node Network

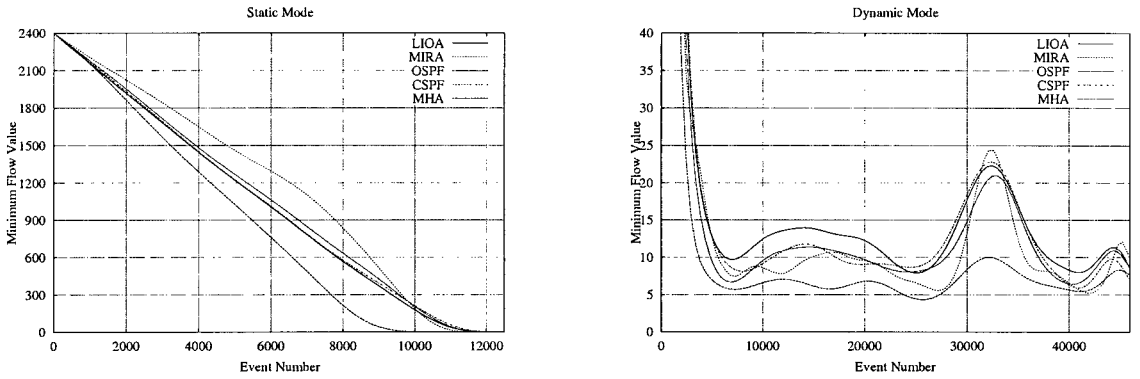

Figure 4.12: (Experiment 4.9) Minimum Flow: 15-node Network

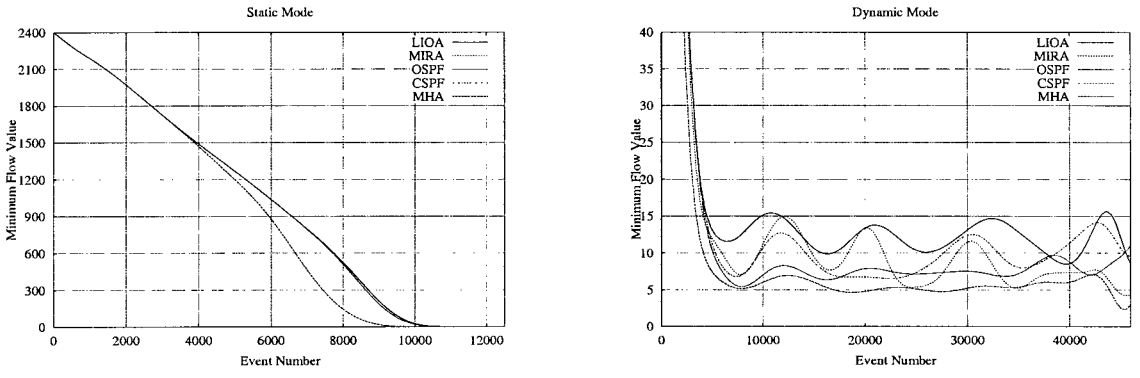

Figure 4.13: (Experiment 4.9) Minimum Flow: 23-node Network

Note that the MIRA routing strategy performs slightly better than the LIOA routing strategy at certain simulation intervals, especially in the case of static simulation in experiment 4.9. This is expected since the main objective of the MIRA routing strategy is the maximization of available flow between individual ingress-egress pairs.

4.3.4 Path Analysis

In this section we evaluate the quality and characteristics of the LSPs computed by the online routing strategies under investigation. The respective LSPs are compared with regards to performance measures such as the path length, path multiplicity, path length frequency and path preference. Path *length* refers to the number of links traversed by a path. The path *multiplicity* of an ingress-egress pair refers to the number of different paths computed for the ingress-egress pair. The path *length frequency* gives an indication of the number of times paths of a certain length were computed and the path *preference* gives an indication of how often the most popular path between an ingress-egress pair was selected. The most *popular* path refers to the path which was computed the most for an ingress-egress pair.

Let $\mathcal{P}_i$ denote the bag of all paths computed for ingress-egress pair $i \in \mathcal{I}$ and let $\mathcal{Q}_i$ denote the set of distinct paths computed for ingress-egress pair $i \in \mathcal{I}$. Then $\mathcal{P} = \bigcup_{i\in\mathcal{I}} \mathcal{P}_i$ denotes the bag of all paths computed between all ingress-egress pairs and $\mathcal{Q} = \bigcup_{i\in\mathcal{I}} \mathcal{Q}_i$ denotes the set of all distinct paths computed between all ingress-egress pairs.

Let $Q_i = \mid \mathcal{Q}_i \mid$ and $P_i = \mid \mathcal{P}_i \mid$ denote the number of distinct paths and the total number of paths computed for ingress-egress pair $i \in \mathcal{I}$ respectively. Let Q_i^k denote the number of paths of length k in the set $\mathcal{Q}_i$ and similarly let P_i^k denote the number of paths of length k in set $\mathcal{P}_i$. We define the cardinality of the empty set $\mid \emptyset \mid = 0$

We define the following path performance measures:

- The fraction of ingress-egress pairs with path multiplicity k:

$$M^k = \frac{1}{\mid \mathcal{I} \mid} \sum_{i\in\mathcal{I}:Q_i=k} 1. \tag{4.9}$$

- The fraction of distinct paths of length k:

$$H^k = \frac{1}{\mid \mathcal{Q} \mid} \sum_{i\in\mathcal{I}} Q_i^k. \tag{4.10}$$

- The fraction of times paths of length k were computed:

$$U^k = \frac{1}{\mid \mathcal{P} \mid} \sum_{i \in \mathcal{I}} P_i^k. \tag{4.11}$$

- The fraction of ingress-egress pairs using their most popular path with a frequency in the interval (a, b):

$$F_{(a,b)} = \frac{1}{\mid \mathcal{I} \mid} \sum_{i \in \mathcal{I}: a < k_i/P_i \leq b} 1 \tag{4.12}$$

  where k_i is the number of occurrences of the element occurring the most often in the bag $\mathcal{P}_i$ and k_i/P_i is the fraction of times k was computed.

We conduct various simulation experiments and analyze the resulting path sets for each of the online routing strategies under investigation.

Experiment 4.10 (Path Length) *A comparison of the path length distribution for the MHA, MIRA, OSPF, CSPF and LIOA routing strategies in both static and dynamic simulation mode.*

Experiment 4.11 (Path Length Frequency) *A comparison of the path length frequency distribution for the MHA, MIRA, OSPF, CSPF and LIOA routing strategies in both static and dynamic simulation mode.*

Experiment 4.12 (Path Multiplicity) *A comparison of the path multiplicity for the MHA, MIRA, OSPF, CSPF and LIOA routing strategies in both static and dynamic simulation mode.*

Experiment 4.13 (Path Preference) *A comparison of the path preference for the MHA, MIRA, OSPF, CSPF and LIOA routing strategies in both static and dynamic simulation mode.*

Experiments 4.10 – 4.13 are conducted using the 12-node network, the 15-node network and the 23-node network topologies. We assume that the LSP arrival rate λ_i is identical to each ingress-egress pair and the holding time μ is the same for each established LSP. The simulation environment is identical to the environment presented in 4.2 for both static and dynamic simulation mode but only a single simulation trials is considered. In the case of the LIOA routing strategy the calibration parameter α is set equal to the optimal values obtained in experiment 4.2.

The results for experiment 4.10 are presented in figures 4.14–4.16. The histograms show the number of different routes H^k of length k for $k = 1, 2, \ldots, 10$. For both the static and dynamic simulation modes the path length distributions of all the routing strategies are more or less the same and the average route length is between 2 and 4 links per path. The only noticable difference that can be observed is the fact that CSPF tends to choose paths with slightly more links than the other routing strategies.

The results for experiment 4.11 are presented in figures 4.17–4.19. The histograms show the total number of times that U^k routes of length k were computed for $k = 1, 2, \ldots, 10$. For both static and dynamic simulation modes the results for all the online routing strategies are similar.

We conclude from experiment 4.10 and 4.11 that all the online routing strategies under investigation achieve the same level of efficiency with regards to path length and path length frequency and we observe no mentionable difference in the behaviour of the routing strategies in this regard.

Figures 4.20–4.22 present the results for experiment 4.12. The histograms show the percentage of ingress-egress pairs with multiplicity M^k for $k = 1, 2, \ldots, 10$. We observe that in the case of path multiplicity there is a noticeable difference in the behaviour of the respective online routing algorithms. Note in particular the difference in behaviour between the CSPF and LIOA routing strategies: the CSPF routing strategy tends to find more distinct paths between individual ingress-egress pairs than the LIOA routing strategy.

The results for experiment 4.13 are presented in figures 4.23–4.25. In this experiment the most popular path between each ingress-egress pair is identified and the frequency of occurrence (the number of times the popular path was computed) is indicated by the histograms which shows the percentage of OD pairs $F_{(a,b)}$ for which the popular path were calculated with a frequency in the interval (a, b). For the LIOA routing strategy in static simulation mode we observe that for 55%–65% of the ingress-egress pairs the popular path was computed with a frequency between 91%–100%. A similar observation is made for the OSPF routing strategy in dynamic mode. Note that the CSPF routing strategy computes its popular path with a lower frequency than the other routing strategies.

Tables 4.21–4.23 present the total number of distinct routes discovered by each routing strat-

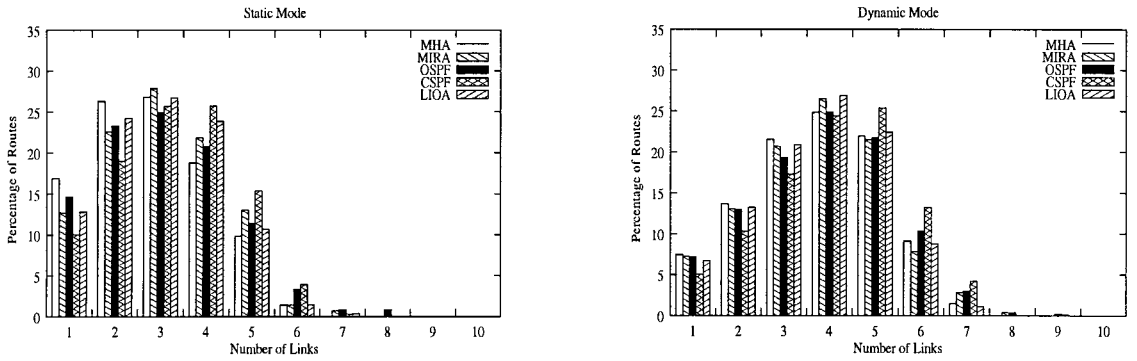

Figure 4.14: (Experiment 4.10) Path Length: 12-node Network

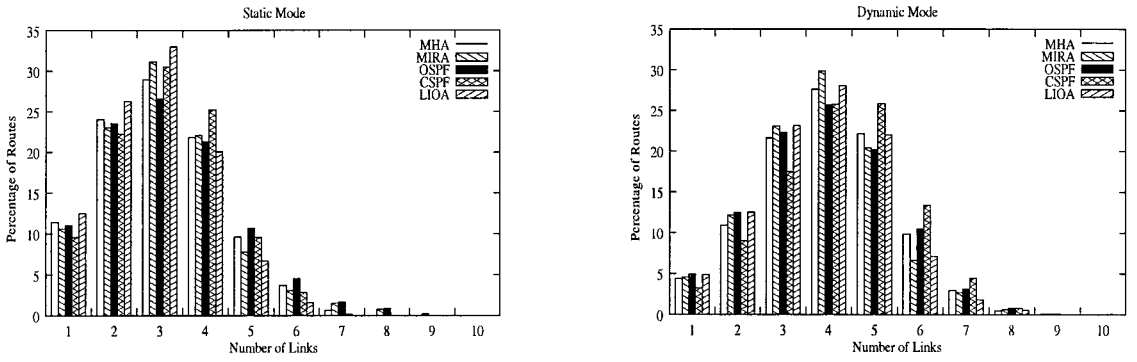

Figure 4.15: (Experiment 4.10) Path Length: 15-node Network

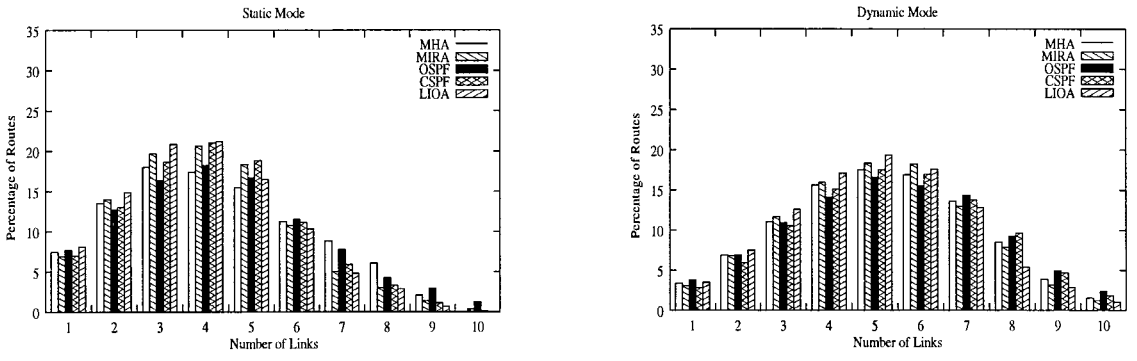

Figure 4.16: (Experiment 4.10) Path Length: 23-node Network

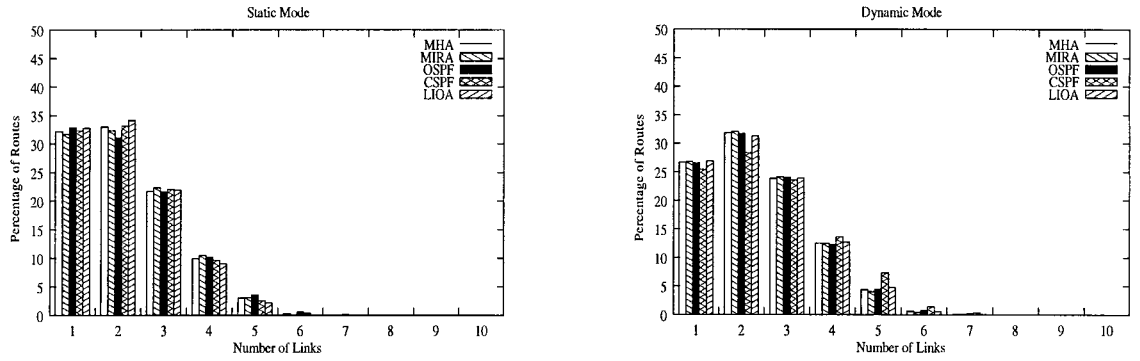

Figure 4.17: (Experiment 4.11) Path Length Frequency: 12-node Network

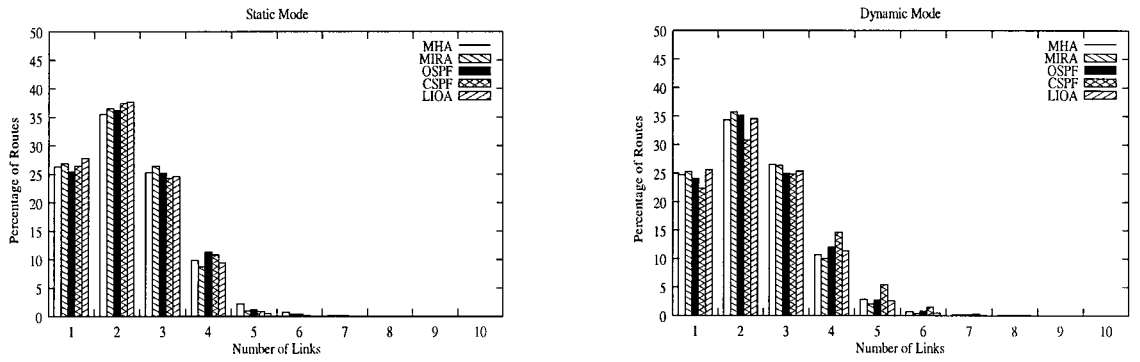

Figure 4.18: (Experiment 4.11) Path Length Frequency: 15-node Network

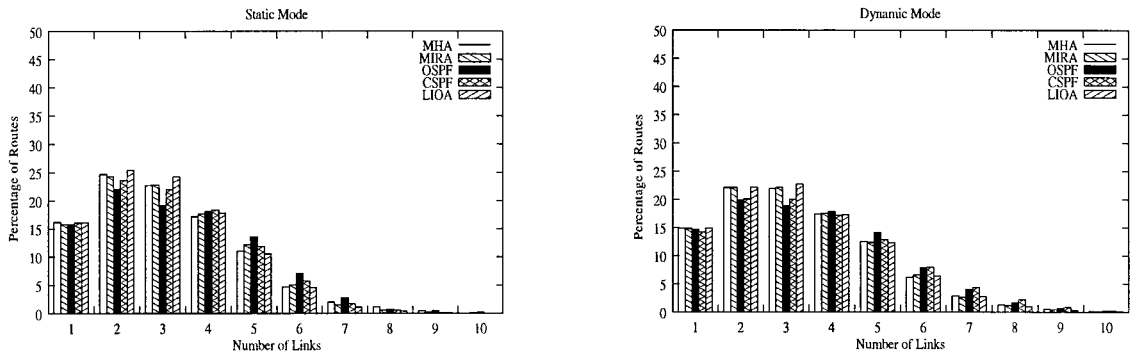

Figure 4.19: (Experiment 4.11) Path Length Frequency: 23-node Network

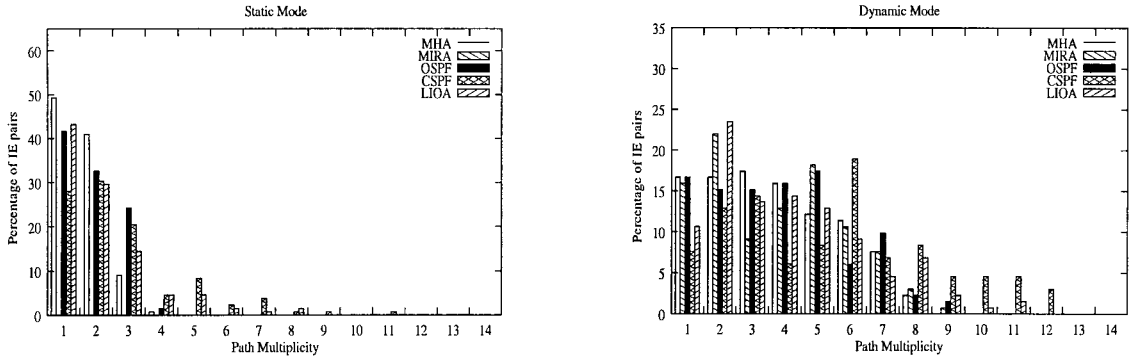

Figure 4.20: (Experiment 4.12) Path Multiplicity: 12-node Network

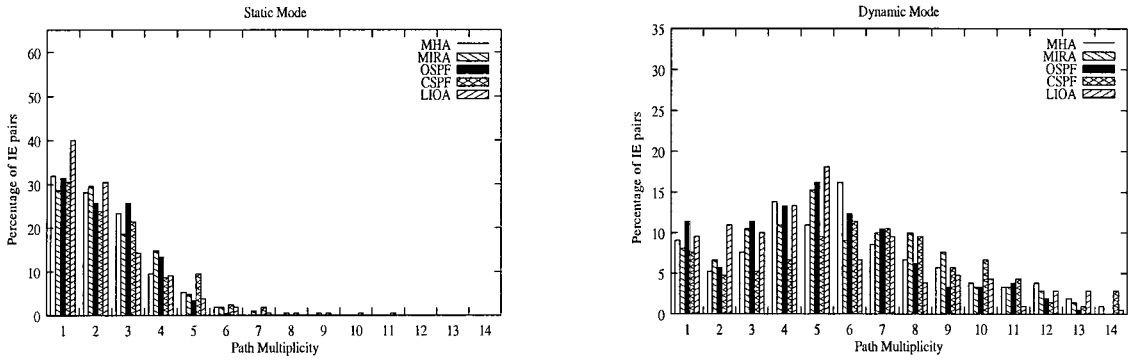

Figure 4.21: (Experiment 4.12) Path Multiplicity: 15-node Network

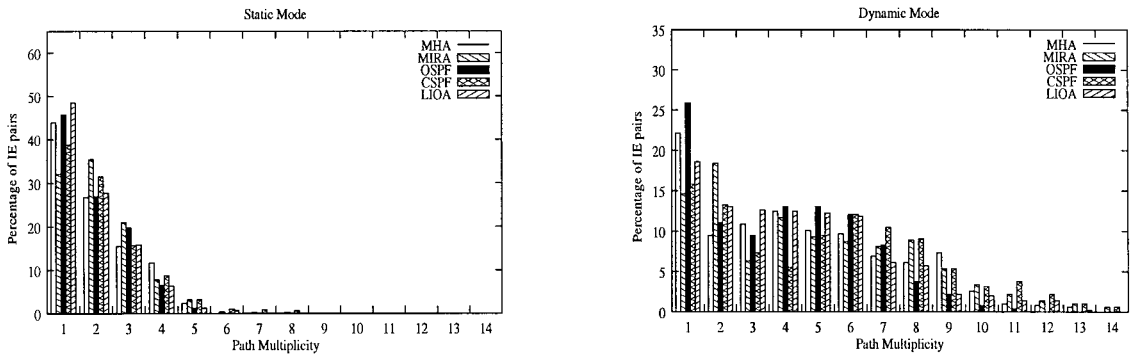

Figure 4.22: (Experiment 4.12) Path Multiplicity: 23-node Network

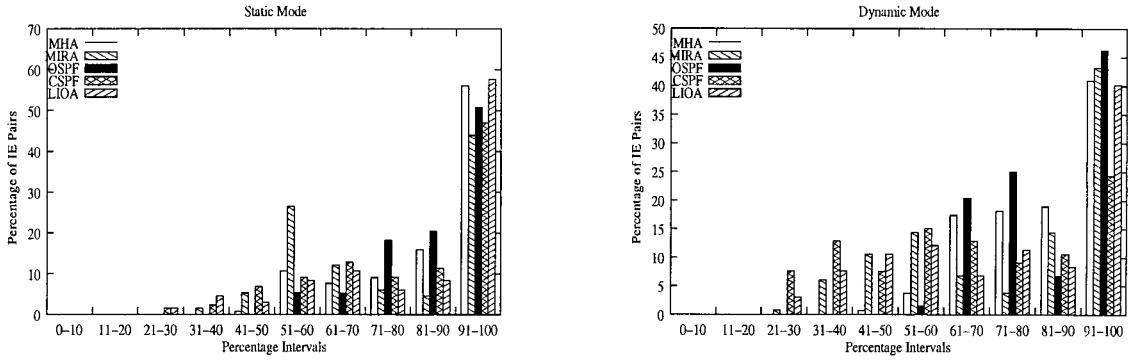

Figure 4.23: (Experiment 4.13) Usage Preference: 12-node Network

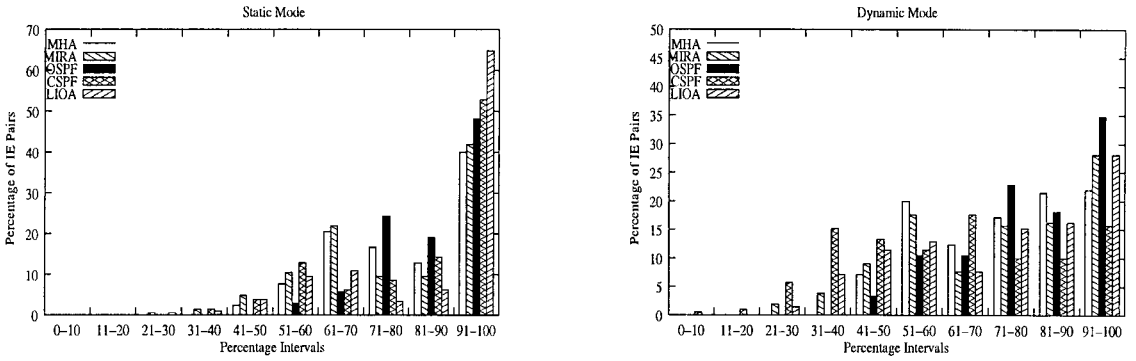

Figure 4.24: (Experiment 4.13) Path Preference: 15-node Network

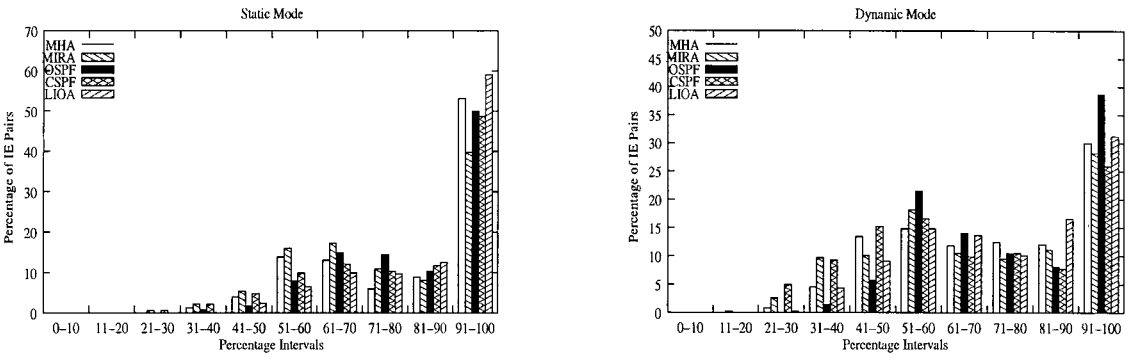

Figure 4.25: (Experiment 4.13) Path Preference: 23-node Network

egy, the average path multiplicity per ingress-egress pair and the average frequency of the popular path computation per ingress-egress pair. Note that on average the CSPF routing strategy computes the most paths between ingress-egress pairs, computes the most paths for the entire network and computes the popular path between each individual ingress-egress pair with the lowest usage frequency. These observations support the results concerning the CSPF routing strategy obtained in experiments 4.12 and 4.13 and reveal an undesirable characteristic of the latter routing scheme: a high traffic oscillation rate which means that LSPs between the same ingress-egress pair do not tend to follow the same route but rather each LSP follows a distinct route.

A high traffic oscillation rate is generally considered to be a weakness of online routing schemes [44]. Stability is an important issue for routing algorithms and a routing algorithm is expected to reach a path equilibrium in finite time, provided no regular topological or traffic changes occur. However, shortest path routing strategies such as the routing strategies investigated in this thesis, tend to oscillate traffic flows between different paths connecting an individual ingress-egress pair.

We conclude that the LIOA and OSPF routing strategies achieve the lowest oscillation rate and generally compute the least number of different paths between individual ingress-egress pairs, while the CSPF routing strategy tends to compute a large number of different paths between individual ingress-egress pairs and hence achieves a high oscillation rate.

4.4 Conclusion

This chapter compared the performance of various online routing strategies for the setup of bandwidth guaranteed LSPs in MPLS networks. The routing strategies under consideration were the Minimum Hop Algorithm (MHA), the Minimum Interference Routing Algorithm (MIRA), Open Shortest Path (OSPF) routing, Constraint Shortest Path First (CSPF) routing and the Least Interference Optimization Routing Algorithm (LIOA).

The characteristics of the LIOA link cost function were explored in order to obtain optimal settings for the calibration parameter α and the interference parameter I_ℓ. Simulation results revealed the following:

- A comparison between the two interference measures defined in equations 3.1 and 3.2 indicated that better results were obtained when I_ℓ were set equal to the interference

	Dynamic Mode			Static Mode		
	Nr of paths discovered	Avg path multiplicity	Avg popular path frequency	Nr of paths discovered	Avg path multiplicity	Avg popular path frequency
MHA	483	4	77%	213	2	79%
OSPF	502	4	78%	245	2	81%
MIRA	498	4	70%	284	2	71%
CSPF	717	5	60%	358	3	74%
LIOA	536	4	73%	281	2	80%

Table 4.21: 12-node Network: Path Statistics

	Dynamic Mode			Static Mode		
	Nr of paths discovered	Avg path multiplicity	Avg popular path frequency	Nr of paths discovered	Avg path multiplicity	Avg popular path frequency
MHA	1282	6	69%	491	2	75%
OSPF	1098	5	75%	489	2	81%
MIRA	1226	6	68%	530	3	73%
CSPF	1667	8	56%	567	3	75%
LIOA	1151	5	70%	449	2	80%

Table 4.22: 15-node Network: Path Statistics

	Dynamic Mode			Static Mode		
	Nr of paths discovered	Avg path multiplicity	Avg popular path frequency	Nr of paths discovered	Avg path multiplicity	Avg popular path frequency
MHA	2272	4	66%	1022	2	76%
OSPF	1954	4	71%	965	2	71%
MIRA	2478	5	64%	1101	2	77%
CSPF	2715	5	59%	1089	2	75%
LIOA	2177	4	70%	941	2	80%

Table 4.23: 23-node Network: Path Statistics

measure defined in equation 3.2. The latter also has the advantage of simplicity and a substantially lower computational complexity than the performance measure defined in equation 3.1 (experiment 4.1).

- Experimentation with various settings of the calibration parameter α revealed that $\alpha \geq 0.5$ achieves good results for bandwidth demand sizes with a small variation while $\alpha \leq 0.5$ achieves better performance when the bandwidth demand size variation is large (experiment 4.2).

Various simulation experiments were conducted to compare the MHA, MIRA, OSPF, CSPF and LIOA online routing strategies with regards to LSP rejection, link utilization, flow analysis and path quality. The results obtained from the experiments can be summarized as follows:

- In a static simulation environment the LIOA and CSPF routing strategies achieves the lowest LSP rejection with 14%–51% improvement over the performance of the MHA routing strategy that achieves the highest LSP rejection (experiment 4.3).
- In a dynamic simulation environment with different LSP request arrival rates between ingress-egress pairs, varying load sizes and different demand size variations the LIOA routing algorithm outperforms all the other routing strategies with regard to LSP rejection and proves to be the most consistent (experiments 4.4–4.5).
- In a dynamic simulation environment with varying LSP holding times the LIOA and CSPF routing strategies achieve the lowest LSP rejection (experiment 4.6).
- The LIOA routing strategy, followed by the CSFP routing strategy, achieve the best bandwidth utilization in both static and dynamic simulation mode (experiment 4.7).
- Flow analysis revealed that the LIOA routing strategy achieves the best bandwidth allocation in dynamic simulation mode, maximizing the amount of available network flow between the ingress-egress pairs of the network (experiments 4.8–4.9).
- Path analysis revealed that all the routing strategies behave similarly with regards to path length and path length frequency. However, the algorithms behave differently in terms of path multiplicity and path preference. Simulation experiments furthermore revealed a high path oscillation rate for the CSPF routing strategy that is generally considered to be an undesirable characteristic of shortest path routing algorithms.

We conclude that, given the results of the simulation experiments, the routing strategies LIOA, MIRA and CSPF that considers dynamic network state information perform better

than the static routing strategies MHA and OSPF that only rely on static network state information. We also observe that the high computational complexity of the MIRA routing strategy does not translate into equivalent performance gains. Although the link cost functions of the LIOA and CSPF routing strategies are closely related, simulation results reveal that the LIOA routing strategy improves on the performance of the CSPF routing strategy for all the network topologies considered and hence the inclusion of the additional cost metrics in the LIOA link cost function results in substantial performance improvements.

Chapter 5

MPLS Failure Recovery with Online Routing

In this chapter we employ the online routing algorithms under investigation in this thesis for LSP rerouting after link failure. We evaluate and compare the rerouting performance of the routing strategies by means of simulation and consider various restoration models proposed for LSP rerouting in MPLS networks. The chapter is organized as follows. Section 5.1 gives a brief overview of the rerouting methodologies proposed for failure recovery in MPLS networks. Section 5.2 presents the simulation environment and evaluates the rerouting abilities of the online routing algorithms. The findings are summarized in section 5.3.

5.1 MPLS Failure Recovery

In [31] the authors identify the main requirements that a path selection algorithm for MPLS networks must satisfy in order to be useful in practice. One of the identified requirements is the ability of the algorithm to achieve good rerouting performance upon link failure. This implies that the routing algorithm should be able to find alternative paths for as many failed LSPs as possible due to the failure or degradation of a link.

The IETF has proposed a framework for MPLS failure recovery [27] that discusses the objectives, motivation and requirements of MPLS-based recovery. The principles of MPLS recovery are presented along with comparison criteria that serve as a basis for comparing different recovery schemes. Several extensions to the MPLS signaling protocols have been proposed to support the signaling and configuration of recovery operations in MPLS networks. See for example [13, 14, 24, 25].

The MPLS recovery framework defines two basic models for MPLS recovery: *rerouting* and *protection switching*. MPLS rerouting is defined as the establishment of paths or path segments on demand for restoring network traffic after the occurrence of a link failure. Signaling is mainly used to establish the paths or path segments bypassing the failure and in general rerouting involves paths established-on-demand with resources reserved-on-demand.

Protection switching pre-computes and pre-establishes a set of recovery paths. When a fault is detected, the protected traffic for which recovery paths have been established is switched to the recovery paths. If resource reservation is required, the resources for the established recovery paths may be pre-reserved.

Various MPLS recovery schemes for both protection switching and rerouting have been proposed. See for example [7, 22, 26, 30, 36] for examples of MPLS recovery with protection switching and [19, 20, 34, 39] for examples of MPLS recovery using rerouting. Since our attention is mainly focused on online traffic engineering and the use of online routing algorithms, we will not consider protection switching but rather employ online routing algorithms for MPLS rerouting purposes.

In a topological context a distinction is made between *local repair* and *global repair*. The intent of local repair is to protect against link failure and to minimize the amount of time required to propagate fault notification signals through the network. In local repair the node immediately upstream of the failure initiates recovery and acts as the *path switch LSR (PSL)*. The PSL is responsible for switching or replicating the traffic of the failed path to the recovery path. The node responsible for receiving the recovery path traffic and either merging the traffic onto the original working path or, if the node is the egress node, passing the traffic on to the higher layer protocols, is known as the *path merge LSR (PML)*.

In global repair the PSL is usually distant from the point of failure and needs to be notified of the link failure by a *fault notification signal (FIS)*. In global repair, end-to-end path restoration applies so that traffic is rerouted from the original ingress node to the egress node. Global repair might be slower than local repair since the FIS must propagate from the point of failure to the PSL to trigger the recovery action.

For the performance evaluation of the rerouting capabilities of the online routing strategies, we will consider three topological restoration models for the recovery of failed LSPs upon link

failure: path-based restoration, link-based restoration and segment-based restoration.

Path-based Restoration

In path-based restoration traffic on a failed path is rerouted on demand along an alternative path computed from its source (the ELR). This path can be either link disjoint with the primary path or with a segment of the failed path. This recovery model is illustrated in figure 5.1. Consider an established LSP along the route $a - b - c - g$ and the failure of link $b - c$. Upon failure detection a FIS is sent to the ELR and upon arrival of the FIS an alternate path $a - d - e - f - g$ is computed along which to reroute the failed LSP. Path-based restoration is an example of a global restoration model and in this case the ELR acts as the LSR.

Link-based Restoration

Link-based restoration is a local restoration model that reroutes the traffic of a failed path by computing a bypass segment that routes traffic around the failed link. The link-based model is illustrated in figure 5.2. Again consider an established LSP along the route $a - b - c - g$ and the failure of link $b - c$. Upon detection of the failure of link $b - c$, a bypass segment $b - e - c$ is computed to replace the failed link and the failed LSP is rerouted along the path $a - b - e - c - g$. In this case node b acts as the PSL and node c acts as the PML.

Segment-based Restoration

Segment-based restoration is a local restoration model that reroutes the traffic on a failed path on a bypass path segment that replaces a portion of the failed path. Figure 5.3 illustrates the segment-based model. Consider an established LSP along the route $a - b - c - g$ and the failure of link $b - c$. Upon failure detection an alternative path segment $b - e - f - g$

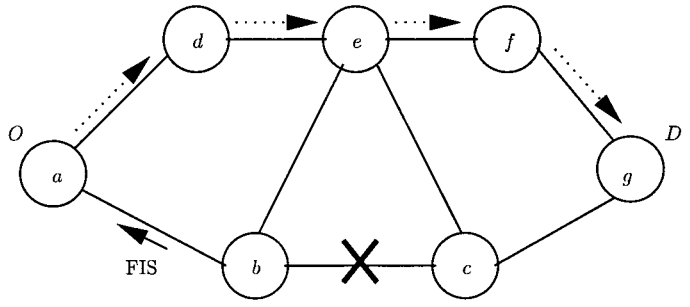

Figure 5.1: Path-based Restoration

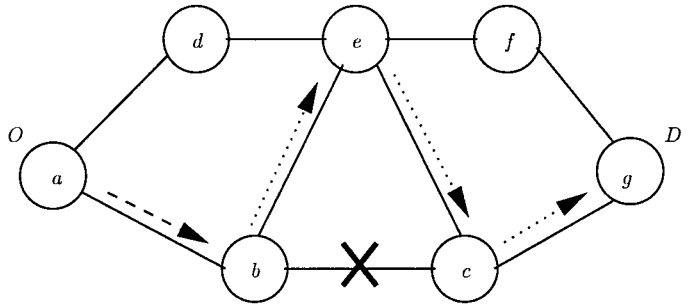

Figure 5.2: Link-based Restoration

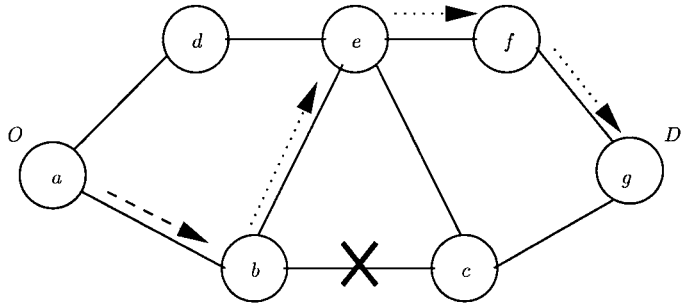

Figure 5.3: Segment-based Restoration

is computed and the failed LSP is rerouted along the path $a-b-e-f-g$. In this case node b acts as the PSL and node g acts as the PML.

5.2 Performance Evaluation

We employ the online routing strategies MHA, MIRA, OSPF, CSPF and LIOA for the rerouting of failed LSPs upon link failure, considering path-based restoration, segment-based restoration and link-based restoration. The rerouting capabilities of the online routing strategies are evaluated by means of simulation.

5.2.1 Simulation Environment

For the performance evaluation of the rerouting capabilities of the online routing strategies the three network topologies presented in chapter 4 are considered: the 12-node topology, the 15-node topology and the 23-node topology. As before each network topology is considered as a directed graph $G(\mathcal{N}, \mathcal{L})$ where $\mathcal{N}$ denotes the set of nodes, $\mathcal{L}$ denotes the set of network

links and $L =\mid \mathcal{L} \mid$ denotes the number of links in the network. The simulation process is identical to the process described in chapter 4 and all experiments are carried out in dynamic simulation mode.

To evaluate the rerouting performance of the online routing strategies upon link failure, performance measures are not only based on the failure of one selected link but rather on the individual failure of each link $\ell \in \mathcal{L}$. Thus for each trial t after e events we measure the rerouting performance of the online routing strategies by failing each of the network links sequentially and employing one of the online routing strategies to carry out either path-based restoration, link-based restoration or segment-based restoration whilst restoring the network state to the state before link failure after each individual link failure.

Let Z denote the routing strategy used to compute paths for the LSP setup requests before link failure and let Z^* denote the routing strategy employed for LSP rerouting upon link failure. Let S and s be network descriptors that describe the state of the entire network (the state of all links and LSPs) where s denotes the resulting network state after each LSP establishment, LSP tear-down operation or LSP rerouting operation. The simulation process for a single simulation trail with A LSP setup request arrivals can be described as follows:

- Simulate A LSP setup requests and employ routing strategy Z to compute a bandwidth guaranteed path for each LSP setup request.
- Save the current network state by setting $S := s$.
- For each $\ell \in \mathcal{L}$:
  - Fail link ℓ by setting $c_\ell := 0$.
  - Employ routing strategy Z^* to reroute all LSPs traversing link ℓ.
  - Restore the current network state by setting $s := S$.
  - Restore link ℓ.

5.2.2 Performance Measures

Let f_ℓ denote the number of LSPs traversing link ℓ and consider the failure of link ℓ. Let f_ℓ^a denote the number of LSPs that have been successfully rerouted and f_ℓ^r denote the number of the LSPs that could not be rerouted. Thus $f_\ell = f_\ell^a + f_\ell^r$. Let $u_\ell = r_\ell / c_\ell$ denote the link

utilization of link ℓ where r_ℓ and c_ℓ denote the reserved capacity and the total capacity of link ℓ respectively. We define the following performance metrics:

- The number of successfully rerouted LSPs for trial t:

$$F_t^a = \sum_{\ell \in \mathcal{L}} f_\ell^a. \tag{5.1}$$

- The number of failed LSPs for trial t:

$$F_t^r = \sum_{\ell \in \mathcal{L}} f_\ell^r. \tag{5.2}$$

- The percentage of successfully rerouted LSPs for trial t:

$$a_t = 100 \frac{F_t^a}{\sum_{\ell \in \mathcal{L}} f_\ell}. \tag{5.3}$$

- The average link utilization for trial t at event e:

$$u_{e,t} = \frac{1}{L} \sum_{\ell \in \mathcal{L}} u_\ell. \tag{5.4}$$

- The average number of LSPs traversing a link for trial t at event e:

$$f_{e,t} = \frac{1}{L} \sum_{\ell \in \mathcal{L}} f_\ell. \tag{5.5}$$

- The average number of successfully rerouted LSPs after T simulation trials:

$$F^a = \frac{1}{T} \sum_{t=1}^{T} F_t^a. \tag{5.6}$$

  (The following averages for T trials are defined similarly: the average percentage of rerouted LSPs a, the average number of failed LSPs F^r, the average link utilization u_e at event e and the average number of LSPs traversing a link f_e at event e.)

- The standard deviation of the link utilization for trial t at event e:

$$u_{e,t}^\sigma = \left(\frac{1}{(L-1)} \sum_{\ell \in \mathcal{L}} (u_\ell - u_{e,t})^2 \right)^{\frac{1}{2}}. \tag{5.7}$$

  (The standard deviation of the number of LSPs traversing a link $f_{e,t}^\sigma$ for trial t at event e is defined similarly. The average standard deviation of the link utilization u_e^σ and the average standard deviation of the number of LSPs traversing a link f_e^σ are defined in a similar manner as expressed in equation 5.6.)

- The improvement index IMP_z which indicates the percentage of improvement in the average percentage of successfully rerouted LSPs a of a routing strategy over strategy z where z indicates the routing strategy achieving the worst performance:

$$IMP_z = 100\frac{F^a - F_z^a}{F_z^a}. \tag{5.8}$$

- The coefficient of variation of the link utilization for T trials at event e :

$$\tau_u = 100\frac{u_e^\sigma}{u_e}. \tag{5.9}$$

(The coefficient of variation of the number of LSPs traversing a link τ_f is defined in a similar manner.)

5.2.3 Rerouting Analysis

The rerouting performance of the online routing strategies is influenced by two factors:

1. The state of the network upon link failure which includes the distribution of residual bandwidth in the network and the number of LSPs traversing each of the links. The failure of a link carrying a large number of LSPs will result in a higher rerouting demand than the failure of a link carrying a small number of LSPs. Similarly, an efficient distribution of residual bandwidth on the network links will result in better rerouting performance than in the case where some links are over-utilized while other links are under-utilized.

2. The rerouting decisions by the respective routing strategies. As in the case when paths are computed for the establishment of LSPs, the respective routing strategies will not base routing decisions on the same criteria when rerouting failed LSPs upon link failure. Hence recovery paths will differ and the routing strategies will not necessarily achieve the same rerouting performance.

To distinguish between the influence of these two factors on the rerouting performance, we consider two groups of simulation experiments – experiments carried out under *homogeneous* network conditions and experiments carried out under *non-homogeneous* network conditions. In a homogenous network environment the state of the network upon link failure is identical before each of the routing strategies is employed for rerouting. This allows for performance evaluation of the routing decisions of the respective strategies with regards to rerouting without taking the network state resulting from routing decisions for prior establishment of LSPs

into account.

In a non-homogeneous network environment the state of the network upon link failure will differ for each of the routing strategies according to the bandwidth allocation and LSP distribution that resulted from the routing decisions made by the respective routing strategies for prior establishment of LSPs. In this case not only the rerouting performance of the respective routing strategies is evaluated, but also the suitability of the network state resulting during the establishment of LSPs for MPLS recovery with rerouting.

We carry out 3 experiments to evaluate the performance of the routing strategies in homogeneous network conditions.

Experiment 5.1 *A comparative evaluation of the rerouting performance achieved by the MHA, MIRA, OSPF, CSPF and LIOA routing strategies when employed for link-based restoration after a link failure in a homogeneous network environment.*

Experiment 5.2 *A comparative evaluation of the rerouting performance achieved by the MHA, MIRA, OSPF, CSPF and LIOA routing strategies employed for segment-based restoration after a link failure in a homogeneous network environment.*

Experiment 5.3 *A comparative evaluation of the rerouting performance achieved by the MHA, MIRA, OSPF, CSPF and LIOA routing strategies when employed for path-based restoration after a link failure in a homogeneous network environment.*

	12-node Network	15-node Network	23-node Network
ω	10	10	10
λ	4	5	2
μ	1	1	1
b	[1,4]	[1,4]	[1,4]
T	30	30	30
A	10000	10000	10000

Table 5.1: Simulation Settings

Each of the experiments 5.1–5.3 are repeated for the 12-node, the 15-node and the 23-node topology. The simulation variables and traffic descriptions are given in table 5.1. We assume that the LSP arrival rate λ is the same to each ingress-egress pair and the average holding times of established LSPs are identical.

For each simulation trial we simulate the arrival of 10000 LSP setup requests and paths for the LSP requests are computed using the MHA routing strategy. After 10000 LSP request arrivals each link is failed sequentially and upon each link failure the failed LSPs are rerouted using the online routing strategy under investigation.

The results for experiments 5.1–5.3 are presented in tables 5.2–5.4. Each table denotes the average number F^a of successfully rerouted LSPs, the average number F^r of failed LSPs, the average percentage a of successfully rerouted LSPs and the improvement factor IMP_{MHA} for each of the network topologies under consideration.

The tables indicate that all the online routing strategies achieve more or less the same level of performance with an average difference of only 1% – 2% between the routing strategy achieving the worst performance level and the routing strategy achieving the best performance level.

We observe a difference in performance between each of the link-based restoration, segment-based restoration and path-based restoration methods. Consider for example the 12-node network: link-based restoration achieves a success rate $a \simeq 37\%$, segment-based restoration achieves a success rate $a \simeq 40\%$ and path-based restoration achieves a success rate $a \simeq 56\%$. Similar behaviour is observed for both the 15-node and the 23-node network topologies and thus we conclude that path- based restoration is the more efficient rerouting method achieving the highest rerouting success rate.

This result is expected: in the case of link-based restoration all LSPs are rerouted from the same LSR and thus the total capacity of links connected to the LSR sets a bound on the number of LSPs that can be rerouted. These links can be seen as bottleneck links. In the case of path-based restoration all LSPs are rerouted from their original ingress nodes, spreading the rerouting process over a larger portion of the network and avoiding the bottleneck effect occurring in link-based restoration. Note however in practice path-based restoration may incur additional time delay for the rerouting process due to the propagation delay of the fault indication signals to the ingress nodes of the LSPs affected by the link failure.

The experiments carried out under homogeneous network conditions indicate that the routing decisions made by each of the online routing strategies for rerouting purposes results in a minor difference in the performance achieved by the various strategies. To explore the suitability and efficiency of the bandwidth allocation achieved by each of the online routing strategies due to the establishment of LSPs prior to link failure, we conduct three experiments in a

Network	Method	F^r	F^a	a	IMP_{MHA}
12-node network	MHA	769	446	36.83	0
	OSPF	769	446	36.83	0
	MIRA	769	446	36.83	0
	CSPF	768	447	36.96	1
	LIOA	766	449	37.09	1
15-node network	MHA	1433	981	40.78	0
	MIRA	1432	982	40.79	0
	OSPF	1432	982	40.79	0
	CSPF	1421	993	41.26	1
	LIOA	1427	987	41.03	1
23-node network	MHA	1805	1357	42.99	0
	MIRA	1801	1361	43.08	0
	OSPF	1805	1357	42.99	0
	CSPF	1798	1364	43.21	1
	LIOA	1791	1371	43.42	1

Table 5.2: (Experiment 5.1) Homogeneous case: Link-rerouting.

Network	Method	F^r	F^a	a	IMP_{MHA}
12-node network	MHA	730	486	40.12	0
	MIRA	730	486	40.12	0
	OSPF	729	487	40.20	0
	CSPF	725	491	40.53	1
	LIOA	726	490	40.47	1
15-node network	MHA	1333	1081	44.97	0
	MIRA	1326	1088	45.21	0
	OSPF	1328	1086	45.18	0
	CSPF	1320	1094	45.51	1
	LIOA	1314	1100	45.77	1
23-node network	MHA	1805	1358	42.99	0
	MIRA	1800	1363	43.09	0
	OSPF	1799	1364	43.18	0
	CSPF	1790	1373	43.46	1
	LIOA	1784	1379	43.63	1

Table 5.3: (Experiment 5.2) Homogeneous case: Segment-rerouting.

Network	Method	F^r	F^a	a	IMP_{MHA}
12-node network	MHA	537	678	55.96	0
	MIRA	535	680	56.11	0
	OSPF	535	680	56.11	0
	CSPF	530	685	56.58	1
	LIOA	529	686	56.61	1
15-node network	MHA	875	1539	63.91	0
	MIRA	863	1551	64.17	0
	OSPF	862	1552	64.42	1
	CSPF	848	1566	65.01	1
	LIOA	842	1572	65.26	1
23-node network	MHA	1193	1969	62.30	0
	MIRA	1179	1983	62.24	0
	OSPF	1183	1979	62.63	0
	CSPF	1168	1994	63.10	1
	LIOA	1161	2001	63.33	1

Table 5.4: (Experiment 5.3) Homogeneous case: Path-rerouting.

Network	Method	F^r	F^a	a	IMP_{MHA}
12-node network	MHA	769	446	36.83	0
	OSPF	747	486	39.59	7
	MIRA	734	489	40.07	9
	CSPF	784	542	41.08	12
	LIOA	688	553	44.83	22
15-node network	MHA	1433	981	40.78	0
	MIRA	1215	1114	47.83	17
	OSPF	1154	1299	53.19	30
	CSPF	1261	1412	53.09	30
	LIOA	1149	1294	53.25	30
23-node network	MHA	1805	1357	42.99	0
	MIRA	1894	1503	44.24	3
	OSPF	1909	1501	44.09	3
	CSPF	1911	1570	45.19	5
	LIOA	1702	1534	47.48	10

Table 5.5: (Experiment 5.4) Non-homogeneous case: Link-rerouting.

Network	Method	F^r	F^a	a	IMP_{MHA}
12-node network	MHA	730	486	40.12	0
	MIRA	696	527	43.18	8
	OSPF	725	509	41.44	3
	CSPF	757	570	43.17	8
	LIOA	671	571	46.23	13
15-node network	MHA	1333	1081	44.97	0
	MIRA	1148	1182	50.74	13
	OSPF	1226	1228	50.32	12
	CSPF	1319	1354	50.99	13
	LIOA	1154	1288	53.10	18
23-node network	MHA	1805	1358	42.99	0
	MIRA	1787	1498	45.60	6
	OSPF	1805	1360	42.96	0
	CSPF	1936	1545	44.46	3
	LIOA	1729	1507	46.67	9

Table 5.6: (Experiment 5.5) Non-homogeneous case: Segment-rerouting.

Network	Method	F^r	F^a	a	IMP_{MHA}
12-node network	MHA	537	678	55.96	0
	MIRA	536	687	56.20	0
	OSPF	522	711	57.84	3
	CSPF	531	795	60.08	7
	LIOA	510	731	59.09	6
15-node network	MHA	875	1539	63.91	0
	OSPF	883	1571	64.22	1
	MIRA	828	1501	64.45	0
	CSPF	848	1825	68.43	7
	LIOA	866	1576	64.77	2
23-node network	MHA	1193	1969	62.30	0
	MIRA	1271	2149	62.84	0
	OSPF	1260	2150	63.11	1
	CSPF	1281	2199	63.24	1
	LIOA	1135	2101	64.39	3

Table 5.7: (Experiment 5.6) Non-homogeneous case: Path-rerouting.

non-homogeneous network environment.

Experiment 5.4 *A comparative evaluation of the rerouting performance achieved by the MHA, MIRA, OSPF, CSPF and LIOA routing strategies when employed for link-based restoration after a link failure in a non-homogeneous network environment.*

Experiment 5.5 *A comparative evaluation of the rerouting performance achieved by the MHA, MIRA, OSPF, CSPF and LIOA routing strategies when employed for segment-based restoration after a link failure in a non-homogeneous network environment.*

Experiment 5.6 *A comparative evaluation of the rerouting performance achieved by the MHA, MIRA, OSPF, CSPF and LIOA routing strategies when employed for path-based restoration after a link failure in a non-homogeneous network environment.*

Experiments 5.4–5.6 are conducted in a similar simulation environment as was introduced for experiments 5.1–5.3 and each experiment is repeated for the 12-node topology, the 15-node topology and the 23-node topology. The simulation variables and traffic description are identical to those presented in table 5.1. We assume that the LSP arrival rate λ_i is the same to each ingress-egress pair and the average holding times of all established LSPs are identical.

For each simulation trial we simulate the arrival of 10000 LSP setup requests and paths for the LSP requests are computed with the online routing strategy under investigation. After 10000 LSP request arrivals each link is failed sequentially and upon each link failure the failed LSPs are rerouted using the same online routing strategy as was used to compute LSPs for the LSP setup requests.

The results for experiments 5.1–5.3 are presented in tables 5.5–5.7 and as before each table denote the average number F^a of successfully rerouted LSPs, the average number F^r of failed LSPs, the average percentage a of successfully rerouted LSPs and the improvement factor IMP_{MHA} for each of the network topologies under consideration.

Note that since the experiments were carried out in a non-homogeneous network environment, the respective online routing strategies do not initially have the same number of connections on each link to reroute. This can be attributed to two factors: firstly, as observed in chapter 4, the routing strategies do not accept the same number of LSPs setup requests and hence there is not the same number of LSPs in the network. Since the LIOA routing strategy generally accepts more LSPs than the other routing strategies, we will expect the network where LSPs were computed with LIOA to contain the most LSPs and thus have the most LSPs to

reroute upon link failure. Secondly, since each routing strategy computes a distinct path set, the distribution of LSPs on the network links differs. Longer paths calculated for LSPs will result in more LSPs traversing each network link and hence more LSPs to reroute after link failure.

The results obtained for experiments 5.4–5.6 conducted in a non-homogeneous network environment show a notable difference in the rerouting performance achieved by the individual routing strategies. In most cases we observe that the LIOA routing strategy successfully reroutes the highest percentage of LSPs with an improvement of up to 30% over the MHA routing strategy which achieves the lowest success rate.

The observations made in the experiments carried out under homogenous network conditions indicate that the performance of the online routing strategies after link failure are more or less identical and can thus not be responsible for the performance difference observed for non-homogenous network conditions. This performance difference is therefore attributed to the distinct network state established by each of the online routing strategies prior to link failure and is influenced by the following two factors:

- the distribution of residual link capacity on the links, and
- the distribution of LSPs on the links.

We study the influence of these factors by conducting an examination of the state of the network links before link failure occurs, thus after the arrival of 10000 LSP setup requests. In particular we pay attention to the bandwidth utilization u_ℓ and the number of LSPs f_ℓ traversing each link $\ell \in \mathcal{L}$. Table 5.8 presents the average link utilization u_A, the average standard deviation u_A^σ of the link utilization between the individual network links , the average number f_A of LSPs traversing a link and the average standard deviation f_A^σ of the number of LSPs traversing each individual network link after T trials. Note that $A = 10000$ represents the number of LSP setup request arrivals.

Intuitively we expect a lower value of u_A and a lower value of f_A to favour the rerouting performance of the routing strategies after link failure. u_A^σ and f_A^σ gives an indication of the spread of u_A and f_A throughout the network respectively. Low values for u_A^σ and f_A^σ are favoured since they indicate a more uniform state of the network links.

When examining the values in table 5.8 it is important not to compare the various rout-

Network	Method	u_A	u_A^σ	τ_u	f_A	f_A^σ	τ_f
12-node network	MHA	33.76	3.71	11	64	6.56	10
	OSPF	34.26	3.64	11	63	6.01	9
	MIRA	33.97	3.77	11	64	6.71	10
	CSPF	36.85	2.60	7	65	3.52	5
	LIOA	34.49	2.30	6	63	3.40	5
15-node network	MHA	43.11	0.73	2	74	1.82	2
	OSPF	43.81	0.55	1	70	2.62	4
	MIRA	41.60	0.38	1	72	1.78	2
	CSPF	47.73	0.66	1	73	2.37	3
	LIOA	43.62	0.52	1	71	2.59	3
23-node network	MHA	41.61	2.10	5	67	2.13	3
	OSPF	44.86	1.21	3	63	1.38	2
	MIRA	43.47	2.01	5	67	2.17	3
	CSPF	45.80	2.13	5	66	1.31	2
	LIOA	42.58	2.06	5	64	1.01	1

Table 5.8: Network State before Rerouting

ing strategies based on a single performance measure. For example even though a low u_A indicates more available bandwidth, in a certain case a high standard deviation for the same case indicates that this bandwidth is not spread uniformly throughout the network and thus may only favour the failure of certain network links. We therefore consider the coefficients of variation τ_u and τ_f as a measure of variability in order to compare the routing strategies, where a lower coefficient of variance indicates a lower variability.

From table 5.8 we conclude that the LIOA and CSPF routing strategies achieves an LSP and bandwidth distribution more suited for rerouting purposes.

5.3 Conclusion

In this chapter we employed the online routing algorithms for LSP recovery upon link failure. We evaluated and compared the rerouting ability of the online routing strategies by means of simulation and considered three restoration models proposed for LSP rerouting in MPLS networks: path-based restoration, link-based restoration and segment-based restoration.

We measured the rerouting performance of the online routing strategies by sequentially failing each link of the network topologies under consideration and employing each of the online routing strategies to reroute the affected LSPs upon link failure. The simulation experiments

were carried out in both homogeneous and non-homogenous network environments. The simulation results indicated the following:

- A significant difference in the rerouting performance is observed for link-based restoration, segment-based restoration and path-based restoration respectively and from these results we concluded that path-based restoration is the most efficient rerouting method achieving the highest rerouting success rate.
- For homogenous network conditions the performance of the online routing strategies after link failure is more or less identical with an average performance difference of only 1% – 2% between the routing strategy achieving the worst performance and the routing strategy achieving the best performance.
- The results obtained from simulation experiments conducted in a non-homogeneous network environment show a notable difference in the rerouting performance achieved by the individual routing strategies. In most cases we observe that the LIOA routing strategy successfully reroutes the highest percentage of LSPs with an improvement of up to 30% over the MHA routing strategy which achieves the lowest success rate.
- After exploring the link utilization and link flow distribution, we concluded that the LIOA and CSPF routing strategies achieve an LSP and bandwidth allocation the most suitable for rerouting purposes.

Chapter 6

Conclusion

6.1 Thesis Summary

The problem addressed in this thesis is the sequential establishment of bandwidth guaranteed LSPs in MPLS networks by means of online routing. The online routing algorithms considered in this thesis include the popular shortest path routing algorithms often deployed for intra-domain routing such as the Minimum Hop Algorithm (MHA), Open Shortest Path First (OSPF) routing and Constraint Shortest Path First (CSPF) routing. We also considered an online routing algorithm recently proposed for the setup of bandwidth guaranteed LSPs known as the Minimum Interference Routing Algorithm (MIRA).

We proposed an online routing framework known as Least Interference Optimization (LIO) that utilizes the current bandwidth availability and traffic flow distribution to achieve efficient traffic engineering. The Least Interference Optimization Algorithm (LIOA) was presented for the setup of bandwidth guaranteed LSPs in MPLS networks.

We carried out various simulation experiments to evaluate and compare the online routing strategies under investigation. Experimentation includes the evaluation of the cost metrics defined in link cost function of the LIO framework and the comparison of the online routing strategies with regard to LSP request rejection, link utilization, flow distribution and path quality. Simulation results revealed that, for the network topologies under consideration, the routing strategies that employed dynamic network state information in their routing decisions (LIOA, CSPF and MIRA) generally outperformed the routing strategies that only rely on static network information (OSPF and MHA). In most simulation experiments the best performance was achieved by the LIOA routing strategy while the MHA performed the worse.

Furthermore we observed that the computational complexity of the MIRA routing strategy does not translate into equivalent performance gains.

We employed the online routing strategies for MPLS failure recovery upon link failure. In particular we investigated two aspects to determine the efficiency of the routing strategies for MPLS rerouting: the suitability of the LSP configuration that results due to the establishment of LSPs prior to link failure and the ability of the online routing strategy to reroute failed LSPs upon link failure. Simulation results revealed similar rerouting performance for all online routing strategies under investigation, but a LSP configuration most suitable for online rerouting was observed for the LIOA routing strategy.

6.2 Future Work

- In the LIO framework an interference parameter is defined in the link cost function and in this thesis two interference measures were proposed as possible values for the interference parameter: an interference measure indicating the decrease in available network flow due to the establishment of LSPs (as defined by the MIRA routing strategy) and an interference measure indicating the number of LSPs traversing a link. The interference parameter plays an important role in the performance of the LIOA routing strategy and it might therefore be useful to experiment with alternative interference measures.

- The LIO framework defines both a multiplicative link cost function and an additive link cost function. In this thesis we experimented mainly with the multiplicative link cost function. Experimentation with the additive link cost function and the addition of additional cost metrics has been reserved for future work.

- In this thesis we assumed that the average arrival rates of LSP setup requests to the individual ingress-egress pairs of the network remained constant for the full duration of each simulation experiment. However, time varying arrival rates, as is often observed in network topologies that span more than one time zone, will provide more insight on the consistency and the adaptability of the online routing algorithms under investigation and the effectiveness of the cost metrics defined by the LIO framework.

Appendix A

Network Topologies

This appendix contains graphical representations of the three network topologies used in this thesis for the evaluation and performance comparison of the online routing strategies under investigation. In all network topology representations the thicker lines indicate OC-48 links while the thinner lines indicate OC-12 links with 48 and 12 units of bandwidth respectively. Note that each line represents two uni-directional links of equal size.

The 12-node Network Topology

The 12-node network was obtained from [11] and represents a commercial backbone topology of the USA. A graphical representation of the 12-node network topology is given in figure A.1.

The 15-node Network Topology

The 15-node network is identical to the network topology presented in [31] and was chosen due to its popularity for online routing performance studies (see for example [41, 43, 11]). A graphical representation of the 15-node network topology is given in figure A.2.

The 23-node Network Topology

The 23-node network is fictitious representation of a backbone topology for the USA. A graphical representation of the 12-node network topology is given in figure A.3.

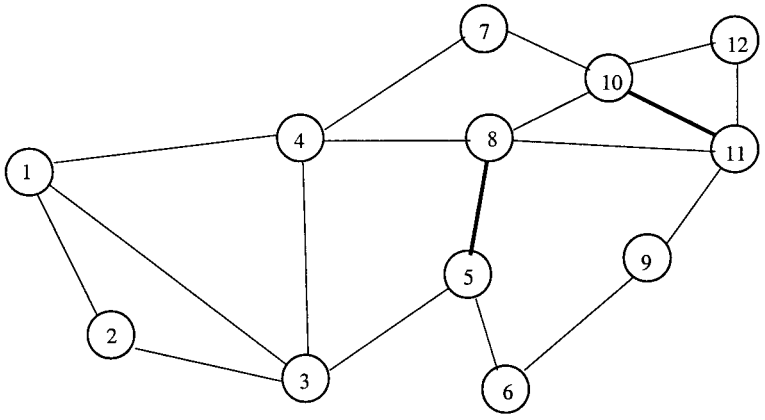

Figure A.1: 12-node Network Topology.

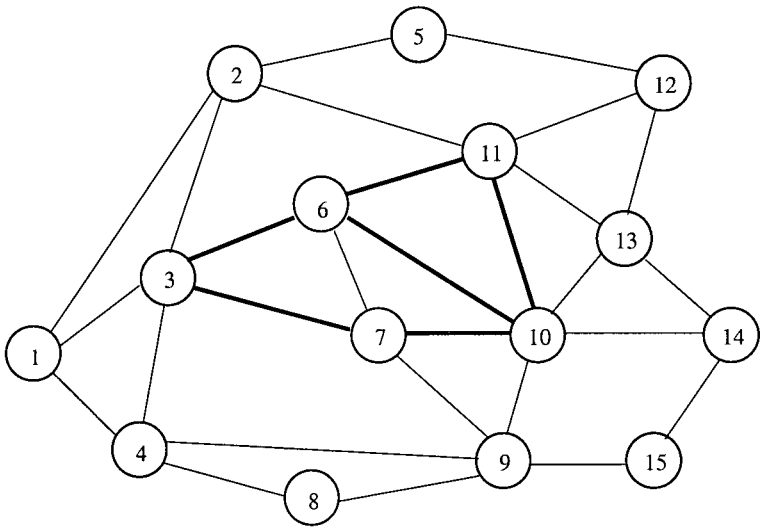

Figure A.2: 15-node Network Topology.

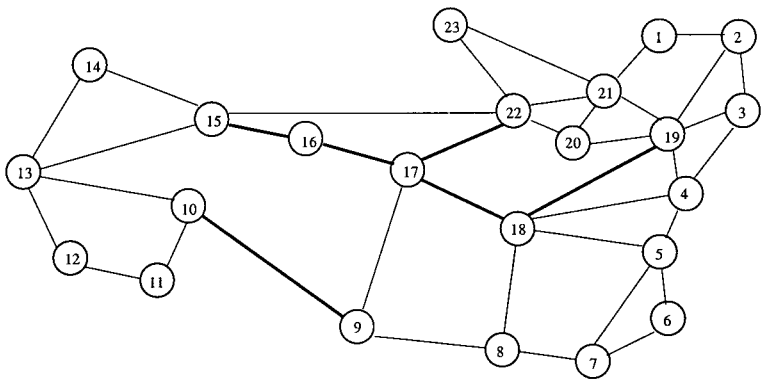

Figure A.3: 23-node Network-Topology.

Appendix B

The Simulation Model

This appendix describes a simulation model that models the arrival and departure of connection-level events according to a Markov chain under both time constant and time varying traffic conditions.

B.1 The Markov Chain Simulator

Let $\mathcal{I}$ denote the set of ingress-egress pairs in the network. LSP setup requests arrive to ingress-egress pair $i \in \mathcal{I}$ according to a Poisson process with average arrival rate λ_i and the LSP holding times are exponentially distributed with parameter $1/\mu_i$. Let m_i denote the number of active LSPs connecting ingress-egress pair $i \in \mathcal{I}$. The total arrival rate to the network is denoted by $\Lambda = \sum_{i \in \mathcal{I}} \lambda_i$.

Connection-level events, which can either be an LSP setup request arrival or an LSP tear-down operation, are generated as follows:

Let

$$\Gamma = \sum_{i \in \mathcal{I}} (\lambda_i + m_i \mu_i). \tag{B.1}$$

Generate a random variable U uniformly distributed on the interval $[0, 1]$. If

$$\Gamma U \leq \Lambda \tag{B.2}$$

then this event is an LSP setup request which is modelled as follows:

- Generate a random variable U^* uniformly distributed on the interval $[0, 1]$. Then this LSP setup request arrives to ingress-egress pair I such that

$$\sum_{i=1}^{I-1} \lambda_i < \Lambda U^* < \sum_{i=1}^{I} \lambda_i. \tag{B.3}$$

- Compute a feasible path along which this LSP will be routed. If such a path exists the LSP is accepted and $m_I := m_I + 1$.

Alternatively, an LSP is torn down between ingress-egress pair I such that

$$\sum_{i \in \mathcal{I}} \lambda_i + \sum_{i=1}^{I-1} m_i \mu_i < \Gamma U < \sum_{i=1}^{I} m_i \mu_i + \sum_{i \in \mathcal{I}} \lambda_i \tag{B.4}$$

and the event is modeled as follows:

- Generate a random variable U uniformly distributed on the interval $[0, 1]$ and let $k = \lceil U m_I \rceil$. Then kth LSP connecting ingress-egress pair I is torn down and set $m_I := m_I - 1$.

B.1.1 Remarks on Implementation

- The simulation model assumes that the holding times of all LSPs connecting ingress-egress pair $i \in \mathcal{I}$ have the same exponentially distributed holding time with parameter $1/\mu_i$. However, this is not essential and the simulation model can be modified to relax this assumption.

- When comparing the performance of two independent simulations with different routing strategies, the uniformly distributed random variables U and U^* should be generated using two independent random number generators. This is to assure the same sequence of LSP request arrivals to both simulations.

B.2 Extension to the Markov Chain Simulator for Time Varying Traffic Conditions

The simulation model described in B.1 assumes that the average arrival rate v_i of LSP requests to ingress-egress pair i remains constant throughout the simulation. However it is possible to extend the simulation model to generate connection level events for the case where the arrival rates vary continuously according to the simulation time.

Let $v_i(t)$ denote the average arrival rate to ingress-egress pair $i \in \mathcal{I}$ at time t and let T be the current simulation time (initially T = 0). For each simulation event, generate an exponentially distributed random time t with parameter

$$\Gamma_T = \sum_{i \in \mathcal{I}} (v_i(T) + m_i \mu_i) \tag{B.5}$$

and set $T := t$. Generate the event as described in B.1, but by replacing the constant arrival rate v_i with the $v_i(T)$ in equations B.1, B.2, B.3 and B.4.

Appendix C

Acquiring Confidence Intervals

Assume M independent replications of a terminating simulation where the independence of the replications is accomplished by using different random number streams for each replication. Let $X_1, X_2, , X_M$ be a sequence of independent identical distributed random variables where X_i is defined as the observation of the chosen performance measure on the ith replication. The mean $\bar{X}$ of the observations is

$$\bar{X} = \frac{1}{M}\sum_{i=1}^{M} X_i$$

and an estimation s^2 of the variation of the observations is given by

$$s^2 = \frac{1}{M-1}\sum_{i=1}^{M}(\bar{X} - X_i)^2.$$

Let $t_{v,\alpha}$ denote the critical point of Student's t-distribution with v degrees of freedom and probability density function

$$f_v(t) = \frac{(\frac{v}{v+x^2})^{(1+v)/2}}{\sqrt{v}B(\frac{1}{2}v, \frac{1}{2}}$$

and define

$$\delta(M, \alpha) = t_{M-1,\alpha/2}\sqrt{s^2/M}.$$

Then $\bar{X}$ is an unbiased point estimate for $\mu = E(X)$ with a confidence level of $100(1-\alpha)\%$ and relative error γ if

$$M \geq \left(\frac{t_{M-1,1-\alpha/2}s}{\bar{X}\gamma}\right)^2$$

where $\gamma = |\bar{X} - \mu|/|\mu|$ and the width of the confidence interval is given by

$$I(\alpha,\gamma) = \left(\bar{X} - \delta(M,\alpha), \bar{X} + \delta(M,\alpha)\right).$$

Bibliography

[1] Cisco Systems. Technical report. http://www.cisco.com.

[2] IBM. Technical report. http://www.ibm.com.

[3] Ipsilon Networks. Technical report. http://www.ipsilon.com.

[4] Toshiba. Technical report. http://www.toshiba.com.

[5] Multiprotocol Label Switching White Paper. Technical report, Future Software Private Limited, India, 1999.

[6] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification. RFC 2205, Internet Engineering Taskforce, September 1997.

[7] T. Cinkler, P. Laborczi, and A. Horvath. Protection through Thrifty Configuration. In *Proceedings of IEEE Infocom*, pages 975 – 987, July 2002.

[8] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms.* The MIT Press and McGraw-Hill Book Company, 1997.

[9] E. Crawley, R. Nair, B. Jajagopalan, and H. Sandick. A Framework for QoS-based Routing in the Internet. RFC 2386, Internet Engineering Taskforce, August 1998.

[10] B. Davie, P. Doolan, and Y. Rekhter. *Switching in IP Networks: IP Switching, Tag Switching and Related Technologies.* Morgan Kaufmann Publishers, 1998.

[11] J.C. de Oliveira, F. Martinelli, and C. Scoglio. SPeCRA: A Stochastic Performance Comparison Routing Algorithm for LSP Setup in MPLS Networks. In *Proceedings of IEEE GLOBECOM 2002*, Taipei, Taiwan, November 2002.

[12] D. Eppstein. Finding the k shortest paths. *SIAM J. Computing*, 28(2):652 – 673, 1998.

[13] A. Iwata et al. Crankback Routing Extensions for MPLS Signaling. Internet draft, work in progress draft-iwata-mpls-crankback-01.txt, Internet Engineering Taskforce, July 2001.

[14] A. Iwata et al. MPLS Signaling Extensions for Shared Fast Rerouting. Internet draft, work in progress draft-iwata-mpls-shared-fastreroute-00.txt, Internet Engineering Taskforce, July 2001.

[15] D. Awduche et al. Requirements for Traffic Engineering Over MPLS. RFC 2702, Internet Engineering Taskforce, September 1999.

[16] D. Awduche et al. Applicability Statement for Extensions to RSVP for LSP Tunnels. Internet draft, work in progress draft-ietf-mpls-rsvp-tunnel-applicability-02.txt, Internet Engineering Taskforce, April 2001.

[17] D. Awduche et al. Overview and Principles of Internet Traffic Engineering. Internet draft, work in progress draft-ietf-tewg-principles-00.txt, Internet Engineering Taskforce, August 2001.

[18] D. Awduche et al. RSVP-TE: Extensions to RSVP for LSP Tunnels. Internet draft, work in progress draft-ietf-mpls-rsvp-lsp-tunnel-09.txt, Internet Engineering Taskforce, August 2001.

[19] D. Gan et al. A Method for MPLS LSP Fast-Reroute Using RSVP Detours. Internet draft, work in progress draft-gan-fast-reroute-00.txt, Internet Engineering Taskforce, April 2001.

[20] D. Haskin et al. A Method for Setting Alternative Label Switched Paths to Handle Fast Reroute. Internet draft, work in progress draft-haskin-mpls-fast-reroute-05.txt, Internet Engineering Taskforce, November 2000.

[21] J. Ash et al. Modification Using CR-LDP. Internet draft, work in progress draft-ietf-mpls-crlsp-modify-03.txt, Internet Engineering Taskforce, March 2001.

[22] K. Owens et al. A Path Protection/Restoration Mechanism for MPLS Networks. Internet draft, work in progress draft-chang-mpls-path-protection-03.txt, Internet Engineering Taskforce, July 2001.

[23] L. Andersson et al. LDP Specification. RFC 3036, Internet Engineering Taskforce, January 2001.

[24] P. Pan et al. Fast Reroute Extensions to RSVP-TE for LSP Tunnels. Internet draft, work in progress draft-ietf-mpls-rsvp-lsp-fastreroute-01.txt, Internet Engineering Taskforce, May 2003.

[25] R. Cetin et al. MPLS Traffic Engineering Management Information Base for Fast Reroute. Internet draft, work in progress draft-ietf-mpls-fastreroute-mib-01.txt, Internet Engineering Taskforce, November 2002.

[26] S. Kini et al. Shared Backup Label Switched Path Restoration. Internet draft, work in progress draft-kini-restoration-shared-backup-01.txt, Internet Engineering Taskforce, May 2001.

[27] V. Sharma et al. Framework for MPLS-based Recovery. Internet draft, work in progress draft-ietf-mpls-recovery-frmwrk-08.txt, Internet Engineering Taskforce, October 2002.

[28] B. Fortz and M. Thorup. Internet Traffic Engineering by Optimizing OSPF Weights. In *Proceedings of IEEE Infocom*, pages 519 – 528, March 2000.

[29] R. Guerin, D. Williams, and A. Orda. QoS Routing Mechanisms and OSPF Extensions. In *Proceedings of 2nd Global Internet Miniconference*, pages 1914 – 1918, Phoenix, Arizona, November 1997.

[30] C. Huang, V. Sharma, K. Owens, and S. Makam. Building Reliable MPLS Networks using a Path Protection Mechanism. *IEEE Communications Magazine*, pages 519 – 528, March 2002.

[31] K. Kar, M. Kodialam, and T.V. Lakshman. Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications. *IEEE Journal on Selected Areas in Communications*, 18(12):2566 – 2579, December 2000.

[32] K. Kar, M. Kodialam, and T.V. Lakshman. MPLS Traffic Engineering Using Enhanced Minimum Interference Routing: An Approach Based on Lexicographic Max-Flow. In *Proceedings of the Eighth International Workshop on Quality of Service (IWQoS)*, pages 2566 – 2579, Pittsburgh, USA, June 2000.

[33] D. Katz, D. Yeung, and K. Kompella. Traffic Engineering Extensions to OSPF Version 2. Internet draft, work in progress draft-katz-yeung-ospf-traffic-09.txt, Internet Engineering Taskforce, October 2002.

[34] M. Kodialam and T.V. Lakshman. Dynamic Routing of Bandwidth Guaranteed Tunnels with Restoration. In *Proceedings of the IEEE International Conference on Communication*, pages 902 – 911, Tel-Aviv, Isreal, March 2000.

[35] M. Kodialam and T.V. Lakshman. Minimum Interference Routing with Applications to MPLS Traffic Engineering. In *Proceedings of the IEEE International Conference on Communication*, pages 884 – 893, Tel-Aviv, Isreal, March 2000.

[36] A.E. Krzesinski and K.E. Müller. Traffic Protection in MPLS Networks using an Off-line Flow Optimization Model. In *Proceedings of SPIE, ITCom-2002*, pages 244 – 255, Boston, USA, March 2000.

[37] J. Moy. The OSPF Specification. RFC 1131, Internet Engineering Taskforce, September 1989.

[38] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031, Internet Engineering Taskforce, January 2001.

[39] H. Saito and M. Yoshida. An Optimal Recovery LSP Assignment Scheme for MPLS Fast Reroute. In *Proceedings: Networks 2002, 10th International Telecommunication Network Strategy and Planning Symposium*, Munich, Germany, June 2002.

[40] H. Smit and T. Li. IS-IS Extensions for Traffic Engineering. Internet draft, work in progress, Internet Engineering Taskforce, 1999.

[41] S. Suri, M. Waldvogel, and P.R. Warkhede. Profile-Based Routing: A New Framework for MPLS Traffic Engineering. In *Proceedings of the 2nd International Workshop on Quality of future Internet Services (QofIS'01)*, pages 138 – 157, Coimbra, Portugal, September 2001.

[42] B. Thomas and E. Gray. LDP Applicability. RFC 3037, Internet Engineering Taskforce, January 2001.

[43] B. Wang, X. Su, and C.L. Philip Chen. A New Bandwidth Guaranteed Algorithm for MPLS Traffic Engineering. In *In Proceedings of ICC*, New York, USA, 2002.

[44] Z. Wang and J. Crowcroft. Analysis of Shortest Path Routing Algorithms in a Dynamic Network Environment. Research notes rn/90/31, March 1990.

[45] Z. Wang and J. Crowcroft. QoS Routing for Supporting Multimedia Applications. *IEEE Journal on Selected Areas in Communications*, 14(7):1228 – 1234, September 1996.

[46] I. Widjaja, I. Saniee, A. Elwalid, and D. Mitra. Online Traffic Engineering with Design-Based Routing. In *15th ITC Specialist Seminar on Internet Traffic Engineering and Traffic Management*, Wurzburg, Germany, July 2002.

[47] S. Yilmaz and I. Matta. On the Scalability-Performance Tradeoffs in MPLS and IP Routing. In *Proceedings of SPIE ITCOM'2002: Scalability and Traffic Control in IP Networks*, Boston, USA, August 2002.