REROUTE SEQUENCE PLANNING IN MULTIPROTOCOL LABEL SWITCHING NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

İZZET GÖKHAN ÖZBİLGİN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF

MASTER OF SCIENCE

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2004

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Tayfur Öztürk
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science

Prof. Dr. İsmet Erkmen
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science

Asst. Prof. Dr. Cüneyt F. Bazlamaçcı
Supervisor

Examining Committee Members

Prof. Dr. Hasan Güran	(METU, EE)	
Asst. Prof. Dr. Cüneyt F. Bazlamaçcı	(METU, EE)	
Prof. Dr. Semih Bilgen	(METU, EE)	
Dr. Ece Güran	(METU, EE)	
Dr. Altan Koçyiğit	(METU, II)	

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited referenced all material and results that are not original to this work.

Name, Last name: İzzet Gökhan ÖZBİLGİN

Signature :

ABSTRACT

REROUTE SEQUENCE PLANNING IN MULTIPROTOCOL LABEL SWITCHING NETWORKS

Özbilgin, İzzet Gökhan

M.S., Department of Electrical and Electronics Engineering

Supervisor: Asst. Prof. Dr. Cüneyt F. Bazlamaçcı

December 2004, 57 pages

The growth of the Internet has caused the development of new protocols that enable IP networks to be engineered efficiently. One such protocol, Multiprotocol Label Switching (MPLS) enables IP datagrams in backbone networks to be forwarded based on the label switching forwarding paradigm. In MPLS networks, rerouting of Label Switching Paths (LSPs) can be needed in order to attain a better resource utilization in the network. In this case, a sequence of LSPs has to be found for their one by one reconfiguration without service interruption, involving the constraint that the link capacities should not be violated at any time during the rerouting process. This reroute sequence planning problem for LSPs is NP-complete. In previous works, the conditions of existence of any feasible reroute sequence are examined and algorithms are described for solving the problem, but it was shown that the problem is computationally hard in real-world situations because of the large amount of routers and LSPs in the network. In this work, we deal with the problem of reroute sequence planning problem of LSPs and present alternative solutions for the case when there is no feasible solution. We introduce a tool for the post-processing phase when a capacity violation is occurred during the sequence planning. We present an algorithm trying to reconfigure LSPs while allowing some interruption or degradation of traffic during the rerouting process.

Keywords: Traffic Engineering, Rerouting, Sequence Planning, MPLS, Label Switched Paths

ÖZ

ÇOKLU PROTOKOL ETİKET ANAHTARLAMA AĞLARINDA YENİDEN YÖNLENDİRME SIRASI PLANLAMA PROBLEMİ

Özbilgin, İzzet Gökhan

Yüksek Lisans Tezi, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Yrd. Doç. Dr. Cüneyt F. Bazlamaçcı

Aralık 2004, 57 sayfa

İnternet'in büyümesi IP ağlarının daha verimli düzenlenmesine olanak sağlayan yeni protokollerin geliştirilmesine neden olmuştur. Bunlardan biri olan Çoklu Protokol Etiket Anahtarlama (ÇPES), omurga ağlarındaki IP verilerin etiket anahtarlama üzerinden yön bulma yöntemine göre yönlendirilmesine olanak sağlar. ÇPES ağlarında Etiket Anahtarlanmış Yol (EAY)'ların yeniden yönlendirilmesi, ağda daha etkin bir kaynak kullanımı sağlamak için gerekli olabilecektir. Bu durumda, EAY'lerin trafik kesintisi oluşturmadan birer birer yapılandırılmasını sağlayacak bir sıranın bulunması söz konusu olacaktır. EAY'ler için bahsi geçen bu yeniden yönlendirme sırası planlama problemi NP-zorluktadır. Önceki çalışmalarda, uygun yeniden yönlendirme sıralarının bulunması şartları incelenmiştir ve problemin çözümü için algoritmalar tanımlanmış, fakat gerçek yaşamda ağ içerisindeki yönlendirici ve EAY'lerin çok fazla oluşundan dolayı problemin hesaplanabilirliğinin zorluğu gösterilmiştir. Bu çalışmada, EAY'lerin yeniden yönlendirme sırası planlama problemi ele alınmış ve uygun bir çözüm olmadığı durumlar için alternatif çözüm yolları sunulmuştur. Sıra planlaması sırasında kapasite aşımı olması durumunda ileri evrede kullanılacak bir araç tanıtılmıştır. Ayrıca yeniden yönlendirme işleminde, ağ trafiğinde kesilme veya azaltmaya izin verererek EAY'lerin yeniden yapılandırılmasını sağlayacak bir algoritma

sunulmuştur.

Anahtar Kelimeler : Trafik Mühendisliği, Yeniden Yönlendirme, Sıra Planlaması, MPLS, Etiket Anahtarlanmış Yol

To My Parents,

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my supervisor Asst. Prof. Dr. Cüneyt F. Bazlamaçcı for his guidance and constructive criticism throughout this study.

I wish also to give my deepest thanks to my parents for their love, support and assistance throughout my life.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ATM	: ASYNCHRONOUS TRANSFER NODE
BGP	: BORDER GATEWAY PROTOCOL
DIFFSERV	: DIFFERENTIATED SERVICE
DLCI	: DATA LINK CIRCUIT IDENTIFIER
FEC	: FORWARD EQUVALENCE CLASS
IETF	: INTERNET ENGINEERING TASK FORCE
INTSERV	: INTEGRATED SERVICE
IP	: INTERNET PROTOCOL
ISP	: İNTERNET SERVICE PROVIDER
LDP	: LABEL DISTRIBUTION PROTOCOL
LER	: LABEL EDGE ROUTER
LSP	: LABEL SWITCHING PATH
LSR	: LABEL SWITCHING ROUTER
MPLS	: MULTI PROTOCOL LABEL SWITCHING
NP	: NON-DETERMINISTIC POLYNOMIAL
OSPF	: OPEN SHORTEST PATH FIRST
QOS	: QUALITY OF SERVICE
RSP	: REROUTE SEQUENCE PLANNING
RSWP	: RESOURCE RESERVATION PROTOCOL
SWP	: SHORTEST WIDEST PATH
TE	: TRAFFIC ENGINEERING
TTL	: TIME-TO-LIVE
VCI	: VIRTUAL CIRCUIT IDENTIFIER
VPI	: VIRTUAL PATH IDENTIFIER
WSP	: WIDEST SHORTEST PATH

CHAPTER 1

INTRODUCTION

The growing number of new applications has driven the demand for increased and guaranteed bandwidth requirements on the Internet and intranets. New bandwidth hungry applications such as voice and multimedia services are also being developed. They bring many problems to present-day networks, like speed, scalability, quality-of-service (QoS) management, traffic engineering and bandwidth-management.

In order to meet the service requirements of these modern applications on the Internet, new techniques and protocols are being developed. The *Internet Engineering Task Force (IETF)* has proposed many service models and mechanisms to meet the demand for the time-sensitive and QoS guaranteed applications while keeping the network fair and efficient for all. Some examples are Integrated Services (IntServ) model [1, 2], Differentiated Services (DiffServ) model [3, 4] and Traffic Engineering (TE) [10, 11, 12].

Multi Protocol Label Switching (MPLS) [5, 6, 7, 8] is an other IETF proposal and an efficient technology for forwarding data packets through a communication network. MPLS, primarily evolved from *Cisco's Tag Switching* [9], uses a technique known as "label switching" to forward data through the network.

MPLS is an approach for achieving the simplified connection-oriented forwarding characteristics of layer 2 switching technologies while retaining the equally desirable flexibility and scalability of layer 3 routing, i.e., a hybrid solution combining the advantages of network layer routing and link layer switching.

Traffic flows in MPLS networks use Label Switched Paths (LSPs) that are previously established by their source and destination routers called Label Edge Routers (LERs). In other words, the ingress-egress points of an LSP are LERs, while the other MPLS capable routers – that can only be transit nodes along the LSPs – are the Label Switching Routers (LSRs).

MPLS is very useful for Internet Service Providers (ISPs). It supports various traffic engineering features to control the path of LSPs in their networks. With the help of these features, they enable an enhanced utilization of network resources, offering QoS guarantees, and increasing network reliability [10].

MPLS can be extended with a bandwidth reservation related information distribution component. Therefore; LERs can route Constraint-Based LSPs using Constrained Shortest Path First (CSPF) routing algorithm for each LSP and one can find a feasible path -if exits- for an LSP to be routed even if the default path selected by the Interior Gateway Protocol contains any link whose reservable bandwidth is less than the given LSPs required bandwidth. In case of heavy loaded networks, after several successive on-demand LSP establishments and deallocations, it is likely that some LSPs do not use shortest possible paths and this will result in a poor resource utilization compared to the optimal state. [11]

A network operator can establish the LSPs explicitly. By using this feature of MPLS, one can achieve to optimize the LSP placement globally with either a traffic engineering tool located in a central place of the network or an algorithm which balances the load in the network [12, 13]. The calculated paths should be routed strictly and explicitly so that the whole path of each LSP is completely determined.

To reroute the LSPs without any interruption of traffic during the rerouting; initially the new path of the LSP ought to be established while the previous one is still carrying traffic so that the traffic is switched easily to the new path and finally the old path is torn down. There are difficulties in this operation. Namely, some of the LSPs may not be rerouteable along their new paths as there might not be enough bandwidth on some links of these paths. Consequently, the reroute sequence should be planned before the reroute action, such that the reroute of the LSPs in the calculated sequence would not exceed any reservable bandwidth threshold. In this work, we deal with the determination of the LSP reroute sequence.

In Chapter 2, an overview of MPLS networks and rerouting is reviewed. The reroute sequence planning (RSP) problem is explained and basic definitions for the problem are given. A previous work on RSP problem is also presented in this chapter.

In Chapter 3, the heuristic algorithms for the solution of reroute sequence planning problem are explained. Moreover, alternative solutions for the case when there is no feasible solution are presented.

In Chapter 4, the software “RSPSln” is explained. The “RSPSln” software is developed specifically for calculating a reroute sequence for the LSPs before the reroute action and testing the heuristic methods and the approaches used in infeasible cases. The features of the software, sub-components of the program and its implementation are also explained in this chapter.

Chapter 5 introduces a detailed comparison of the heuristic algorithms and approaches used in infeasible cases which are presented in Chapter 3. It explains which method is better in what conditions over which networks stating their advantages and disadvantages. Problem instance generation and the benchmark procedure and investigations of the applied benchmark test are also presented in this chapter.

Chapter 6 concludes the study and summarizes the work done stating possible directions for further studies.

CHAPTER 2

REROUTE SEQUENCE PLANNING PROBLEM

This section will briefly introduce the MPLS networks, their background, components and rerouting scheme. It also defines the reroute sequence planning problem in MPLS networks.

2.1 Overview of MPLS Networks and Rerouting

MPLS architecture combines the scalability and flexibility of Layer 3 routing with the performance, QoS, and traffic management of Layer 2 switching. By combining the best network layer routing and link layer switching, it provides traffic engineering in IP networks and speeds up the packet forwarding. The primary aspects of MPLS that enable these are separation of control and forwarding components, the label stack and the LSPs.

Rerouting of LSPs in the MPLS layer can happen due to a number of reasons. When a new LSP is admitted into the network, if the available bandwidth on the route of this LSP is insufficient, or when a link or a node (or some other component) fails, then one or more LSPs may be rerouted over new paths. Also networks need to perform load balancing or re-optimization of the LSPs from time to time such that the network resources are optimally allocated.

2.1.1. MPLS Background and Operation

Multi-Protocol Label Switching Networks (MPLS), emerged from the evolution of routing and forwarding protocols, is a highly efficient scheme for forwarding data packets through a communication network [5]. It uses a technique known as label switching to forward data through the network. A small fixed format label is encapsulated within each data packet on its entry into the MPLS network.

In the MPLS core, LSRs read only the label, not the network layer packet header. Labels have only local significance between two devices that are involved in communication. At each hop across the network, the routing of the data packet is based on the value of the incoming label and eventually issued to an outwards interface with a new label value. The path that data traverses through a network is defined by the transition in label values, as the label is swapped at each LSR. Since the mapping between labels is constant at each LSR, the path is determined by the initial label value. Such a path is called a Label Switched Path (LSP). At the ingress to an MPLS network, each packet is examined to determine which LSP it should use and hence what label to assign to it. Here, the IP packets are classified based on the information carried in the IP header of the packets and the local routing information maintained by the LSR and a label is assigned to them. The labels are then distributed to the neighboring LSRs, and further associated and distributed till the egress LER is reached.

Each LSR uses the label to forward the packet. At each LSR the outgoing label replaces the incoming label and the data packet is switched to the next LSR. The process of switching the label is known as Label Swapping. The set of all packets that are forwarded in the same way is known as a Forwarding Equivalence Class (FEC). One or more FECs may be mapped to a single LSP. Classification and filtering of the information packet happen only once, at the ingress edge. At the egress edge, labels are stripped and packets are forwarded to their final destination [5, 7, 8, 14].

The reader should note that the given successive details in separate paragraphs below are numbered consecutively within Figure 2-1. Within this context, Figure 2.1 shows the MPLS architecture and in general, the operations within an MPLS network can be described as follows. Primarily, once a packet enters the ingress LER, based on routing and policy requirements, the LER selects and applies a label to the packet header and forwards the packet. Thus, the LER determines the FEC for each packet, deduces the LSP to use and adds a label to the packet. The LER then forwards the packet on the appropriate interface for the LSP.

Secondly, packet is then be forwarded through the MPLS network based on its associated FEC. At each hop in the network, a router examines a label to figure out the next forwarding hop for the packet. This eliminates resource intensive address lookups that reduce overall packet throughput and limit scalability. Packets are forwarded along a LSP where each LSR makes forwarding decisions based solely on the contents of the label. At each hop, the LSR strips off the existing label and apply a new label that tells the next hop LSR how to forward the packet. The labels are distributed between LERs and LSRs using Label Distribution Protocol (LDP) or Resource Reservation Protocol (RSVP) or they can also be chosen based on routing protocols like Open Shortest Path First (OSPF).

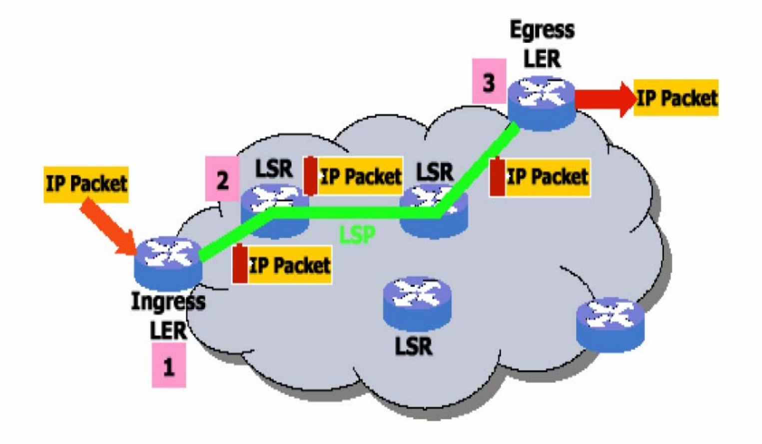

Figure 2-1 MPLS Architecture

Lastly, when the packet reaches the Egress LER, the Egress LER will pop the labels (i.e., remove MPLS header) and copy Time-to-Live (TTL) information from the label into IP header. Then it will recompute the IP checksum and continue to forward packet according to the IP header information [5, 6, 8, 15].

2.1.2. MPLS Components

MPLS has many new features to support the forwarding of IP packets. These new features enable it to function at a high degree of performance and intelligence than the previous technologies like ATM.

MPLS Header

The label is a condensed view of the header of an IP packet containing information needed to forward the packet from source to destination. Unlike IP header, it does not contain an IP address, but a numerical value agreed upon by two MPLS nodes to signify a connection along an LSP. The label is short, fixed length, physically contiguous identifier, which is used to identify the path a packet should traverse (FEC). It is encapsulated in a Layer-2 header along with the packet. The receiver router examines the packet for its label content to determine the next hop. Once a packet has been labeled, the rest of the journey of the packet through the backbone is based on label switching. The label values are of local significance only, meaning that they pertain only to hops between LSRs.

Once a packet has been classified as a new or existing FEC, a label is assigned to the packet. The label values are derived from the underlying data link layer. For data link layers (such as frame relay or ATM), Layer-2 identifiers such as data link connection identifiers (DLCIs) in the case of frame-relay networks, or virtual path identifiers (VPIs)/virtual channel identifiers (VCIs) in the case of ATM networks, can be used directly as labels. The packets are then forwarded based on their label value [15, 8].

The 32-bit MPLS header contains the label field (20-bits) carrying the actual value of the MPLS label. The experimental (Exp) field (3-bits) is reserved for experimental use. A single bit stack (S) field supports a hierarchical label stack. It is set to one for the last entry in the label stack (i.e. for the bottom of the stack) and cleared to zero for all other label stack entries. In addition, an eight bit time-to-live (TTL) field provides conventional IP TTL functionality (Figure 2-2) [5, 16].

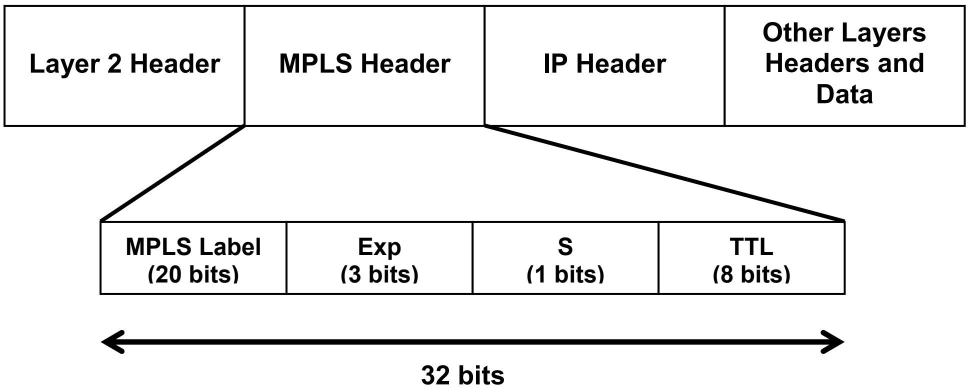

Figure 2-2 MPLS Header Format

Forward Equivalence Class (FEC)

The forward equivalence class (FEC) has a very special meaning in MPLS. It represents a group of packets that have the same requirements for their transport. All packets in such a group are forwarded in the same manner to the destination. Each LSR will build a table to specify how a packet should be forwarded. This table is called a label information base (LIB), which is comprised of all kinds of label bindings that the LSR receives. In MPLS, the packet is assigned to a particular FEC as it enters the network. FECs are usually based on service requirements for the packets.

Table 2-1 shows a simple example of the LIB tables. Consider one flow as a stream of regular data exchange between servers (e.g., file transfer protocol [FTP]) and the other flow as a packet stream of an intensive video stream, which requires the traffic engineering parameters of QoS (e.g., videoconferencing). These packet streams are classified into 2 separate FECs at the ingress LSR. The label mappings associated with the streams are 3 and 9 and the input ports at the LSR are 1 and 2, respectively. The corresponding output interfaces are 3 and 1, respectively. Lastly, label swapping must also be done, and the previous labels must be exchanged for 6 and 7, respectively [8].

Table 2-1 Label Information Base

INPUT PORT	INCOMING PORT LABEL	OUTGOING PORT	OUTGOING PORT LABEL
1	3	3	6
2	9	1	7

LSRs and LERs

The routers in the MPLS can be classified into two types: label edge routers (LERs) and label switching routers (LSRs).

An LSR is a high-speed router in the center of an MPLS network. LSR also participates in the establishment of LSPs by appropriate label signaling protocols. Based on the established paths, data traffic can be switched with very high speed.

An LER is router that operates between the access network and MPLS network. After LSPs are established, LERs forward the traffic on to the MPLS network, using the label signaling protocol at the ingress, while at the egress they distribute the traffic back to the access Networks. The LER plays a very important role in the assignment and removal of labels, when traffic enters or exits an MPLS network.

Label-Switched Paths (LSPs)

It's the path for a labeled packet to go through a network according to a FEC. MPLS provides two ways to set up an LSP, namely hop-by-hop routing and explicit routing. In hop-by-hop routing, every LSR independently selects the next hop for a given FEC, based on any available routing protocols, such as OSPF and so on. In explicit routing, the ingress LSR specifies the list of nodes through which the packet should go. It is worth to be mentioned here that the setup for an LSP is unidirectional, which means the return traffic must take another LSP.

Label Stack

One of the characteristics that distinguishes MPLS from earlier label swapping technologies (such as frame relay and ATM) is the concept of label stacking, which is an ordered set of labels affixed to a packet. Label stacking allows multiple labels to be assigned to the same packet at one or more nodes in the network, in a hierarchical arrangement [18].

As the packet traverse the network, only the topmost label is swapped. The labels are organized in a last-in first-out (LIFO) manner, which means the topmost label signifies the highest LSP, and each successive label signifies the next lowest LSP. Label stack supports a capability of traffic classification and helps to reduce both the size of the forwarding tables on the LSR and the complexity of managing the data traversing across the backbone. It also improves traffic engineering in the network.

Label Distribution Path (LDP)

In order for an LSR to swap the label on an incoming packet and forward it to its downstream peers, it must have a method of learning what label value its downstream peers are expecting. Currently, several protocols can be used for the distribution of labels between LSR peers, such as LDP, constraint based routed label distribution protocol (CR-LDP), RSVP or BGP [8, 15].

2.1.3. Rerouting

Traffic Engineering (TE) has become an essential requirement for Internet Service Providers (ISPs) to optimize the utilization of existing network resources and to maintain a desired overall QoS with fewer network resources [19].

One of its key objectives is to balance the loads across a network. This optimization of network utilization is not easily achieved, since the routing algorithms are mainly based on shortest path computations using simple additive link metrics. These two objectives are usually incompatible, i.e. a path with the least number of hops does not necessarily have to be the path with the best resource consumption. This results in subsets of network resources becoming congested, while other resources along alternate paths remaining underutilized.

This type of poor utilization can also be seen during the rerouting of LSPs in the MPLS. In an operational MPLS network, a bandwidth CSPF computation algorithm for serving the LSP demands offers a good practical solution. However, the edge routers have only local information about the network resources, thus these local routing decisions made in LERs may result in a degraded global network performance [20, 24].

One solution for this problem is to combine the normal use of CSPF with global re-optimization of LSPs. The calculated new paths should be strictly explicit routes, i.e. the whole path of each LSP is completely determined. To avoid the interruption of traffic through the LSPs during the rerouting from the old to the new paths, the reroute sequence should be planned before the rerouting action such that the rerouting of the LSPs in the calculated sequence would be feasible, i.e. rerouting action should not exceed any reservable bandwidth threshold. This sequence planning procedure, defined and investigated in this work, can be realized as a new means of traffic engineering that helps global path optimization.

2.2. Problem Description

In the Reroute Sequence Planning (RSP) problem, the MPLS network is given the reservable link bandwidth values and with LSPs routed on their original (old) paths. In this network, there is no capacity violation or a similar problem. Every packet coming to the edge router knows the right path to go to the destination and therefore the number of LSPs and their paths are certain. Moreover the optimized (new) paths of the LSPs for the same network which can be calculated by any routing algorithm such as Shortest Widest Path (SWP), Widest Shortest Path (WSP) or a global LSP optimizer defined in [17] are also given.

The goal in RSP problem is to find a sequence of the one-by-one reconfiguration of the LSPs from their original paths to the optimized paths. In other words, the target is to seek a feasible sequence for one-by-one rerouting of LSPs so that the maximal reservable bandwidth of the links should not be exceeded at any time during the rerouting process. Therefore, to avoid interruption of traffic through the LSPs during the rerouting, the new path of the LSP must be established first while the old one is still carrying the traffic. Then the traffic can be switched easily to the new path, finally tearing down the old path.

As it can be seen in Figure 2-3, the only way to achieve the required rerouting is to perform the task in the order of LSP 1, LSP 2 and LSP 3. There is no other feasible reroute sequence for these LSPs.

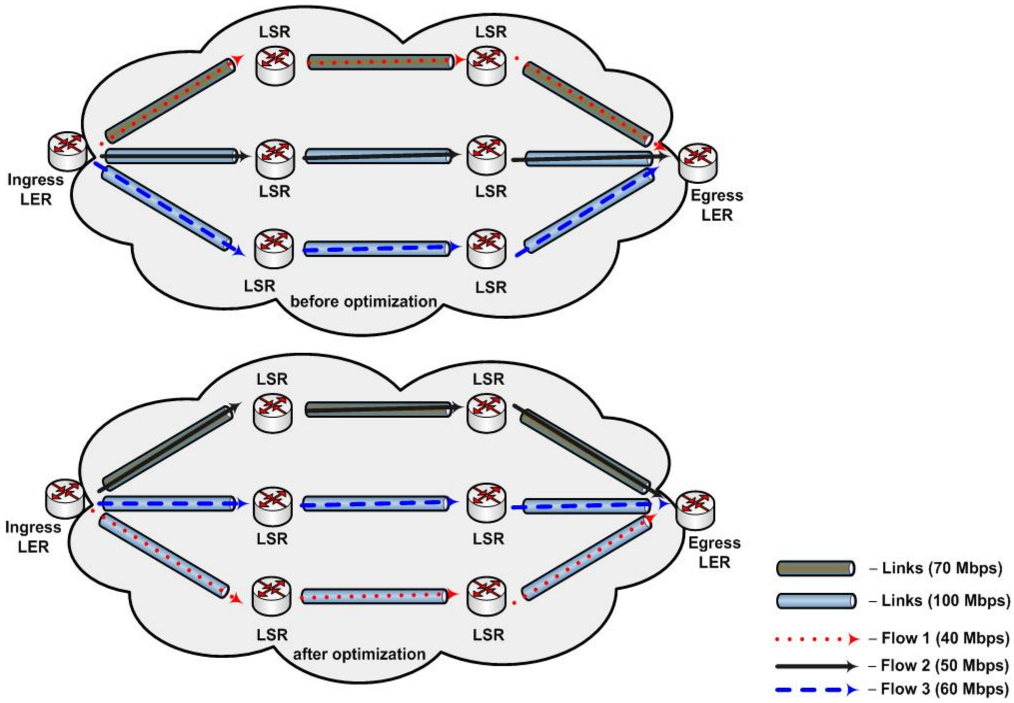

Figure 2-3 Paths that cannot be rerouted in optional sequence without capacity violation

In this work two restrictions are considered in order to make the reroute process practically relevant:

- LSPs can not be split into several paths, i.e., they must be rerouted entirely in a single step during the rerouting process, and

- LSPs must be rerouted directly to their new paths, i.e., no temporary paths can be used during the rerouting process.

For the graph based definition of the RSP problem, the same terminology is used in [17]. Let the directed graph *G= (V, E, C)* represent the physical topology of the MPLS network, where

V: the set of nodes

E: the set of directed edges, i.e. ,

$$E: \{e= (u, v): u ,v \in V \}$$

c: edge capacity function corresponding to the total reservable bandwidth of the links, i.e.,

$$c: E \rightarrow R^+$$

Moreover, suppose a given set of LSPs are denoted as;

$$L =\{ l = (s, t, b, P, Q) \mid s, t \in V, b \in R^+ , P = \{ e_1 = (s, u_1), e_2 = (u_1, u_2), e_3 = (u_2, u_3), \ldots, e_k = (u_{k-1} , u_k) \} , Q = \{ f_1 = (s, v_1), f_2 = (v_1, v_2) , f_3 = (v_2,v_3), \ldots, f_r = (v_{r-1}, v_r) \} \},$$

where for each l_i = (s_i, t_i, b_i, P_i, Q_i) , s_i and d_i denote the source and the destination node respectively, b_i denotes the transmission capacity (flow value) of the LSP, P_i and Q_i denote the old and new paths (ordered set of edges) of LSP l_i, respectively. Let *n* be the number of LSPs in the LSP set *L*.

It is assumed that the system of old paths with the corresponding capacities is feasible as well as the system of new paths. That is, for each edge the given edge capacity *c(e)* is not violated by the paths using that edge. Thus,

$$\sum_{i\,:\,e \in P_i} b_i \le c(e) , \forall e \in E \qquad \textbf{(1)}$$

$$\sum_{i\,:\,e\,\in\,Q_i} b_i \;\le\; c(e)\,,\ \forall e \in E \qquad \textbf{(2)}$$

We will also introduce a Boolean variable $Z_{e,i}$, for every edge and LSPs to display if there is a common edge between the old and the new paths of an LSP. Let

$$Z_{e,i} = \begin{Bmatrix} 1, \ \textbf{iff} \ e \in P_i \cap Q_i\,, \ \forall e \in E, \ i = 1,2,......n \\ 0, \ \textbf{otherwise} \end{Bmatrix} \qquad \textbf{(3)}$$

The goal in the RSP problem is to determine a reroute sequence

$$S=\{s_1,s_2,s_3.....s_n\}$$

of LSPs that enables the LSP rerouting process which means that the first LSP to be rerouted is l_{s_1} and the last one to be rerouted is l_{s_n} , such that the following constraints are also satisfied:

$$\forall s_i \in [1,n]\,,\ i \ne j \ \Rightarrow\ s_i \ \ne\ s_j \qquad \textbf{(4)}$$

$$\sum_{i\,:\,e\in P_{s_i},\,i\ge j} b_{s_i} \;+\; \sum_{i\,:\,e\in Q_{s_i},\,i\le j} b_{s_i} \;-b_{s_j} * Z_{e,s_j} \;\le c(e)\,,\forall j \in [1,n]\,,\ \forall e \in E \qquad \textbf{(5)}$$

The above constraints guarantee to reroute without exceeding the capacity on the edges. It can be seen that the capacity constraint is liable to Boolean variable Z_{e,s_j} since the common edges of the old and new paths of an LSP should not be reserved twice during the rerouting process. The first summation is the bandwidth reserved by LSPs that are not rerouted away from this edge yet. The second summation however is the bandwidth reserved by LSPs that are already rerouted to this edge. The third term subtracts the bandwidth of the LSP being rerouted at each step *(j)*, if the edge exists in both the old and the new one, so that the bandwidth of this LSP will not be added twice.

The RSP problem described above is a problem that has arised recently, so there is not much work carried in connection with it yet. Nevertheless, every other day, the increase in QoS, load balancing and reliable network intended Traffic Engineering applications will increase the research effort about this matter.

The RSP problem is proved to be NP-hard, as it can be traced back to the well-known partitioning problem [17, 28], indicating that exact solutions are probably hard to find in reasonable time by some exact search procedure. For this reason, only heuristic solutions can be found to the problem of finding a suitable permutation of LSPs for the reroute process in the literature.

The RSP problem was firstly introduced by Jozsa and Magyar [17], in which four heuristic algorithms were presented and investigated on real-world backbone networks taken from [30]. Then the problem was extended to protected traffic flows (i.e. traffic flows with active and backup paths) and the same heuristics were applied in [28]. Later, El-Hawary *et al.* [21] introduced some refinements into the existing heuristic approaches, which are evaluated by simulation using data again from [30].

It turned out in these works that some RSP problem instances can not be solved by these simple algorithms already at moderate network load. These results serve as a motivation for further investigations from another point of view, namely, the generation of artificial examples in the test scenarios, which provably have feasible solutions. Alternative solutions are also proposed by Jozsa and Makai [27] for the cases when feasible solutions do not exist.

In the present study all the above algorithms are evaluated through empirical comparisons on randomly generated networks different than the ones in [30] and post-processing approaches for infeasible cases proposed in [27] are applied.

Furthermore, new post-processing algorithms are presented and applied within the scope of this thesis.

CHAPTER 3

SOLUTION APPROACHES

This chapter covers the simple heuristic approaches that were introduced in [17] for RSP problem while revealing probable solution approaches for infeasible cases.

3.1. Simple Heuristics

Herewith, four heuristic approaches for evaluation of sequential rerouting are presented. They all commonly constitute iterative techniques in which a single LSP is chosen at each step. The iteration scheme is then repeated until all LSPs sequential rerouting is generated one after another greedily. There actually lie four methods for which an LSP is selected at each step.

It is a common feature for all four greedy selection methods that the selection is based on rerouteable LSPs if there are any, otherwise such an LSP is selected whose rerouting violates some of the edge capacities. The LSP selection is done again by the corresponding greedy selection rule.

Given below, in Figure 3-1 is the basic reroute sequence evaluation algorithm flow chart.

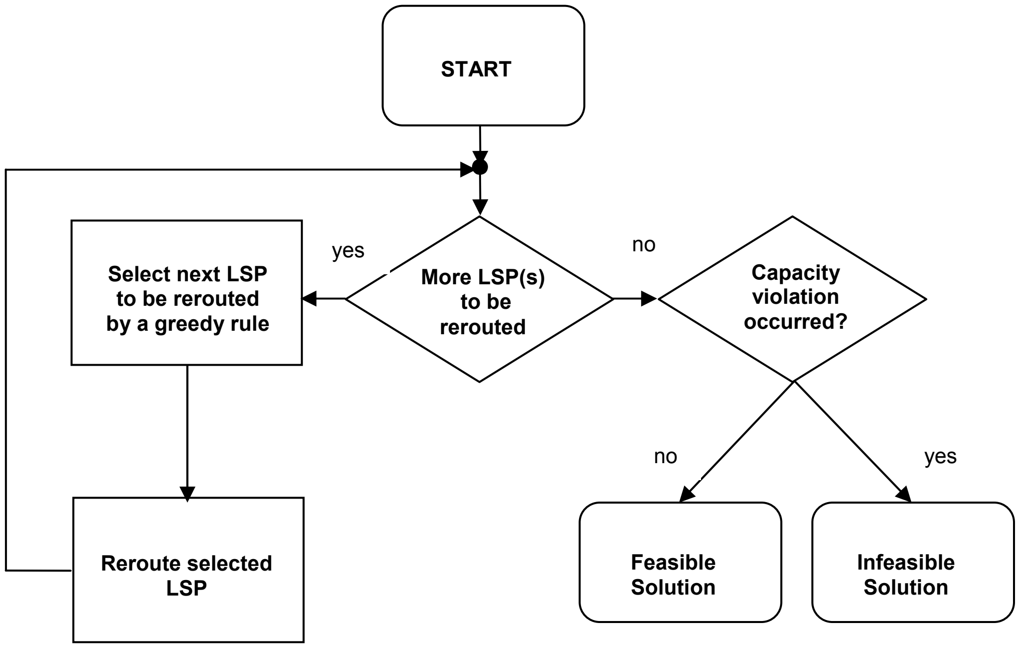

Figure 3-1 The Flowchart of the Basic Reroute Sequence Calculation Algorithm

The key step of the above algorithm is the selection of the LSP to be actually rerouted. Evidently not all n! permutations of the LSP order can be checked in reasonable time. This constitutes the major reason for advising such a greedy solution. Consequently, four methods for the LSP selection are introduced. All these methods assign a greedy utility value o_i to candidate LSP l_i in each iteration, and then they choose the candidate LSP that has the actually "greatest" utility value assigned. Note that if l_i is already rerouted, then the corresponding o_i value of that l_i is set to $o_i = -\infty$. The different approaches for evaluating the greedy utility value o_i for the candidate LSPs are categorized in four groups and all of them use the following terminology:

c (e) is the maximal capacity of edge *e*

r (e) is the actual reserved capacity of edge *e*

R (e) is the maximal capacity reservation on edge *e* during the whole reroute process, that is:

$$R(e) = \max r(e) \qquad \textbf{(6)}$$

$f(e)$ is the actual free capacity of edge e, that is:

$$f(e) = c(e) - r(e) \qquad \textbf{(7)}$$

$a(e)$ is the summed bandwidth of such LSPs that has to be allocated to the given edge in the subsequent rerouting steps, that is:

$$a(e) = \sum_{j:e \in Q_j \setminus P_j} b_j \quad \text{for all } j \text{ where } l_j \text{ is not rerouted yet.} \qquad \textbf{(8)}$$

With this terminology, the principle of selecting an LSP in the different approaches might be summarized as follows:

Random Selection (RS)

The first method selects the LSPs randomly: It builds up the reroute sequence by choosing the LSPs one by one randomly. This is a very simple method, but it provides a good benchmark for comparisons. The assignment of o_i can be done in constant time.

Minimal Violation (MV)

This method MV calculates for each non-rerouted LSP l_i the greatest capacity violation on its new edges (that are distinct from every old edge) if it is rerouted:

$$o_i = -\max_{e \in Q_i \setminus P_i} \{b_i - f(e)\} \qquad \textbf{(9)}$$

If some edges would be violated in case of rerouting LSP l_i, then $o_i < 0$, otherwise $o_i \geq 0$. The idea of this ranking is that the LSP to be rerouted next should violate the edge capacities in the slightest possible degree. If the selected LSP does not violate any capacity, the above rule results in such a way that the minimal free capacity of the edges decreases in the slightest possible degree. The complexity of computing o_i is proportional to the maximal number of edges in the new path of an LSP.

Maximal Freeing (MF)

The maximal freeing approach uses the capacity value *a(e)* to be routed for each edge *e*. It calculates the o_i values representing the total amount of capacity that is to be freed on the old edges of l_i for the subsequent LSP rerouting:

$$o_i = \sum_{e \in P_i \setminus Q_i} \{a(e) - f(e)\} \qquad \textbf{(10)}$$

The idea behind this rule is to favor those LSPs at the selection, which contain edges (in their old paths) that are in many new paths of LSPs (that are not rerouted yet) and has relatively few actual free capacity. The complexity of this method is equivalent to the one of the Minimal Violation.

The Most Rerouteable (MR)

In this method o_i is calculated to represent the number of LSPs that can be rerouted without capacity violation after the successful reroute of LSP l_i. By this approach we select a rerouteable LSP and in the next step we can select one from the maximal number of rerouteable LSPs. In this case the greatest o_i may be the same for more than one LSP, which is why we modified the utility value in the following way: we decrease the value of o_i with the summed ratio of rerouteable edges and total number edges to be rerouted for all LSPs if l_i is rerouted. The benefit of this approach is that it looks forward one reroute step. On the other hand, it has relatively larger computational complexity compared to the other methods, because the number of elementary steps needed for calculating each o_i is proportional to the maximal number of edges in the new path of an LSP multiplied with the number of LSPs to be rerouted.

A Refinement

Above described greedy methods are refined in the following manner. As mentioned before, the methods consist of a simple iteration, where in each step one LSP that will be rerouted in that step is selected. However, if there is currently no LSP that can be rerouted without exceeding the capacity constrains, unfortunately an LSP that violates some of the edge capacities have to be selected. Again the LSP selection is done by the corresponding greedy selection rule and the rule does not take into account how much violation is occurred.

Now, if there are not any rerouteable LSPs, then such an LSP is selected whose rerouting violates some of the edge capacities but the violation is aimed to be kept at minimum. For this reason, the greedy approach minimal violation (MV) is used during the reroute sequence calculation as shown in Figure 3-2.

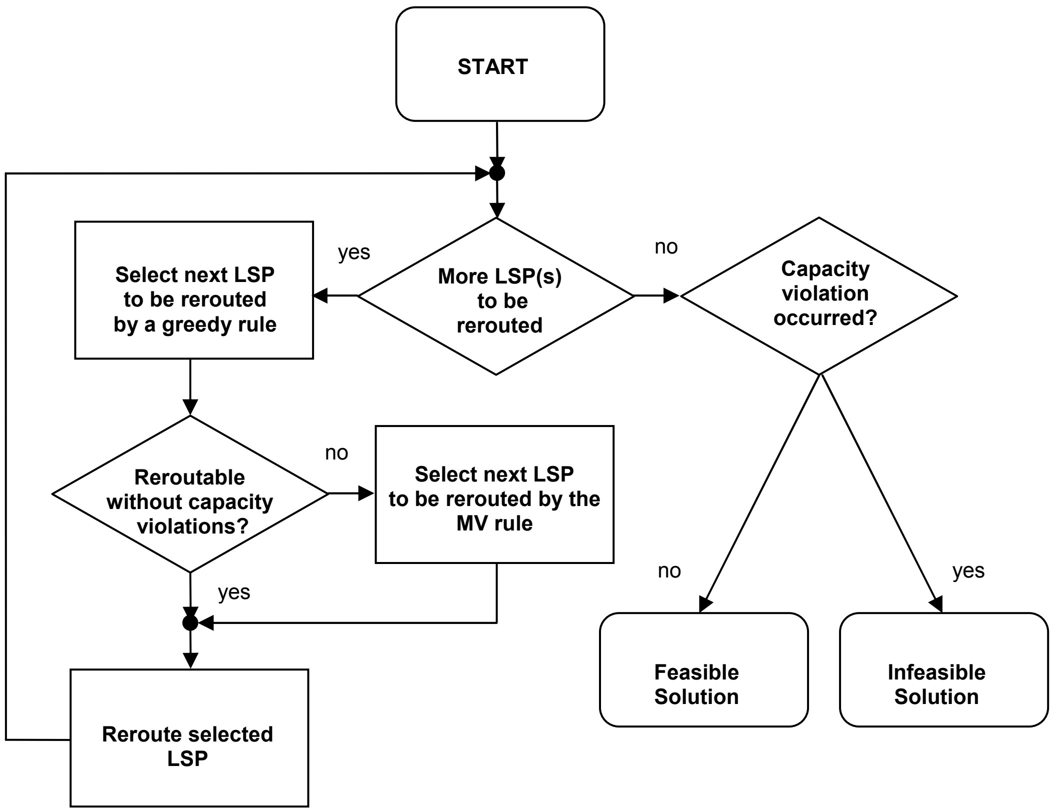

Figure 3-2 The Flowchart of the Basic Reroute Sequence Calculation Algorithm with Refinement

3.2. Sequence Planning in Infeasible Cases

On real network situations, usually it is not possible to derive feasible solutions by the above-described algorithms. If that would be the case, there are two possible approaches: One can

a) either say that there exists no solution hence respective LSPs can not be reconfigured on their recently generated paths, or alternatively

b) continue the rerouting process while allowing some interruption or degradation of traffic for the purpose of reconfiguration of LSPs.

One should note that, here in the latter approach, some capacity violations are allowed in the reroute sequence calculation. Despite the existence of these violations, they appear only in the calculations, and they are eliminated prior to performing the rerouting action. Although several approaches are possible in infeasible cases, two of them are applied in this work. Moreover, one of these approaches is refined and with the aid of the results of these two methods, a combination of the two is also developed (Figure 3-3).

3.2.1. The Interrupting Approach

The interrupting approach, explained below, allows some LSPs to be interrupted before the rerouting process so that the remaining LSPs can be rerouted without edge capacity violation according to the determined schedule.

First, the reroute sequence is calculated while allowing some bandwidth thresholds to be exceeded. When this sequence calculation is finished, there will be a list of edges together with corresponding bandwidth values indicating the maximal violation of those edges in the reroute process, that is,

$$V(e) = R(e) - c(e) \quad \textbf{(11)}$$

where *R (e)* is defined in eqn. (6).

At this phase some LSPs have to be selected, such that if they are temporarily deallocated at the beginning of the reroute process, the remaining LSPs can be rerouted to their new paths in the calculated sequence without any edge capacity violation. Finally, the previously temporarily deallocated LSPs are established on their new paths within the desired order.

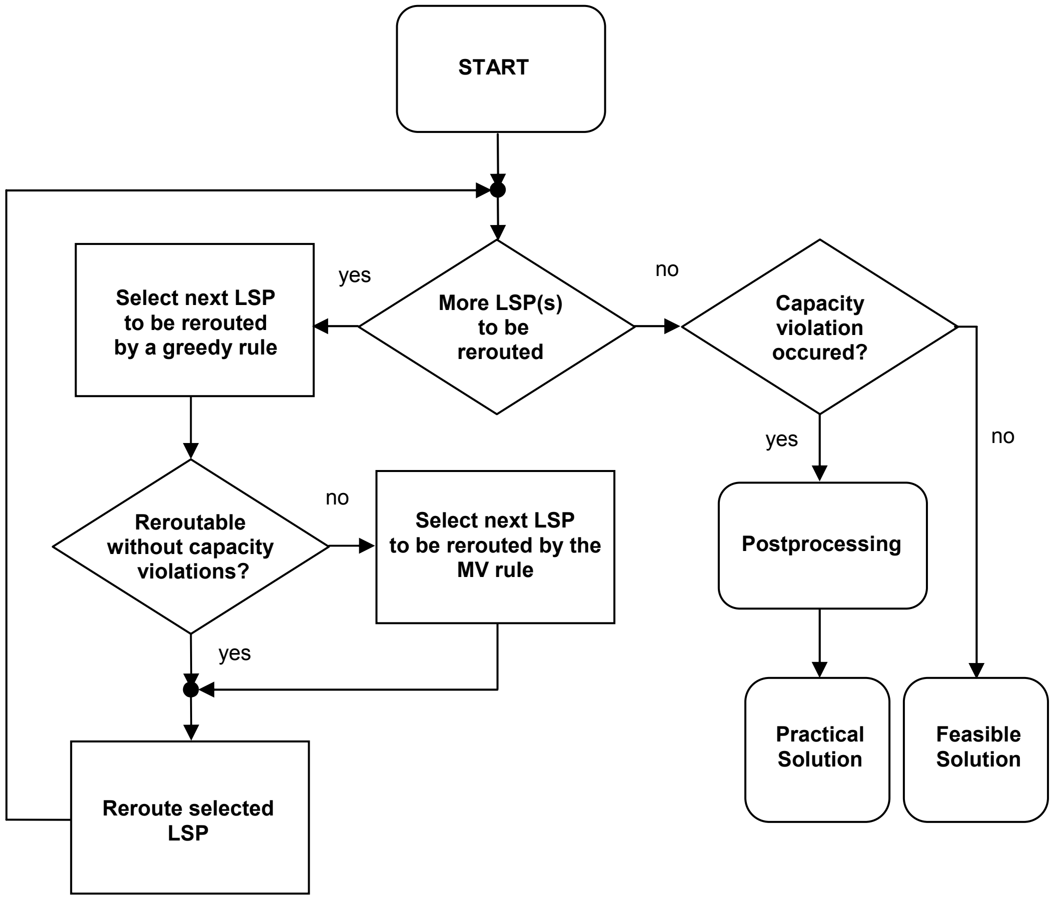

Figure 3-3 The Flowchart of the Basic Reroute Sequence Calculation Algorithm with Postprocessing

Once some LSPs are selected to be temporarily deallocated, note that the traffic through these LSPs are interrupted during whole rerouting process. Although this is the price of not violating the bandwidth threshold, the number of such LSPs should be kept at a minimum. Therefore; these LSPs are selected one-by-one, and the calculated sequence is controlled whether it works or not in each step (i.e. all other LSPs except the deallocated one can be rerouted to their new paths in the calculated sequence without and edge capacity violation). This iterative LSP selection process for deallocation runs until all the violations are eliminated during the rerouting of all remained LSPs.

The selection of LSPs for deallocation is achieved in each step by considering the total decrease in capacity violation and the capacities of the edges which are not violated with the deallocation of l_i. For this, two values are calculated in the following way for each candidate LSP l_i that has not been selected for deallocation before:

d_i is the total decrease in capacity violation with the deallocation of l_i, that is,

$$d_i = \sum_{e \in P_i} min \{b_i, max\ [0,V(e)]\} \quad \textbf{(12)}$$

u_i is the amount of capacities that do not decrease any violation with the deallocation of l_i, that is,

$$u_i = \sum_{e \in P_i} max \{0, b_i - max\ [0,V(e)]\} \quad \textbf{(13)}$$

$$= |\ P_i\ | * b_i - d_i$$

where $|\ P_i\ |$ denotes the path length of LSP l_i (i.e. how many number of edges LSP passes through)

After these calculations for each candidate LSP l_i that has not been selected for deallocation before, the LSP with the geatest $d_i - u_i/w$ is selected, where control parameter w is set to be equal to the average link capacity in the network in our investigations. As w tends to infinity, the selected LSP will eliminate the most amount of violation, but it may have much redundant capacity deallocation. With lower w values the number of deallocated LSPs can be enormously high, although there will not be that much of unnecessary deallocated capacity. The iterative LSP selection process for deallocation runs until all the violations are eliminated.

3.2.2. The Shrinking Approaches

The shrinking approach is another postprocessing method which can be used to eliminate the violations appeared during the reroute sequence calculation. It decreases the reserved bandwidth of all LSPs before the rerouting process, which results in temporary service degradation but in fortunate cases this amount of degradation is so small that the network users do not perceive it. The first step is common with the interrupting approach: the reroute sequence is calculated while allowing capacity violation. Then the bandwidth reservations of LSPs are decreased so that all violations would be eliminated. In the simplest case the bandwidth values are decreased uniformly: the original values are multiplied by $1/(1+X)$ where X is the maximal violation (defined and investigated in section 5.1). At the final step, after the rerouting process the bandwidth values are restored.

By using this approach one can find the total decrease in the bandwidth of LSPs (i.e. the total bandwidth decrease in the network) so that there will be no more violation during the rerouting process in the calculated sequence. Here, the value X, utilized for the replica of capacity violation, is a critical value since assigning new bandwidths by using larger values than X will, though there does not occur any violation, bear an unnecessary shrinkage of bandwidth in the network. On the other hand, if a value smaller than X is used during the shrinking of bandwidths of LSPs, this will cause violation during the rerouting process in the calculated sequence.

Furthermore, the shrinking approach can be applied to the LSPs in a different manner. In the shrinking approach briefly stated above, the successive bandwidth values of all LSPs were decreased uniformly with respect to the given ratio, i.e. *1 / (1+X)*. Nevertheless in certain cases, instead of all LSPs, decreasing the bandwidth values of only certain LSPs, will suffice a correct and desired working of the rerouting sequence. An approach which can be accepted as a refinement for the shrinking approach can be summarized as follows:

As in the above approaches, the method comprises the technique of calculating reroute sequence while allowing some bandwidth thresholds to be exceeded. When the sequence calculation is completed, there will be a list of edges which are violated during the rerouting process. At this stage, LSPs which involve one of these edges occurring on either the old or optimized paths are selected and respective bandwidth values of only these selected LSPs are degraded with respect to the above denoted ratio. Consequently, this will eliminate all the violations during the rerouting process in the calculated sequence and enable the degradation of bandwidths on certain number of LSPs instead of all, allowing rest to have their original capacities.

3.2.3. The Combined Approach

The combined approach might not be considered as a new post-processing method. Nevertheless within the context of Traffic Engineering it is regarded to be an empirical solution for rerouting successive LSPs. This approach can well be briefly defined as the combination of shrinking and interrupting approaches. In such a combined approach, deallocation as it is experienced in interrupting approach or shrinking bandwidths of certain LSPs are accomplished for the purpose of discarding likely occurring violations during the routing process; therefore a valid combination of these approaches might be preferred as an alternative. As a result, certain differentiation in between the evaluated values will be observed.

Assume a case in which 10% of LSPs are deallocated through the interrupting approach and 5% degradation in bandwidths of LSPs is accomplished following the shrinking approach for elimination of violation of edges that has taken place. In that case a degradation over 5% on bandwidths will be effective in deallocation of lesser LSPs. This means for example that by increasing the number of combinations, usable number of viable options can be increased for rerouting process for the network operator. Furthermore, for selecting the values in decreasing the bandwidths values, certain percentage of the critical value (the value defined in the shrinking approach) may well be taken into account. Consequently maximum and minimum cases are found for pre-evaluated LSPs.

It is worth noting that by feasible solution we mean a solution that does not have capacity violation. The rerouting in the calculated sequence can thus be performed without any service interruption or degradation. However, non-feasible solutions having some capacity violations can also be possible practical solutions, because in the above-described ways the rerouting can be performed in the calculated sequence without real capacity violations.

CHAPTER 4

AN INTEGRATED TOOL FOR THE REROUTE SEQUENCE PLANNING PROBLEM: RSPSln SOFTWARE

RSPSln software includes a label switched paths (LSPs) module with a graphical user interface (GUI) that generates a set of LSPs in the given fixed network and a test module for both reroute sequence planning approaches and post processing approaches for the infeasible cases.

This chapter describes the software by screenshots and gives information about the mechanisms running behind its interface.

4.1. Using RSPSln

The main GUI of RSPSln is shown in Figure 4-1. By using the buttons for the LSPs creation, one or more LSPs can be generated according to the network topology you select. In this work we used a universal network topology generation tool called BRITE, the Boston university Representative Internet Topology gEnerator, to generate our random network models [29]. The LSPs are created on these BRITE outputs according to the α and β parameters to be defined in section 5.1 later.

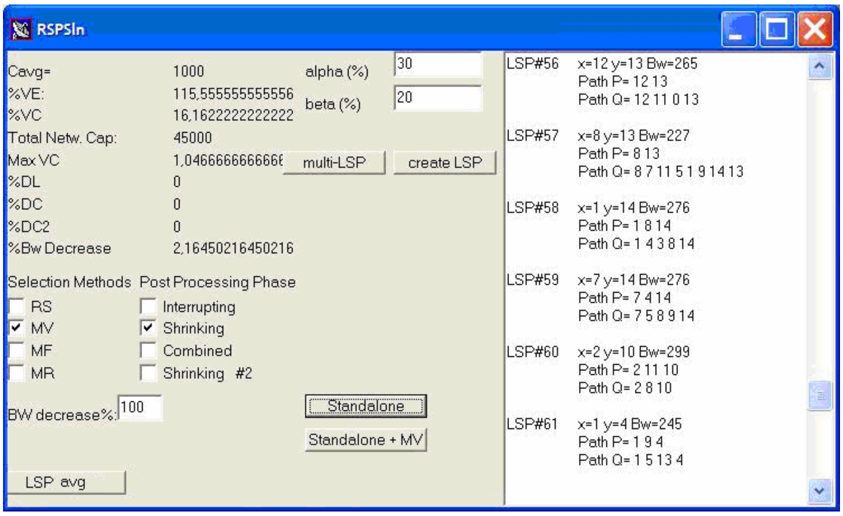

Figure 4-1 Main GUI of RSPSln

Checkboxes located on below left can be organized in the form of three groups. We herewith title group 1, for the ones on which 'heuristic approaches' are checked. Once the button is checked, the assigned sequential search is then carried out according to that approach and serves the results. (Figure 4-2).

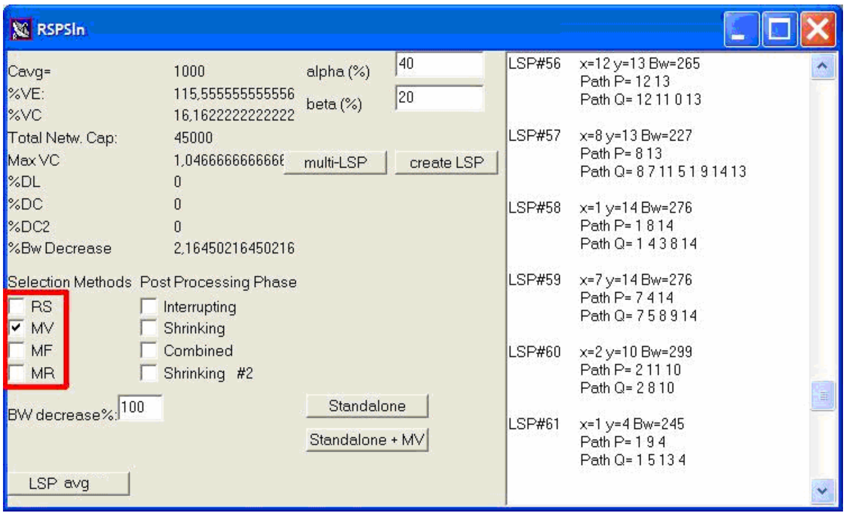

Figure 4-2 Selection Methods Checkboxes

On the other hand, group 2 checkboxes reveal the type of post processing procedures subject to a suitable approach when there exists no feasible solution. By this means, trials with the same initialization steps are checked with different post processing approaches on the same time. This procedure is disclosed in Figure 4-3.

Moreover, there is an input-box near these checkboxes, which one can specify the percentage of decrease when the combined or the shrinking approach is selected.

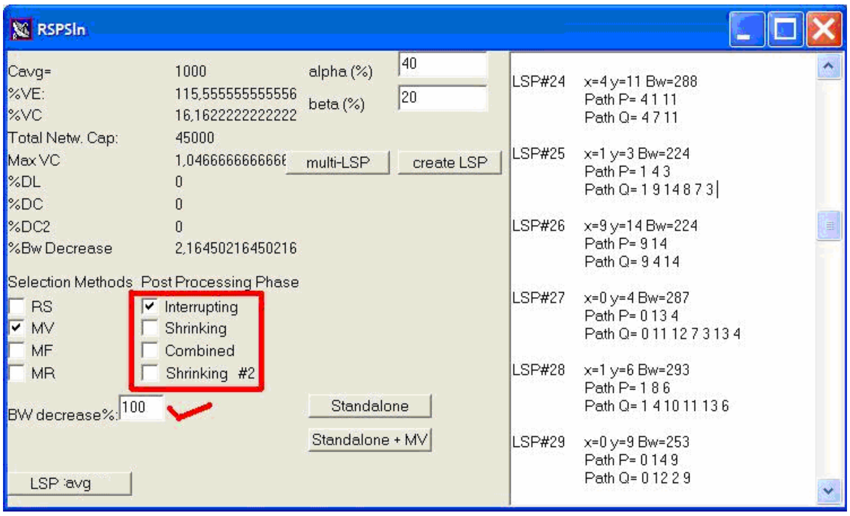

Figure 4-3 Postprocessing Checkboxes

Lastly, group 3 buttons specify the situation when violation occurs. Explicitly speaking, they specify the approaches given on group 1 by acquiring them either on their own or by supervising by minimal violation approach when there exists violation during sequential procedures (Figure 4-4).

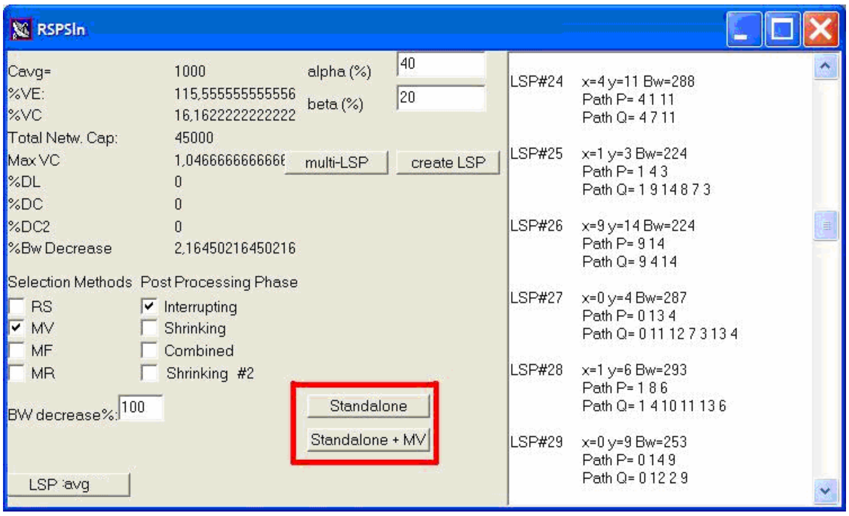

Figure 4-4 Run Buttons

The button which is titled as ‘LSP average’ and which is located on extreme below right, enable the user to reveal average of the evaluated results. Once this button is pressed, the user can evidently select an LSP result among numerous alternatives and respective averages from the relevant window. Details can be traced on Figure 4-5.

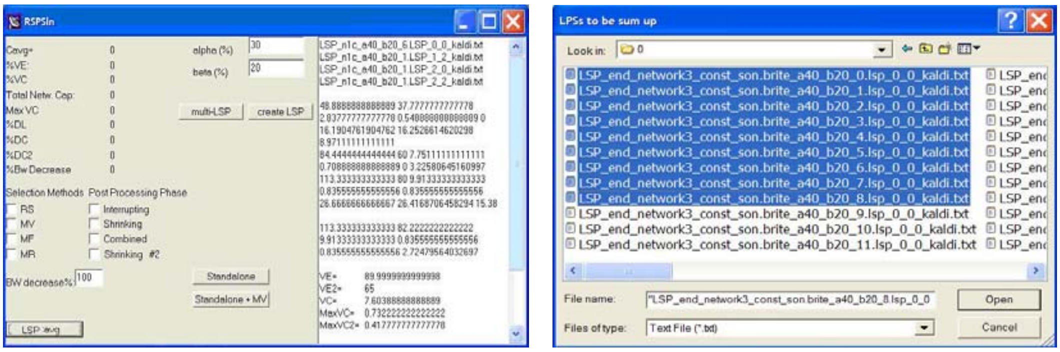

Figure 4-5 LSP Average Button

4.2. Implementation and Sub-Components

Borland C++ Builder is used as the development environment and C++ programming language is used for programming. A Pentium III-1400 MHz PC with 512 MB RAM is used during development and for the benchmarking of the proposed approaches.

The C++ version of BRITE 1.0 [29] is used for generating random networks. The command line interface of BRITE is used during the generation and all topology generation parameters are specified in a configuration file which is manually created. The command line interface directly involves the BRITE generation engine. It must receive as input a configuration file, a location for an export file and a seed file. BRITE uses pseudo-random numbers at different points during the generation process. For example, nodes are placed randomly in the plane, nodes are interconnected according to certain probability function, etc. The seed file contains seeds to initialize the pseudo-random number generator independently each time it is required. All topology generation parameters are specified in the configuration file which is manually created. Finally, when BRITE is started to build the topology with the given files, a status window, as shown in Figure 4-6, appears that details the topology generation process and the BRITE output is formed.

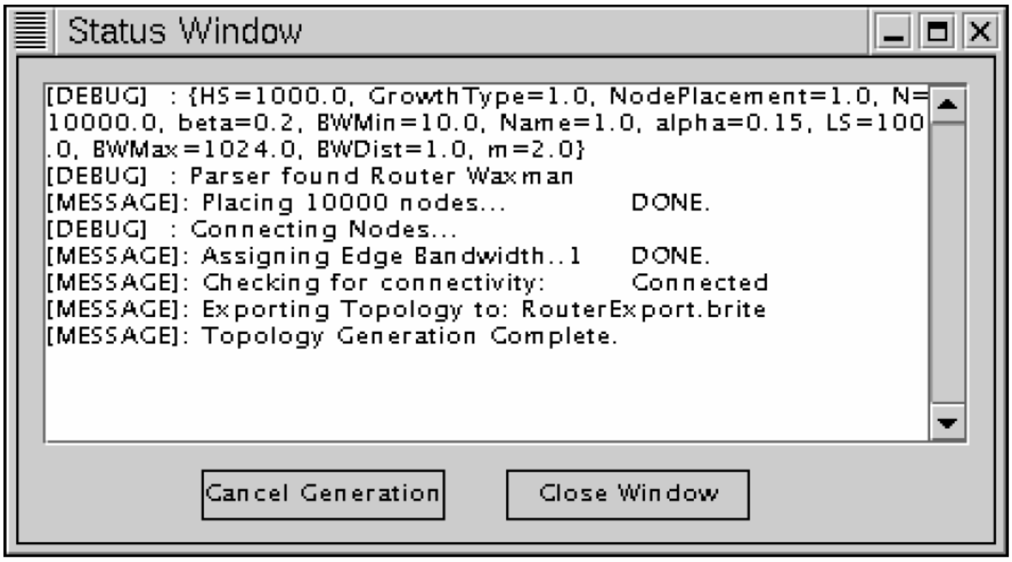

Figure 4-6 Status Window Showing Generation Process

A BRITE formatted output file which is used for the creation of LSPs contains information about the nodes and edges, and information specific to the model used to generate the topology. As an example, the output file shown in Figure 4-7 corresponds to a Waxman topology of 5 nodes and 8 edges.

Dijkstra's algorithm [Appendix A] is used for finding the primary and secondary paths of the LSPs. Since CSPF is implemented during the decision of primary paths, a standard Dijkstra minimum hop path computation is performed. But small modifications are made to apply the algorithm for finding the secondary paths (i.e. the paths found by SWP). Dijkstra's algorithm is modified by just changing the criteria which selects the next node added to set M, so that the next node to be added is the node that is connected to set M with largest bandwidth.

```
Topology: ( 5 Nodes, 8 Edges )
Model ( 1 ): 5 1000 100 1 1 2 0.15 0.2 1 10 1024

Nodes: (5)
0 216.00 663.00 3 3 -1 RT_NONE
1 347.00 333.00 3 3 -1 RT_NONE
2 384.00 926.00 3 3 -1 RT_NONE
3 27.00 309.00  4 4 -1 RT_NONE
4 212.00 187.00 3 3 -1 RT_NONE

Edges: (8):
0 2 0 312.08 1.04 10.00 -1 -1 E_RT_NONE
1 2 1 594.15 1.98 10.00 -1 -1 E_RT_NONE
2 3 1 320.90 1.07 10.00 -1 -1 E_RT_NONE
3 3 2 712.84 2.38 10.00 -1 -1 E_RT_NONE
4 4 0 476.02 1.59 10.00 -1 -1 E_RT_NONE
5 4 3 221.61 0.74 10.00 -1 -1 E_RT_NONE
6 0 3 401.29 1.34 10.00 -1 -1 E_RT_NONE
7 1 4 198.85 0.66 10.00 -1 -1 E_RT_NONE
```

Figure 4-7 Output Format of a Topology with 5 Nodes and 8 Edges

CHAPTER 5

COMPUTATIONAL STUDY

This chapter introduces the results of several experimental tests carried out to verify the performance of the heuristic algorithms proposed in this work. In order to investigate the algorithms, simulations were performed on a large number of problem instances. First, the problem instance generation and the characteristics of the investigated networks are described. Then, the parameters which constitute the base of the comparison of the algorithms are defined. Finally, the numerical results of the simulation scenarios are shown.

5.1. Instance Generation

All of the presented approaches are simulated on the network topologies shown in Table 5-1. Network-1 can be considered as a small network with equal edge capacities. Network-2 is also a small one but with all edges bearing a capacity value uniformly distributed between 500Mbps and 1500Mbps. Network-3 is a larger network with equal edge capacities and the Network-4 is again a large one with the edge capacity values uniformly distributed.

Once these characteristics of the networks are determined, then the graphs are generated by a universal topology generation tool called BRITE described in detail in [29]. Under certain configuration of the parameters, BRITE generation model is equivalent to Waxman [28] which is one of the most common random graph models.

Table 5-1 Characteristics of the Investigated Networks

network number	|V|	|E|	node degree	edge capacities (Mbps)	average edge capacity (cavg)
1	15	45	3	Constant: 1000 for all	1000
2	15	45	3	Uniformly distributed between 500 and 1500	835
3	25	75	3	Constant: 1000 for all	1000
4	25	75	3	Uniformly distributed between 500 and 1500	980

Therefore; by using this tool, four different connected Waxman graphs are obtained, in which nodes are placed randomly, bandwidths of edges are assigned and the node degree is obtained. For example Figure 5-1 shows the Waxman graph generated by this tool with the parameters of Network-4.

After the infrastructure is formed, the instances are created for the reroute sequence planning problem on these graphs in the way explained in [21]: First, a set of LSPs is generated using Constrained Shortest Path First (CSPF) algorithm for the given fixed networks. Then, these paths of the LSPs are changed to the optimized ones using Shortest Widest Path (SWP) algorithm for the purpose of acquiring the task to reroute the first set of LSPs on their new paths.

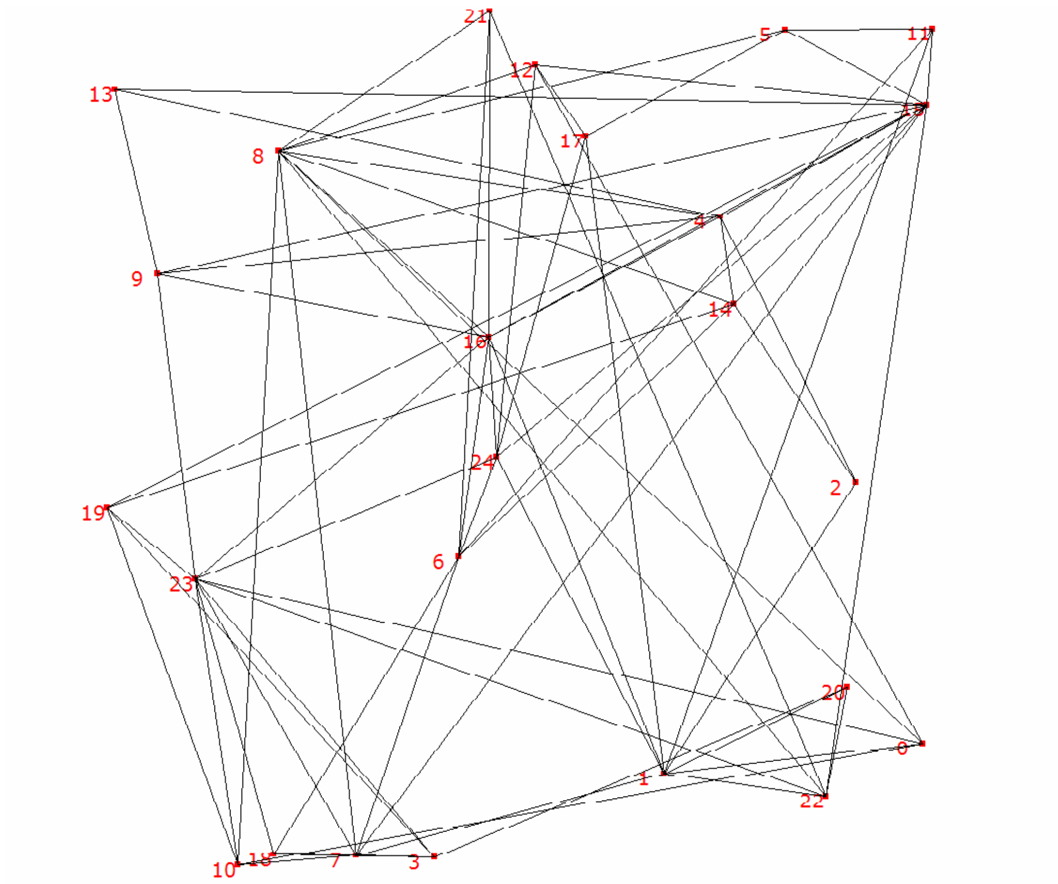

Figure 5-1 The Waxman Graph with the Characteristics of Network 4

CSPF algorithm [23, 24, 25] searches for a LSP that will satisfy the imposed constraints and then finds the shortest path among these feasible paths (i.e. selects the feasible path with the lowest number of hops). Since the only constraint taken into account in our work is the bandwidth, the paths found by CSPF algorithm are the shortest paths for a certain bandwidth demand between certain source-destination node pairs. However, a bandwidth CSPF algorithm in itself can not always yield network wide optimal paths, since it uses only local information. Therefore; SWP algorithm [22, 23] may be used to balance the load on the network. SWP algorithm selects the path with the maximum bandwidth among all feasible paths. If there are several such paths, the one with the minimum hop count is preferred. If again there are several such paths with the same hop count, one is randomly selected. Therefore; the paths of LSPs will be changed.

In this task, there are two control parameters that are utilized in "problem instance generation" procedure. These are defined as α and β successively and the parameter α controls the free capacity in the network, whereas the parameter β enables the demands of the created LSPs. Once the parameters α and β are determined for fixed networks then, one can proceed for the creation of LSPs with these acquired values. Lastly, alternative routes are adopted via the same network for the purpose of optimization of the network. This procedure is accomplished according to the means given below.

Since the topology of the network is given with fixed link capacities, the first task is to establish a number of LSPs in the network. For this, primarily the source and destination nodes of the subsequent LSP demands are chosen randomly. Let $b_{max} = \beta * c_{avg}$ where c_{avg} is the average edge capacity of the network shown in Table 5-1. Then select a value between $b_{max}/4$ and b_{max} randomly for the purpose of assigning a certain demand for each particular LSP whose source and destination nodes are already determined. Naturally as β decreases, the demand value of LSPs shrinks. Hence more LSPs will be generated since a saturation is intended within the quoted network.

Once the demand and source-destination nodes assignment processes are finalized, these LSPs are routed by CSPF algorithm. In case of inexistence of feasible path for the actual LSP demand, the procedure skips the LSP and generates another one, then retries the routing procedure. Alternatively if a feasible path is succeeded, the level of free capacity of the network is controlled and determined. If the free capacity does not exceed the level displayed by the control parameter α, the iteration of LSP generation is re-started. This iteration of LSP routing is repeated until the free capacity in the network decreases to the value shown by parameter α. Thus, there may be several LSP demands between each node-pair, which represents the current trend of distinguishing more types of traffic classes. To sum up, the problem instance generation has two parameters that define a problem class for a given network topology, namely the desired remaining free capacity *(α)* in the network after the CSPF routing and the relative demand size *(β)*. However, it is worth noting that the above problem instance generation results in remaining free capacity with great dispersion on the edges, i.e., there are some under utilized and fully utilized edges even if the value of α is relatively great.

After a feasible routing of the LSPs by CSPF is obtained and an LSP recalculation using the SWP algorithm is performed successively to get a set of new LSPs instead of the old ones, this means that reroute sequence planning problem instances are created.

We have defined four problem classes for the computational testing of the algorithms, by applying (i) *α = 40%, β = 20%, (ii) α = 40%, β = 30%, (iii) α = 60%, β = 20%, and (iv) α = 60%, β = 30%.* Here, two values are used for *α;* 40%, for high load and 60% for small load and similarly two values are used for *β*; %20, for small flows and 30% for large flows. Fifty random instances of the reroute sequence problem were generated for each of the four network topologies of Table 5-1 with each of the above four parameter combinations, giving 4*4 = 16 different scenarios and 50*16 = 800 investigated test instances in total for a single approach.

Random instances of the rerouting sequence are generated for each of the four network topologies. The results are averaged for the four networks and the test instances.

The investigated metrics are:

a) Edge Violation (*VE%*): the ratio of the violated edges to the total number of edges

b) Capacity Violation (*VC%*): the ratio of summed capacity violation on the edges to the total capacity in the network (that is the sum of edge capacities)

c) Maximal Violation (*maxVC%*): the greatest edge violation in percentage where the edge violation is defined by the maximal capacity excess during the rerouting action compared to the total capacity (corresponding to the maximal reservable bandwidth) of the edge

d) Success Ratio (*SR%*) : the ratio of the number of cases without capacity violations to the total number of examined test instances

e) Deallocated LSPs (*DL%*) : the ratio of the deallocated LSPs to the total number of LSPs

f) Deallocated capacity (*DC%*) : the ratio of the deallocated LSP capacity to the total LSP capacity (that is the sum of the bandwidths of LSPs)

g) Bandwidth Decrease (*BWd%*) : the ratio of the amount of decrease in the bandwidth of LSPs to the total LSP capacity

We have run the presented four methods for the generated test instances. The results showed, as it has been expected, that none of the methods were able to find a reroute sequence without temporary edge capacity violation for most of the generated test instances. Actually, the underlying problem is NP-hard. More significantly, with the limited problem setting discussed in this work that the LSP reroutes must be performed in one step, i.e., without using temporary paths, it is very probable that there are no feasible solutions at all.

In consequence of the above situation, we have to aim to solve the generated instances of the reroute sequence planning problem with either temporarily deallocating some old LSPs or degrading the bandwidths of some LSPs or the combination of both. In this case we solve the problem with temporary edge capacity violations and then select a small subset of old LSPs that are deallocated at the beginning of the reroute process to facilitate the feasible reroute of the remaining ones or decreasing the bandwidths of some LSPs in order to eliminate the violations.

5.2. Results

Obtained results given on tables are averages for the four networks and for the 50 test instances. In other words, a certain value of (*α, β*) pair with respect to a problem class of a chosen method, is simply the average of overall 200 values leading to each 50 values for the four different networks. Moreover, the last row of the tables contains averages for all instances.

The first two tables are entirely created on self application of heuristic approaches in case of a violation occurred during the reroute sequence calculation. Remaining ones comprise refinement procedure applied to these heuristic approaches, (i.e., proposing reroute of LSP via Minimal Violation approach in case of a violation) and created by this means.

Once the table results are examined for each LSP selection method, the reader will observe the existence of the amount of violated edges (*VE %*) together with the amount of total violated capacity (*VC %*) values in Table 5-2. On the other hand, Table 5-3 demonstrates the success ratio (*SR %*) together with the maximal violation amount occurred on an edge during the rerouting process for the chosen method. Nevertheless, this table is arranged in such a manner - that the average values are herewith repeated – so that the general behavior of each selection method can be well understood.

Table 5-2 Temporary edge violations for the methods

problem class (α,β)	LSP selection method							
	RS		MV		MF		MR	
	VE%	VC%	VE%	VC%	VE%	VC%	VE%	VC%
40,20	83,52	36,61	79,66	21,14	85,32	23,62	78,80	23,29
40,30	76,62	32,79	72,09	20,67	77,77	23,19	71,76	24,53
60,20	16,86	3,20	4,93	0,35	7,20	0,57	1,94	0,11
60,30	20,84	4,29	9,34	0,94	14,07	1,52	6,18	0,80
average	49,46	19,22	41,51	10,78	46,09	12,23	39,67	12,18

Observing the results, the success ratio of Random Selection is nearly zero, i.e., it could not succeed in solving any of the problem instances. On the other hand, Most Routable is better than the other greedy approaches in terms of success ratio. This result is not surprising because it looks forward one rerouting step. Moreover, as it shall be foreseen, the case in which feasible solutions substantially occur is when the corresponding (*α, β*) problem instance yield to (60, 20). The reason for this is obvious; i.e it preserves the case with largest free capacity in the network and lowest bandwidth demand for LSPs. When the relevant metrics related to violation are examined, it shall be observed that the total violated capacity will be minimal in Minimal Violation method, but it will be maximal in the Random Selection Method.

Table 5-3 Average values of the methods

LSP selection method	SR%	maxVC%	VE%	VC%
RS	0,13	52,69	49,46	19,22
MV	15,88	32,34	41,51	10,78
MF	5,63	35,80	46,09	12,23
MR	31,88	28,93	39,67	12,18

Now, let us examine the results of the algorithms with refinement.

When Table 5-4 and Table 5-5 are examined, the influence of refinement to the algorithms can be observed. It should be noted that while the success ratio of the Most Rerouteable and Minimal Violation methods remains expectedly unchanged, there is an increase in feasible solution seeking ratios of Random Selection and Maximal Freeing methods, respectively. Nevertheless, as expected, relevant metrics related to violation of all methods tend to decrease. Besides, it is observed that the refinement is particularly influential on low loaded networks.

Table 5-4 Temporary edge violations for the methods with refinement

problem class (α,β)	LSP selection method							
	RS		MV		MF		MR	
	VE %	VC %	VE %	VC %	VE %	VC %	VE %	VC %
40,20	79,24	19,93	79,66	21,14	79,64	20,60	78,03	18,63
40,30	71,70	19,90	72,09	20,67	71,99	20,20	71,06	18,86
60,20	2,51	0,11	4,93	0,35	2,64	0,12	1,80	0,07
60,30	6,74	0,57	9,34	0,94	7,11	0,61	5,73	0,48
average	40,05	10,13	41,51	10,78	40,35	10,38	39,15	9,51

Table 5-5 Average values of the methods with refinement

LSP selection method	SR%	maxVC%	VE%	VC%
RS	26,75	27,54	40,05	10,13
MV	15,88	32,34	41,51	10,78
MF	29,13	27,65	40,35	10,38
MR	31,88	26,41	39,15	9,51

Differing from the previous act, in cases where algorithms are run with the refinement, post processing is pursued in the absence of a feasible solution.

Similar to the edge violation figures, the Table 5-6 being the first among these tables, shows the amount of deallocated LSPs (*%DL*) and the ratio of deallocated LSP capacity to the total rerouted capacity (*%DC*). The results are again averages for the four networks and 50 test instances. The figures show that the Minimal Violation LSP selection performed worst with respect to both indicators, while the Most Rerouteable method performed best. The differences in the averages are not so significant and the values for both *DL%* and *DC%* are small in the case of networks with small load.

Table 5-6 Temporary LSP Deallocations during the Reroute Process

problem class (α,β)	LSP selection method							
	RS		MV		MF		MR	
	DL %	DC %	DL %	DC %	DL %	DC %	DL %	DC %
40,20	28,27	32,58	28,66	33,23	28,23	32,75	27,86	31,93
40,30	33,05	38,22	33,67	39,36	33,47	38,85	32,50	37,75
60,20	2,01	2,08	3,81	4,13	1,91	1,89	1,31	1,33
60,30	5,90	6,63	8,03	9,11	6,18	6,79	4,65	5,27
average	17,31	19,87	18,54	21,46	17,44	20,07	16,58	19,07

In the mean time, both Table 5-7 and Table 5-8 involve successively the results derived through shrinking approach which is the alternative post-processing approach. These tables show the ratio of the amount of decrease in the bandwidth of LSPs to the total LSP capacity (*BWd %*). While evaluating the *BWd %* figures in Table 5-7, it should be noted that the bandwidths of all LSPs are uniformly decreased in ratio with a critical value of (*1 / (1 + maxVC)*). On the other hand, in the approach we proposed, only the LSPs which involve one of violated edges occurring on either the old or new path are selected and respective bandwidth values of only these selected LSPs are decreased uniformly with the same ratio and these results are presented in Table 5-8.

Significantly speaking the values in the shrinking approach which we have proposed, as revealed on Table 5-8, are less than the ones given on Table 5-7 when compared in terms of *BWd%*. Therefore; further decrease on respective bandwidths of LSPs obtained and still there is no violation during the rerouting processing.

Table 5-7 Temporary Bandwidth Degradation during the Reroute Process (Approach 1)

problem class (α,β)	LSP selection method											
	RS			MV			MF			MR		
	DL%	DC%	BWd%	DL%	DC%	BWd%	DL%	DC%	BWd%	DL%	DC%	BWd%
40,20	0,00	0,00	30,36	0,00	0,00	32,02	0,00	0,00	30,39	0,00	0,00	29,72
40,30	0,00	0,00	34,05	0,00	0,00	35,91	0,00	0,00	34,47	0,00	0,00	33,59
60,20	0,00	0,00	2,25	0,00	0,00	8,41	0,00	0,00	1,99	0,00	0,00	1,61
60,30	0,00	0,00	6,97	0,00	0,00	11,47	0,00	0,00	6,75	0,00	0,00	5,68
average	0,00	0,00	18,41	0,00	0,00	21,95	0,00	0,00	18,40	0,00	0,00	17,65

Table 5-8 Temporary Bandwidth Degradation during the Reroute Process (Approach 2)

problem class (α,β)	LSP selection method											
	RS			MV			MF			MR		
	DL%	LSP%	BWd%	DL%	LSP%	BWd%	DL%	LSP%	BWd%	DL%	LSP%	BWd%
40,20	0,00	97,31	29,91	0,00	96,27	31,32	0,00	96,48	29,82	0,00	95,81	29,02
40,30	0,00	96,46	33,53	0,00	94,63	34,75	0,00	95,26	33,51	0,00	94,30	32,50
60,20	0,00	22,92	1,62	0,00	19,79	3,06	0,00	14,72	1,28	0,00	8,29	0,70
60,30	0,00	43,51	5,91	0,00	32,59	6,47	0,00	30,35	4,74	0,00	21,77	3,47
average	0,00	65,05	17,74	0,00	60,82	18,90	0,00	59,20	17,34	0,00	55,04	16,42

Moreover, the *LSP%* on Table 5-8 denotes the ratio of the LSPs whose bandwidth values are decreased to the total number of LSPs. When the results are observed, it can be seen that the number of LSPs whose bandwidths remain untouched are high in the networks with small loads and small demands of LSPs and nearly 40% of LSPs preserves their original capacities in the average.

Lastly, the tables below show the results when the combined approach is chosen as the post processing method. Here the tables are formed for the values of

50, 10 and 0 percent of *maxVC*, respectively. As we mentioned before, when the critical value *maxVC* is used during the degradation of LSPs' bandwidths, there is no violation in one edge during the reroute sequence. However, if we decrease the bandwidths of LSPs with a certain percent of *maxVC*, then there is a need to deallocate some LSPs in order to get no violation.

The tables below involve the *DL%* and *DC%* values to the corresponding *BWd%* values for each LSP selection method and problem class (*α, β*). Observing these, it can be seen that there are not great differences between the results for the same problem class, but different LSP selection method.

Table 5-9 Combined Approach (Max VC * 50%)

problem class (α,β)	LSP selection method											
	RS			MV			MF			MR		
	DL%	DC%	BWd%	DL%	DC%	BWd%	DL%	DC%	BWd%	DL%	DC%	BWd%
40,20	8,93	9,76	18,08	9,06	9,85	19,25	9,59	10,34	18,14	8,72	9,44	17,66
40,30	10,49	11,85	20,91	10,49	11,90	22,16	10,57	11,95	21,10	10,14	11,68	20,50
60,20	0,87	0,89	1,17	1,63	1,67	3,55	0,92	0,91	1,05	0,78	0,78	0,85
60,30	2,66	2,96	3,66	3,82	4,27	6,39	2,94	3,15	3,72	2,23	2,41	3,12
average	5,74	6,36	10,95	6,25	6,92	12,84	6,01	6,59	11,00	5,47	6,08	10,53

Table 5-10 Combined Approach (Max VC * 10%)

problem class (α,β)	LSP selection method											
	RS			MV			MF			MR		
	DL%	DC%	BWd%	DL%	DC%	BWd%	DL%	DC%	BWd%	DL%	DC%	BWd%
40,20	23,94	27,25	4,30	24,09	27,62	4,61	24,09	27,62	4,30	23,28	26,49	4,17
40,30	28,00	32,50	5,12	28,25	32,89	5,49	28,54	33,16	5,17	27,31	31,91	5,01
60,20	1,52	1,59	0,24	1,75	1,78	0,30	1,53	1,53	0,22	1,10	1,11	0,18
60,30	4,83	5,46	0,84	6,85	7,72	1,41	5,26	5,72	0,81	3,93	4,47	0,68
average	14,57	16,70	2,62	15,24	17,51	2,95	14,85	17,01	2,63	13,90	16,00	2,51

Table 5-11 Combined Approach (Max VC * 0%)

problem class (α,β)	LSP selection method											
	RS			MV			MF			MR		
	DL%	DC%	BWd%	DL%	DC%	BWd%	DL%	DC%	BWd%	DL%	DC%	BWd%
40,20	28,27	32,58	0,00	28,66	33,23	0,00	28,23	32,75	0,00	27,85	31,93	0,00
40,30	33,05	38,22	0,00	33,67	39,36	0,00	33,47	38,85	0,00	32,47	37,75	0,00
60,20	2,01	2,08	0,00	3,81	4,13	0,00	1,91	1,89	0,00	1,30	1,33	0,00
60,30	5,90	6,63	0,00	8,03	9,11	0,00	6,16	6,79	0,00	4,65	5,27	0,00
average	17,31	19,87	0,00	18,54	21,46	0,00	17,44	20,07	0,00	16,57	19,07	0,00

It should be noted that Table 5-11 bears a special case, or in other words, preserves the same results with the interrupting approach. Taking 0% percent of *maxVC* during the shrinkage of bandwidth of LSPs means that how many LSPs are needed to be interrupted in order to eliminate all the violations during the rerouting process when there is no decrease in the bandwidth of LSPs. Hence, this is nothing but the interrupting approach itself.

Finally, Table 5-12 has been derived with all average values obtained via the combined approach in every relevant table. The first row of this table denotes the satisfactory working of the sequence when only bandwidth shrinkage is applied. The last row, on the other hand, denotes the values acquired without any bandwidth decrease. Moreover, while proceeding from top to bottom, the amount of shrinkage of bandwidths is decreased whereas number of LSPs and the amount of capacity that will be deallocated increases for the corresponding row.

Table 5-12 Combined Approach with Different Max VC Values

max VC (%)	LSP selection method											
	RS			MV			MF			MR		
	DL%	DC%	BWd%	DL%	DC%	BWd%	DL%	DC%	BWd%	DL%	DC%	BWd%
100	0,00	0,00	18,41	0,00	0,00	21,95	0,00	0,00	18,40	0,00	0,00	17,65
50	5,74	6,36	10,95	6,25	6,92	12,84	6,01	6,59	11,00	5,47	6,08	10,53
10	14,57	16,70	2,62	15,24	17,51	2,95	14,85	17,01	2,63	13,90	16,00	2,51
0	17,31	19,87	0,00	18,54	21,46	0,00	17,44	20,07	0,00	16,57	19,07	0,00

Network operator, consequently, is apt to select a combination to optimize the working situation of the reroute sequence. This evidently will enable an utter flexibility for the operator. With the results displayed, an average of 10% decrease on bandwidths of LSPs will mean an optimality that will be derived by interrupting 5% of relevant LSPs. However, it is very important to note that these results depend on the instance generation to a great extent and in another context the algorithms might perform differently.

CHAPTER 6

CONCLUSION

The validity and success of next generation IP networking are based on the ability to offer and support QoS to customers. It is obvious that Traffic Engineering (TE) is critical for this as well as for efficient network resource utilization and operation. In this work, we investigate the reroute sequence planning of LSPs in MPLS network, which can be realized as an alternative recent means of the TE tool that helps a global path optimization.

The reroute sequence planning of LSPs is an awkward task with great practical impact. We deal with a simplified problem setting when the LSPs have to be rerouted in a single step as in the previous works. The related optimization problem is NP-hard and the real-world problem sizes are huge, therefore only heuristic algorithms are examined for seeking efficient results.

The contributions of the thesis are threefold. One is the detailed comparison of the heuristic algorithms found in the literature in order to get feasible solutions to RSP problem. Secondly, in case of infeasible solutions, post-processing algorithms are investigated from practical point of view and a new post-processing approach is offered. Finally, the last contribution is the development of an integrated software tool for the analysis of RSP problem which may also form the core of a more complicated design tool for reroute sequencing problem later.

We hereby studied four greedy methods in the literature for solving the simple reroute sequence planning problem. The algorithms were evaluated through empirical comparisons involving randomly generated instances belonging to different problem classes. These algorithms are run in two different ways; first the LSP selection is done by the corresponding greedy selection rule whether violation is occurred during the reroute sequence calculation or not and in the second one, the Minimal Violation method is used unless there are any routable LSPs during rerouting.

Both of the methods perform worse in terms of success ratio, i.e. they could not succeed in solving most of the problem instances generated. Also it must be noted that this is valid for even moderate networks, i.e. some RSP problem instances can not be solved by these simple algorithms already at 60% network load (which means 40% spare capacity on average). Consequently the actual share of the success is originated from the results when the free capacity of the network ranges around 60%.

However, it should be noted that when the algorithms are refined the success ratio of the Most Rerouteble and Minimal Violation methods remain unchanged, but there is an increase in feasible solution success ratios of the Random Selection and Maximal Freeing methods, respectively. Nevertheless, the amount of violation occurred during the rerouting is decreased for all the selection methods.

These results serve as a motivation for further investigations from another point of view, namely, reconfiguration of LSPs while allowing some interruption or degradation or both interruption and degradation of traffic during the rerouting process is needed when the previously described heuristics can not find any feasible solutions.

Following these assessments, while searching reroute sequence when despite refined algorithms are utilized but feasible solution is not acquired, choosing temporary service degradation as post-processing sounds a suitable observation. It shall naturally be worth noting that the amount of this degradation must be kept so small that the network users do not perceive it. When this amount does not suffice, tending towards the interrupting approach will be influential. Nevertheless, while carrying out this degradation procedure, instead of the previous mentioned approach (i.e. lowering bandwidths of all LSPs) using the proposed scheme of shrinking (i.e.

lowering bandwidths of LSPs covering only the edges causing violation) will be more suitable. Even more, in light networks, since such LSPs will be less, the method offered will be more effective.

All these tests to verify the performance of the heuristic algorithms investigated in this work are performed on graphs randomly generated for different constraint sets by using the software tool RPSln which is developed for this purpose. RSPSln software includes a label switched paths (LSPs) module with a graphical user interface (GUI) that generates a set of LSPs in the given fixed network and a test module for both reroute sequence planning approaches and post processing approaches for the infeasible cases.

In summary, we propose to use the Most Reroutable selection method in practice. Furthermore, when the problem can not be solved without violations, we suggest to recalculate the sequence by again Most Reroutable method, but in this case LSP selection is done by the approach Minimal Violation (MR) if there is currently no LSP that can be rerouted without exceeding the capacity constraints. This situation will lower the amount of capacity violation constituted on edges. Under these circumstances, by making use of shrinking approach we offered, temporary degradation of LSPs' bandwidths will be effective to solve the reroute sequence planning problem.

As future work, the methods to solve RSP problem and post-processing approaches should be investigated further in more detail. Maybe trying other general heuristics such as tabu search or simulated annealing will improve the success probability of RSP. Moreover, updating the sequence after each deallocation may increase the performance in infeasible cases. Finally, allowing temporary paths or partial reroutes during the rerouting process may be good solutions in order to decrease the amount of temporary edge capacity violation.

REFERENCES

1. Wroclawski, J., “The use of RSVP with IETF integrated service” RFC 2210, IETF, (1997).

2. Zhang, L., Deering, S., Estrin, D., Shenker,S., and Zappala, D., “RSVP: a new resource ReSerVation protocol,” IEEE Network, Volume 7, (1993), pages 8-18.

3. Nichols, K., “Definition of the Differentiated services Field in the IPv4 and IPv6 headers” RFC 2474, IETF, (1998).

4. Blake, S., “An Architecture for Differentiated Services Field”, RFC 2475, IETF, (1998).

5. Rosen, E., Viswanathan, A. and R. Callon, “Multiprotocol Label Switching Architecture”, RFC 3031, IETF, (2001)

6. Brittain, P., Farrel, A., “MPLS Traffic Engineering: A Choice Of Signaling Protocols”, White Paper, Data Connection Ltd., (2000), Available at: http://www.dataconnection.com

7. Viswanathan, A.; Feldman, N.; Wang, Z.; Callon, R., ”Evolution of Multiprotocol Label Switching” IEEE Communications Magazine, Volume 36, Issue 5, (1998), pages 165 –173.

8. “MPLS Tutorial”, International Engineering Consortium (IEC) Tutorial Section on MPLS, IEC, Available at: http://www.iec.org/online/tutorials/mpls

9. Rekhter, Y., Davie, B., Katz, D., Rosen, E., Swallow, G., Cisco Systems Tag Switching Architecture Overview, RFC 2105, 1997.

10. Awduche, D., “MPLS and traffic engineering in IP networks”, IEEE Communications Magazine, Volume 37, Issue 12, (1999), pages 42-47.

11. Awduche, D., Malcolm, J., Agogbua, J., O’Dell, M., and McManus, J., “Requirements for Traffic Engineering Over MPLS”, Internet Engineering Task Force, RFC 2702, (1999).

12. Xiao, X., Hannan, A., Bailey, B., Ni, L. M., “Traffic engineering with MPLS in the Internet”, IEEE Network, Volume 14, (2000), pages 28-33.

13. Jozsa, B. G., Kiraly, Z., Magyar, G., Szentesi, A., “An Efficient Algorithm for Global Path Optimization in MPLS Networks”, Submitted to Optimization and Engineering 2, (2001), pages 321-347.

14. Chung, J., “Analysis of MPLS traffic engineering”, Circuits and Systems, Proceedings of the 43rd IEEE Midwest Symposium on, Volume 2, (2000), pages 550-553.

15. “MPLS- An introduction to multiprotocol label switching”, White Paper, Nortel Networks, (2001), Available at: http://www.nortelnetworks.com

16. MPLS Resource Center, Available at: http://www.mplsrc.com

17. Jozsa, B. G., Magyar, G., “Reroute Sequence Planning for Label Switched Paths in Multiprotocol Label Switching Networks”, The 6th IEEE Symposium on Computers and Communications, Hammamet, Tunisia, (2001), pages 319-326

18. Awduche, D., Jabbari, B., “Internet Traffic Engineering Using Multi-protocol Label Switching (MPLS)”, Computer Networks, (2002), pages 111-129.

19. Long, K., Zhang, Z., Cheng, S., “Load Balancing Algorithms in MPLS Traffic

Engineering", IEEE Workshop on High Performance Switching and Routing, (2001), pages 175-179.

20. Girish, M.K, Zhou, B., Hu, J.Q., "An Algorithm for Rerouting in Traffic Engineering of MPLS Based IP Networks", The 3rd IEEE Workshop on IP Operations and management (IPOM 2003), Missouri, USA, (2003), pages 115-118.

21. El-Hawary, M. F., El-Derini, M. N., Aly, H. H., "Heuristics for Rerouting of Label Switched Paths in MPLS Based IP Networks", The 8th IEEE International Symposium on Computers and Communication (ISSC 2003), Antalya, Turkey, (2003), pages 957-962.

22. Wang, Z., Crowcroft, J., "QoS Routing for Supporting Multimedia Applications", IEEE Journal on Selected Areas in Communications, Volume 14, No. 7, (1996), pages 1228-1234.

23. Calle, E., "Enhanced fault recovery methods for protected traffic services in GMPLS networks", PhD Thesis, Department of Computer Science, Universitat de Girona, (2004), Available at: http://www.tdx.cesca.es/TESIS_UdG/AVAILABLE/TDX-0920104-095653//teco.pdf

24. Juva, I., "Analysis of Quality of Service Routing Approaches and Algorithms", Master's Thesis, Department of Engineering Physics and Mathematics, Helsinki University of Technology, (2003), Available at: http://www.tct.hut.fi/tutkimus/ironet/papers/Juva_thesis.pdf

25. Szviatovszki, B., Szentesi, A., Jüttner, A., "Minimizing re-routing in MPLS networks with preemption-aware constraint-based routing", Computer Communications 25, (2002), pages 1076-1084.

26. Waxman, B. M., "Routing of multipoint connections", IEEE Journal on Selected Areas in Communications, Volume 6, No. 9, (1988), pages 1617-1622

27. Jozsa, B. G., Makai, M., "On the solution of reroute sequence planning problem in MPLS networks", Computer Networks, Volume 2, No.2, (2003) , pages 199-210.

28. Jozsa, B. G., "Reroute Sequence Planning for Protected Traffic Flows in GMPLS Networks", IEEE International Conference on Communications, Volume 5, New York, (2002), pages 2702-2706.

29. Medina, A., Lakhina, A., Matta, I., Byers, J. W., "BRITE: An Approach to Universal Topology Generation", IEEE Network Magazine, (2001), pages 346-356.

30. R. Haynal's ISP Page, Available at: http://navigators.com/isp.html

APPENDIX A

DIJKSTRA'S ALGORITHM

Let N(G), denoted by N for simplicity, be the set of nodes in the network, by M the set of nodes incorporated by the algorithm, and by A the set of links in the network. Let T denote N - M, the set of nodes yet to be incorporated.

When the algorithm starts set M contains only the source node s. The cost function C(n) is the weight of the path from node s to node n. For each node n in set T let the cost C(n) be the weight of link (s, n) and the predecessor pred(n) be node s. If there is no link (s,n), then C(n) = ∞ and pred (n) = 0.

The cost function values of nodes in set T are called temporary labels, and cost functions of nodes in set M are called permanent labels since they do not change any more during the execution of the algorithm.

Then, while the set of incorporated nodes M is different from the set of all nodes N:

Choose the node u with the smallest cost function of the nodes in T. Include node u to set M and remove it from set T. Then update the temporary labels for all the nodes remaining in set T by calculating the new cost functions. The label changes if a node n € T is reached through node u, which was just added to set M, with smaller cost than the current label on n. That is, if the cost of the path from s to u plus the link weight on link (u, n) is smaller than the current cost on the path from s to n, the label is changed. The predecessor of node n is then also changed to u.

The nodes are added one by one to set M until all nodes have permanent labels and the set T is empty. The algorithm then stops, having calculated the shortest path from node s to every other node in the network.

- M = { s }
- C(s) = 0, pred (s) = 0
- for each node n € (T)
  - – C (n) = w (s, n), pred (n) = s if (s; n) € A,
  - – C(n) = ∞ otherwise

- While N ≠ M
  - – Choose u € T so that C(u) = min {C(v) : v € T }
  - – M = M U { u }, T = T - { u }
  - – For each node n € T that is a neighbor of u
    - * if C(n) > C(u) + w(u, n) then
      - C(n) = C(u) + w(u,n), pred(n) = u