NOTE TO USERS

A Multipath Traffic Engineering Scheme with Path Protection for Real-time Applications in MPLS VPNs

Olivera Arezina

A Thesis

In

The Department

Of

Electrical & Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science
in Electrical & Computer Engineering
at Concordia University
Montréal, Québec, Canada

August 2005

Your file Votre référence
ISBN: 0-494-10231-4
Our file Notre référence
ISBN: 0-494-10231-4

Abstract

A Multipath Traffic Engineering Scheme with Path Protection for Real–time Applications in MPLS VPNs

Olivera Arezina

A growing number of corporations are considering IP Virtual Private Network (VPNs) for interconnection between different sites. Network–based IP VPNs, enable service providers with an IP backbone to offer VPN service on a large scale, in a scalable and manageable way. Multiprotocol Label Switching (MPLS) is one of the technologies used in network–based IP VPNs to establish backbone tunnels – label switched paths (LSPs) between an ingress – egress pair of nodes. In order to provide service differentiation, separate LSPs, that carry only real–time VPN traffic, can be deployed. These real–time LSPs can then be routed, based on the strict service level agreement (SLA) requirements that apply to real–time applications. To provide resilience in the face of failure, real–time applications can be supported by a

combination of a link/node protection and a path protection scheme. In this thesis we address the following traffic engineering problem: How to route primary and protection real–time LSPs so that the SLA requirements expressed in terms of latency, jitter, bandwidth guarantees, and resilience are met? We formulate a Mixed Integer Linear Programming (MILP) optimization problem that takes into account the SLA requirements in one step – the integrated model. The integrated model is NP-complete and has a large number of variables and constraints. We propose a two–step heuristic that can be solved in a context of a backbone MPLS VPN network. The heuristics offers two tradeoffs: one between the link utilization (service quality experienced by the traffic) and the number of LSPs (management complexity), and another between the maximum delay difference on primary and protection LSPs (service quality experienced by the traffic in the case of failure) and the overall reliability of the solution. We apply the heuristic to a simulation model that approximates a real–world backbone network. We explore service quality/management complexity and service quality/reliability dependencies in our simulation scenarios.

Acknowledgments

My sincere gratitude goes to my supervisors Professor Anjali Agarwal and Professor Brigitte Jaumard.

I thank Professor Anjali Agarwal for being supportive of my ideas, for her patience, encouragement, availability and understanding.

I consider myself lucky that I have met Professor Brigitte Jaumard at a critical time in my thesis research. Professor Jaumard provided much needed guidance and help in terms of the resources and expertise that facilitated the completion of the thesis. I am truly grateful...

A special whole-hearted thanks to my parents Vladimir and Draga Arezina for their unconditional love and support. They are and will always be my inspiration and the source of pride and strength. I also thank to my very special brother Milan Arezina and his family for their pure love and constant encouragement.

Thanks to all my fellow students at the University of Montreal and University of Concordia Linh, Alain, Hichem, Thomas, Romaric, Antoine, Vasu... for feeling accepted and for making the long work hours fun.

And in the end a special bubbly, rosy thanks to my husband Pedja Sekaric, for supporting my dream, for believing in me all the way through and for loving me.

Olivera Arezina, August 2005.

I dedicate this work to my parents, my husband and to two special little people, my niece Kristina and my nephew Marko

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Multiprotocol Label Switching (MPLS) started out as an answer to a need to speed up the complex route lookup based on a destination IP address, but it found its application in traffic engineering for backbone networks as well as in implementing a cost-effective VPN service [9].

Backbone networks have a high transmission rate which implies a high degree of traffic aggregation. The aggregates of traffic flows – traffic trunks, have different statistical properties than the flows that compose them. They have smaller short term fluctuation rates, and make a more efficient use of fixed capacity trunks over which they are routed. In the core network, they become a natural unit of traffic engineering [10]. IP technology based on the statistical multiplexing of packets does not offer a

mechanism for the manipulation of the traffic aggregates. MPLS technology provides that functionality by allowing the creation of virtual circuits across an IP network, although at a lower layer.

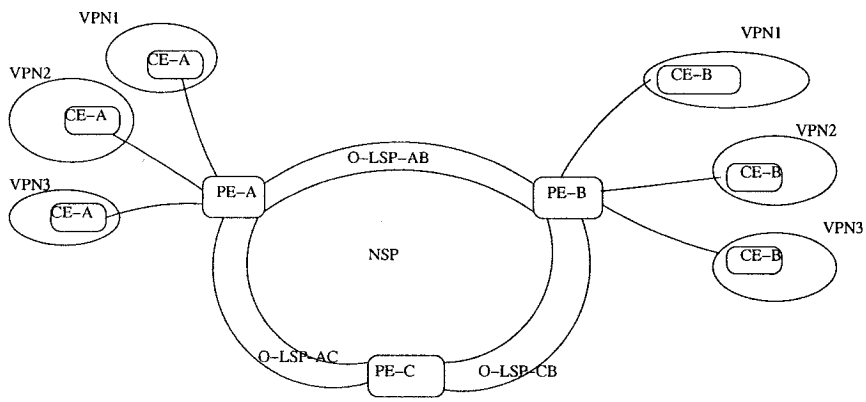

Figure 1.1: MPLS-based VPNs

Virtual Private Networks (VPNs) answer the need to connect distant users who trust each other and, at the same time, to isolate the traffic they exchange, from other users of the network. One of the basic business problems that are solved by a VPN is the creation of an Intranet between multiple sites that belong to the same organization. A Network Service Provider (NSP) can have several VPN customers. Consider a set of m customers that have various sites at different locations that they want to connect. To achieve this, m VPNs need to be provisioned. Clearly a scalable solution must scale well with the increase in the number of customers. MPLS-based VPNs offer such a solution through its label stacking mechanism. A customer data packet carries two labels when traversing the backbone network. The outer–top label directs the packet to the correct provider edge router (PE router) . The second–inner

label indicates how that PE router should forward the packet to the customer edge router (CE router). Let's assume that all the customers at location A connect to the PE–A router and all the customers at location B connect to the PE–B router, see Figure 1.1. All the packets of all the customers entering at location A with destination B will have the same outer label (use the same MPLS tunnel – LSP) to get to B. However, they will all have a distinct inner label. This is how the isolation among customer traffic is achieved. Effectively, this will result in the creation of an *outer LSP* connecting PE–A and PE–B, and a number of *inner VPN specific LSPs* carried within the outer one, see Figure 1.2. The outer LSP therefore becomes a tunnel that makes inner LSPs transparent to the intermediate LSRs which simplifies the forwarding tables at these LSRs as they are completely oblivious of the existence of various VPNs. This is how the scalability is achieved.

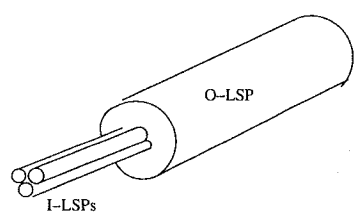

Figure 1.2: MPLS label stacking

In this thesis we will study the VPN traffic engineering problem. We assume that each VPN customer provides a traffic demand matrix whose elements are the bandwidth requirement between an ingress–egress pair of the VPN. The goal of the VPN traffic engineering problem is to find a route for each of these demands. We

are going to study the problem in the context of a backbone network that belongs to one NSP. We are also going to assume that significant amount of traffic generated by VPNs is real-time traffic in which case quality of service (QoS) becomes an important requirement for the VPN customer. This means that the MPLS VPN solution must be able to provide a service that require strict performance guarantees defined by SLAs and expressed in terms of latency, jitter, bandwidth guarantees, the resilience in the face of failure and downtime.

An example of a real–time application is VoIP. Possible problems that affect quality of voice communication in the Internet are packet loss and delay. Delay has a fixed and a variable part. The fixed part is comprised of the propagation delay and all the fixed transit delays incurred through the network. The variable part is due to the queuing delays and other possible delays introduced by the operation of the network elements. To achieve a good level of interactivity, end–to–end delay has to be below a given threshold (below 150ms) [13]. This translates into an important SLA parameter for backbone networks – maximum POP to POP delay, typically set between 50ms to 80ms [15]. There are two main causes of packet loss and packet delay: congestion and failure of network components. Congestion results in the delay variation through the increase in the queuing delays and packet loss due to the dropping of the packets when the queues are full. The failure of network components results in packet loss and the change to the fixed part of the delay due to the change in the path traversed by packets. To assure good performance of VOIP application both congestion and

network failures must be addressed. The same applies to other real–time applications.

MPLS traffic engineering (MPLS–TE) offers a solution to both congestion and network failures. By setting up LSPs along links with available resources, it ensures that sufficient bandwidth is available on the path. However current MPLS-TE mechanisms operate on traffic aggregates that are composed of traffic flows belonging to different service classes. These traffic flows will thus receive the same treatment in network elements. To allow for a class based differentiation at network elements, a traffic aggregate is split into multiple traffic trunks, each aggregating traffic flows that belong to the same service class. These class–based traffic trunks are then transported over separate LSPs that can potentially follow different paths through the network. This approach is termed MPLS DiffServ traffic engineering (DS–TE) [4]. When routing class–based traffic trunks, the requirements specific to that class such as bandwidth, preemption priority, limit to the portion of traffic from that particular class on a link, can be taken into consideration. Going back to the VPN traffic engineering, one possible approach is to map all the demands using the same ingress–egress pair onto one O-LSP. This O-LSP would then carry traffic belonging to different service classes. With the DS–TE mechanism in place, it is possible to split the O-LSP traffic aggregate into several class-based traffic trunks, and then solve the problem of traffic engineering of these traffic trunks separately for each class. Clearly different set of constraints applies to different classes. Thus previously mentioned VPN traffic engineering problem can be decomposed into a set of class–based VPN

traffic engineering problems, one for each service class. Our focus will be on what we call VPN real–time traffic engineering problem, where the goal is to route the real–time demands. We assume that within an NSP network several service classes are supported. One such service class is defined for all real–time applications. The solution to the VPN traffic engineering problem where traffic demands belong to other service classes is outside the scope of this thesis.

We have mentioned that failure of network elements is one of the main causes of performance degradation in VOIP traffic. The same applies in general to other real-time applications. To assure good performance, it is not sufficient to only resolve congestion problems. A sound failure recovery procedure must also be in place. MPLS-TE has a set of procedures that provide the protection for the traffic carried on the LSPs. There are two major recovery models: restoration and protection. Restoration performs rerouting of the paths after the occurrence of the fault. This approach although simpler and more cost-effective is inherently slower than protection switching and could cause unacceptable losses and delay for real-time traffic. Protection pre-establishes the paths or path segments based on certain criteria. When a fault is detected, the protected traffic is switched over to the backup path. There are two protection mechanisms, path protection and link/node protection. Link/node protection (local protection) is faster but is less scalable. It is effective when there are components that are failing much more often than others. The intent of path protection is to protect against any link/node failure on the primary (working)

path. This is achievable if the primary and backup paths are chosen as to minimize the probability of their simultaneous failure. With MPLS-based recovery, it is possible to offer different levels of protection for different classes of service, based on the service requirements. A suitable solution for real–time applications could be a combination of link/node protection and pre–established path protection. In that case, there is a need to solve the traffic engineering problem for the protection paths. We are going to assume that such a solution is used by the NSP, that is, fault resilience is achieved by using a path protection mechanism, which extends the VPN traffic engineering problem we define for real–time applications. Our goal will be not only to find a route for each of the real–time VPN demands, but also to find a route for their protection/backup path.

Therefore the problem we will address in this thesis is the VPN real–time traffic engineering problem where a path protection mechanism is used to provide a reliable solution. In the reminder of the thesis we will refer to it as the VPN RT–TE problem. Our goal is to find primary and protection/backup routes for VPN demands generated by real–time applications.

1.2 Contributions

The VPN RT–TE problem defined in Section 1.1 translates into the following set of requirements:

- Delay on the primary routes is within given bounds.

- Bandwidth requirements both on the primary and backup routes are met.
- The proportion of traffic allocated on a link by primary/backup routes (link utilization) is within given limits.
- The probability of simultaneous failure of primary/backup paths is minimized.
- Relative delay defined as a delay difference on a primary and its backup path is within given bounds.
- The solution (the size of the algorithm) should not depend on the number of VPNs.

The main contributions of this thesis are:

- A mathematical formulation of the problem taking into consideration the overall set of requirements listed above: the integrated model. The resulting formulation corresponds to a mixed integer quadratic problem (MIQP). It is known that MIQP problems are NP-hard [21].
- A scalable two step heuristic solution to the above optimization problem. By decomposing the problem in two phases we were able to reduce it to two LP problems that can be optimally solved. The proposed heuristic allows two tradeoffs. The first tradeoff is between the maximum relative delay (service quality experienced by the traffic in the case of failure) and the overall reliability of the solution. The second tradeoff is between the link utilization (service

quality experienced by the traffic) and the number of O–LSPs (management complexity).

- A *jointness* metric assigned to a pair of paths that share the same ingress/egress nodes. This metric is intended to be a measure of the probability of simultaneous failure of the paths.

- The input to the VPN RT–TE problem consists of an IP topology and a traffic demand matrix. NSPs consider this information proprietary, which makes access to real world data for simulation scenarios difficult. In the absence of real world data, we created a simulation model that approximates a real world scenario. We based our model on the NSF network. In our scenarios we used traffic demand matrices provided in [16].

1.3 Thesis organization

In this chapter, we present the context and motivation, and define the VPN RT–TE problem we study in this thesis. In Chapter 2 we give some more information related to the context of the problem and an overview of the related work. We discuss in details the problem requirements defined in Section 1.2, and define the input data of the problem in Chapter 3. Chapter 4 contains mathematical formulations of the integrated model as well as of the two step heuristic. In Chapter 5 we present our simulation model. We show and discuss the results of the experiments. We end the

thesis with a conclusion and future work section in Chapter 6.

Chapter 2

Background and Related Work

2.1 Background

In the past years network–based VPNs have gained important market interest. The reason is in their scalability and manageability. In network–based VPNs customers do not need to implement VPN specific functions. Customer sites are connected through CE routers to PE routers. The PE routers need to maintain separate VPN routing and forwarding instances (VFRs) for every supported VPN. Global IP routing tables are used in PE routers to forward IP packets that do not belong to a VPN. VFR tables are used to forward information within the VPN. A PE can have multiple VFR instances, each of which can be perceived as a virtual router (VR). A VR per supported VPN is deployed in every PE router. Interfaces on PE routers are associated with individual VFRs. The information learned through these interfaces forms what is known as a routing context for a given VFR. In MPLS–based VPNs, the Border Gateway Protocol (BGP) is used to disseminate the VPN specific routing

information learned by a PE to other PEs. Extended BGP attributes needed for this functionality are presented in RFC 2283 [11]. Based on the information stored in VRs, packets are forwarded to their destination using MPLS. A PE router binds a label to each customer prefix learned from a CE router. The label is included in the network reachability information for the prefix and advertised to other PE via BGP. When a PE router receives a packet from a CE router, it labels the packet with the label that corresponds to the destination prefix, it learned from the destination PE router. This is the inner MPLS label. It then labels the packet with the outer label used to route the packet to the destination PE router. Backbone routers forward the packet based on the outer label. When the packet arrives to the destination PE router, it pops the outer label and uses inner label. It maps the inner label to a prefix based on the information in the VFR table and forwards the packet to the destination CE router [9].

MPLS tunnels that connect PE routers can be routed using the information provided by IP routing protocols. Destination based routing often produces unbalanced traffic distribution. A solution to that is explicit routing, where routes are computed based on the performance objectives. MPLS provides the essential capabilities needed for traffic engineering with explicit routes [1].

2.2 Related Work

Traffic engineering problems for MPLS networks have been extensively studied. MPLS–TE provides a mechanism for explicit routing of packets. Explicit routing is a capable solution for improving network utilization. Traffic engineering mechanisms are also used for resource provisioning needed to accommodate QoS requirements. In [1] practical issues of traffic engineering and a working solution for traffic engineering with MPLS is given. The focus of [1] is on the mechanism by which MPLS provides traffic engineering and constraint based routing in an Internet Service Provider (ISP) network. In [2] a classification and a formulation of basic traffic engineering problems that arise in MPLS networks is given. The first problem formulated is the constraint based routing or network resource optimization problem which deals with optimal placement of LSPs in a network. The second is the connection admission problem which helps deciding if an LSP is to be admitted by the network. The third problem is rerouting of LSPs in a case of failure. And finally the network design and capacity planning problem is presented. All the problem formulations are link based. In [3] the problem of routing of LSPs is formulated with the objective to minimize the maximum link utilization. Link based IP and MIP formulations are presented and a set of four heuristics for the IP formulation. In [6] the routing problem is expanded to include multiple objectives. The first objective is to minimize the maximum link utilization (congestion), and the second is to minimize the overall cost of the solution. A two – step heuristic is proposed. In the first step a routing problem is formulated using

the first objective. There can be multiple solutions to step one. In the second step, an LSP routing problem with the cost minimization objective is solved. The value for the maximum link utilization obtained in the first step is used as a parameter. All the formulations are link based. Constraints such as maximum number of hops in a path and link/node inclusion/exclusion are incorporated in the model. In [5] a similar multiobjective problem with the focus on link utilization and resource usage is considered in a VPN DiffServ/MPLS context. A path based formulation is given. The problem is formulated as a two step MIP problem. A two step LP heuristic is proposed as a solution. The VPN–TE problem, we study in this thesis, is also a multiobjective optimization problem. However, the optimization objectives are different. Our focus is on the minimization of the propagation delay on the primary and backup paths, and on the minimization of the amount of network resources shared between primary and backup tunnels. In the heuristic solution to the problem we take the two step approach similar to [5], and [6], and use a path based formulation. In order to find maximally disjoint primary/backup tunnels we introduce jointness metric. The jointness metric associates a "failure cost" to a pair of paths. The lower the cost, the lower is the sharing of the resources between them. The notion of the "failure cost" of a pair of paths has been introduced in [7]. In [7] the problem of routing primary and backup paths has been considered. However, they consider dynamic routing of demands, where for each demand a primary and backup path is calculated as it arrives. Also, in their problem, they do not consider capacity and relative delay

constraints. To calculate the "failure cost" they use the correlated overlay link failure probability model. Our jointness metric calculates the "failure cost" of two paths based on the amount of resources they share at the IP level. Papers [2], [3], [6], and [5] focus on the problem of routing of LSPs where traffic demands for multiple source/destination pairs are given a priori. The VPN RT–TE problem we study uses the same assumption.

Chapter 3

Problem Description

In Section 1.1 we have introduced the VPN RT–TE problem. In this chapter we are going to take a closer look at the problem and give a mathematical formulation.

3.1 Problem Definition

The input to the VPN RT–TE problem consists of a) an IP network made up of IP routers and IP layer logical links and b) a set of m VPN traffic demand matrices. An element in the ith VPN traffic matrix is the real–time bandwidth requirement between an ingress–egress (source–destination) pair sd, denoted by b^i_{sd}. NOTE: We will be using terms ingress–egress and source–destination interchangeably. Our goal is to find primary and backup routes for all b^i_{sd} demands while satisfying the following requirements:

Requirement 1 The solution (the size of the algorithm) should not depend on the number of VPNs.

Requirement 2 Bandwidth requirements both on the primary and backup routes are met.

Requirement 3 Delay on the primary routes is within given bounds.

Requirement 4 Relative delay defined as a delay difference on a primary and its backup path is within given bounds.

Requirement 5 The proportion of traffic allocated on a link by primary/backup routes (link utilization) is within given limits.

Requirement 6 The probability of simultaneous failure of primary/backup paths is minimized.

The problem defined above is essentially a multicommodity network flow problem. Such a problem can be formulated using a link–based formulation where the optimization variables indicate an inclusion of a link in a route, or a path–based formulation where the optimization variables make a path selection. We find the path–based formulation more convenient because it allows us to express the delay requirements, Requirement 3 and Requirement 4, within the definition of a set of potential paths rather than by introducing them as constraints in the model. The problem can be formulated in the following way. For each source–destination pair we define a set of potential paths. The algorithm needs to find primary and backup routes for the demands b^i_{sd} by placing those demands onto one of the paths from the set. In the reminder of this section we further explore the problem requirements.

3.2 Requirement 1: The Scalability of the Solution

For a given sd, one approach would be to place each demand b^i_{sd} onto a separate LSP. This approach is clearly not a scalable one as the overall number of LSPs would grow with the number of VPNs. This would be the opposite of Requirement 1. Another approach is to put all the b^i_{sd} demands onto one LSP. This is clearly a scalable solution, however it would be beneficial to allow the bifurcation – the set of the b^i_{sd} demands to be routed over multiple paths when needed (e.g., there is no sufficient capacity to place all the demands onto one LSP). This approach is presented in [5]. The number of LSPs in this case does not depend on the number of VPNs but it depends on the available link capacity. In this way instead of having one O–LSP per source – destination pair we will end up with a set of O–LSPs per source destination pair. The idea is that individual demands placed in I–LSPs would be distributed among the available O–LSPs. The individual demands are not bifurcated as can be seen in Figure 3.1. The problem of distribution of I–LSPs among the set of O–LSPs can be formulated as a version of a Knapsack problem [5]. Our focus is on the problem of routing O–LSPs. The distribution of I–LSPs is outside of the scope of this thesis.

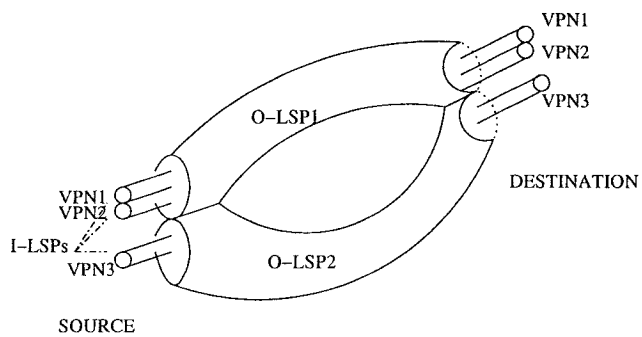

Figure 3.1: Traffic bifurcation

3.3 Requirement 2: Path Protection

When it comes to path protection there are different bandwidth allocation schemes. Here we assume that primary paths are utilized to route the demands. When a failure on a primary path occurs the traffic is rerouted on its backup path. In the mean time resources allocated on the backup path are available to preemptible low priority traffic. The same bandwidth is reserved on both primary and backup paths. This scheme is referred to as the full backup path bandwidth allocation. It is possible to allow the resources allocated on the backup path to be shared with other backup paths if the likelihood of the simultaneous failure of their corresponding primary paths is low, referred to as the shared backup path bandwidth allocation [7]. In this thesis we consider the full backup path bandwidth allocation scheme. It is a more costly solution but simpler to model and implement. We intend to explore the solution to the VPN RT–TE problem assuming the shared backup path bandwidth allocation in the future. In our multipath routing approach an individual demand is

placed into an I–LSP and then assigned to one of O-LSPs. We first define O–LSPs that act as tunnels and later we place individual I–LSPs into these tunnels. In order to protect the demands placed in the primary O–LSP tunnels we need another set of tunnels (O–LSPs) to act as a protection set. Effectively we need to define two sets of O–LSPs, one set – primary set, to be used to route the demands and another set to be used as a backup set onto which the demands would be rerouted in the case of failure. In the context of the full backup allocation scheme, the bandwidth that needs to be reserved for the backup set equals the bandwidth reserved for the primary set. It is important to notice that we do not make any constraints in terms of the number of O–LSPs within the primary and backup sets. This number does not need to be equal. The only requirement we have is that the total bandwidth reserved for the primary O–LSPs defined as $\sum_i b^i_{sd}$ equals the bandwidth reserved for the backup O–LSPs. Figure 3.2 shows a primary set that consists of one O–LSP and a backup set that has two O–LSPs. In general, a solution can have p primary O–LSPs and q backup O–LSPs.

3.4 Requirement 3: Delay Minimization

In Section 1.1 we discussed the parameters that affect the quality of voice communication in the Internet, namely loss and delay. That discussion applies in general to other real–time applications. The fixed part of the delay is determined by the propagation delay on the path. The variable part as well as a portion of the total loss

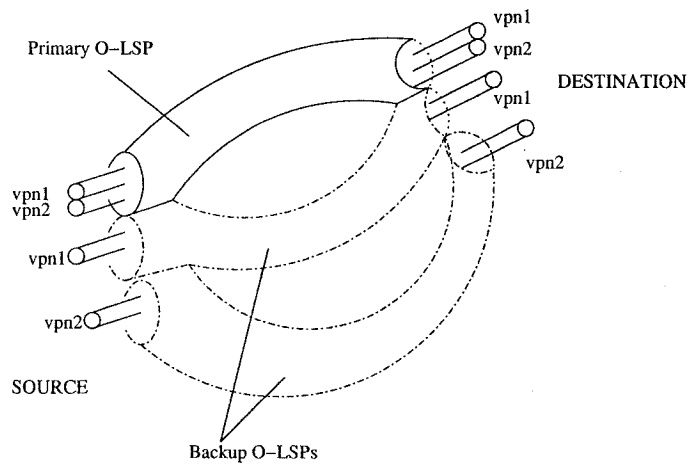

Figure 3.2: Primary and backup tunnel sets.

is a function of the queuing delay caused by the congestion in network elements. The queuing delay and loss requirements are difficult to incorporate into an optimization model. A common approach is to convert them into the equivalent bandwidth requirement . The assumption is that if the equivalent bandwidth is guaranteed on the path, the queuing delay and loss in network elements will be minimal, thus the only parameter of interest becomes the propagation delay which depends on the choice of the path (transmission delay in backbone networks is small in comparison to the propagation delay) [8]. MPLS–TE offers mechanisms for bandwidth reservation. Our assumption is that the demands given in a traffic matrix are expressed in terms of the equivalent bandwidth. In this case, in our model, when routing both primary and backup tunnels, we need to limit the propagation delay on the paths. We achieve this by defining a set of the potential paths for a source–destination pair so that all the paths in the set have their propagation delay less or equal to a parameter delay_max.

A typical value for delay_max in a backbone network is between 50ms and 80ms [15].

3.5 Requirement 4: Relative Delay

In Requirement 4 we define relative delay as a difference in propagation delay between primary and backup paths used to route a demand. In our model we aim to limit the allowable relative delay in order to prevent a major change in the propagation delay the traffic will experience in case of failure. In the case of a VoIP application, a 50ms relative delay would cause a significant performance degradation [15]. In our model, we introduce a parameter λ as the maximum allowable delay difference between a primary O–LSP and a backup O–LSP defined as a percentage of the delay on the primary O–LSP.

3.6 Requirement 5: Link Utilization

Our backbone network supports multiple classes of traffic. This is achieved by employing the MPLS DS–TE mechanisms. To assure class based differentiation of traffic an NSP needs to configure queue sizes and scheduling policies to accommodate different SLAs. It is impractical to base the configuration on a current link load, instead the relative proportion of each traffic type is fixed a priori and queue sizes and scheduling policies are set accordingly. We express this limit in the Requirement 5. We introduce a parameter μ that corresponds to the portion of the link capacity

available to primary/backup O–LSPs.

3.7 Requirement 6: Failure Probability Factor

In Requirement 6 we express the following goal: find two sets of O–LSPs, a primary and a secondary set, between an ingress/egress pair of nodes so that the probability of their simultaneous failure is minimized. To accomplish the requirement we introduce the failure probability factor $f_{pp'}$. For a primary O–LSP p value of $f_{pp'}$ defines a "cost" of choosing p' to be a member of the secondary set. Note that the "cost" is not the probability of their simultaneous failure, it is a metric that reflects that probability: the higher the probability, the higher is the cost.

There are different ways to define $f_{pp'}$ values. The approach depends on the assumptions related to the network we intend to model. If the network uses IP-level restoration, with no protection at lower layers, then a failure of any component at or below the IP layer will manifest itself as an IP link failure. Let's assume that the IP network is built on top of a Dense Wavelength Division Multiplexing (DWDM) infrastructure. Than, at the IP level it is not known if links share a fiber or another piece of optical equipment. Links disjoint at the IP level may not be disjoint at the optical level. Finding maximally disjoint paths (node/link disjoint) at the IP level may not provide the desired level of availability if the paths share links at the optical level. Also some components are failing more frequently then others. Understanding the cause of link failures and their statistical properties can be used to build a failure

model for a given IP topology that would show more accurately the likelihood of two paths failing at the same time, expressed as an $f_{pp'}$ value.

In our model we assume the existence of a lower layer protection mechanism, and focus on router and IP link failures. In order to maximize the reliability of the solution we need to minimize the resources (nodes/links) shared by the primary and backup set. We define $f_{pp'}$ to be a measure of link-node jointness of the two paths. Let's assume that a path p has $lmax_p$ links. We define $f_{pp'}$ such that

$$f_{pp'} = J(sn, sl)$$

where J is a *jointness metric*, sn represents the number of nodes (excluding v_s and v_d) a path p shares with p' $(0 \leq sn \leq lmax_p - 1)$ and, sl represents the number of links the path p shares with p', $(0 \leq sl \leq lmax_p)$. Clearly, the more link/node joint paths there are (more resources they share), the higher is the probability of them failing at the same time. In Section 4.6 we define a concrete *jointness metric* and provide the reasoning behind it.

Chapter 4

Mathematical Formulation

4.1 The Integrated Model and the Two–step Heuristic

In Chapter 3, we have introduced the parameters that are the input to the VPN RT–TE problem and discussed in detail the requirements. Our next step is the mathematical formulation of the requirements. As already stated in Section 1.2 when all the requirements are taken into consideration we end up with a mixed integer quadratic problem. It is possible to linearize a quadratic problem by introducing additional variables and constraints, and transform it into a mixed integer linear problem. While such a mixed integer problem can be solved optimally for small networks, the number of variables becomes too large when it is applied to real backbone networks. We call this one step formulation of the problem – the integrated model. Figure 4.1 depicts its input and output.

When it comes to routing primary and backup LSPs, there are slightly different requirements. The primary LSPs are intended to route the traffic most of the

$$\begin{array}{c} IP\ topology \\ VPN\ demand\ matrices \\ delay\_max,\ \mu,\ \lambda \end{array}$$

$$\Downarrow$$

$$\begin{array}{ll} PRIMARY & O-LSPs \\ BACKUP & O-LSPs \end{array}$$

Figure 4.1: Integrated model.

time, backup LSPs are used only in a case of failure or maintenance operations. The main requirement when routing primary LSPs is performance optimization and minimization of the cost of the solution. For the backup LSPs the main requirement is to maximize the reliability of the solution without compromising the performance. In terms of the cost it is more acceptable to use a more expensive solution since the bandwidth on the backup paths is used only during the recovery period. Although decoupling of the problem of routing primary and backup tunnels does not lead to an optimal solution of the integrated problem, it makes sense in the light of the different set of requirements. This is the approach we take in an attempt to provide a practical solution. In the first step we solve the problem of routing of the primary tunnels. The output of the first step of our solution, we use as an input to the problem of finding backup tunnels. By decomposing the original problem in two steps we reduce it to two LP problems that can be optimally solved. The LP problems can be

$IP\ topology$
$VPN\ demand\ matrices$
$delay\_max, \mu$

$$\Downarrow$$

$$PRIMARY \quad O-LSPs$$

Figure 4.2: Step One.

$IP\ topology$
$VPN\ demand\ matrices$
$Primary\ O-LSPs$
$delay\_max, \mu, \lambda$

$$\Downarrow$$

$$BACKUP \quad O-LSPs$$

Figure 4.3: Step Two.

solved by CPLEX software [20] for large number of variables. Figures 4.2 and 4.3 show the input and output parameters to the step one and step two of the solution respectively. In the following sections we provide mathematical formulations for the

integrated model as well as for the two-step heuristic.

4.2 Notation

Let us represent the network by a graph $G = (V, L)$ where V is a set of n nodes (IP routers) and L the set of IP logical links between the nodes. Each link $\ell \in L$ has a bandwidth capacity u_ℓ.

- Traffic

  T = $n \times n$ traffic matrix such that $T_{sd} = \sum_i b^i_{sd}$ = total requested bandwidth from v_s to v_d over all VPNs where m is the number of VPNs and v_s and v_d a source/destination pair of nodes.

  $\mathcal{SD}$ = $\{(v_s, v_d) : T_{sd} > 0\}$, set of source and destination pairs with some traffic.

- Network topology

  p = a routing path,

  $\mathcal{P}_{sd}$ = set of potential routing paths from v_s (source) to v_d (destination),

  $\mathcal{P} = \bigcup_{sd \in \mathcal{SD}} \mathcal{P}_{sd}$ = overall set of potential paths,

  $\mathsf{delay}(\ell)$ = propagation delay on a link ℓ.

4.3 Mathematical Formulation: An Integrated Model

4.3.1 Variables

- Traffic flows

x_p^1 = fraction of bandwidth flow of T_{sd} (assuming $p \in \mathcal{P}_{sd}$) that goes over p, with p being a primary path

$$0 \leq x_p^1 \leq 1 \qquad p \in \mathcal{P}_{sd}, (v_s, v_d) \in \mathcal{SD}.$$

x_p^2 = fraction of bandwidth flow of T_{sd} (assuming $p \in \mathcal{P}_{sd}$) that goes over p, with p being a backup path

$$0 \leq x_p^2 \leq 1 \qquad p \in \mathcal{P}_{sd}, (v_s, v_d) \in \mathcal{SD}.$$

$$y_p^1 = \begin{cases} 1 & \text{if } p \text{ is used as a primary path, and hence } x_p^1 > 0, \\ 0 & \text{if } p \text{ is not used as a primary path, and hence } x_p^1 = 0. \end{cases}$$

$$y_p^1 \in \{0, 1\} \qquad p \in \mathcal{P}_{sd}, (v_s, v_d) \in \mathcal{SD}.$$

Note that a given path p can be unused, i.e., be neither a primary path nor a backup path.

$$y_p^2 = \begin{cases} 1 & \text{if } p \text{ is used as a backup path, and hence } x_p^2 > 0, \\ 0 & \text{if } p \text{ is not used as a backup path, and hence } x_p^2 = 0. \end{cases}$$

$$y_p^2 \in \{0, 1\} \qquad p \in \mathcal{P}_{sd}, (v_s, v_d) \in \mathcal{SD}.$$

4.3.2 Parameters

- $\delta_\ell^p = \begin{cases} 1 & \text{link } \ell \text{ is part of path } p \\ 0 & \text{otherwise} \end{cases}$
- $f_{pp'}$ = failure probability factor that reflects the probability of simultaneous failure of a primary path p and a backup path p'.
- λ = delay difference between a primary and a backup path as a percentage of the primary path delay.
- μ = a portion of the link bandwidth u_ℓ available to real-time traffic.
- M an arbitrary large constant.

4.3.3 Constraints

- Delay constraint on the paths; we will limit the set of potential paths such that

$$\mathsf{delay}(p) \leq \mathsf{delay\_max} \qquad p \in \mathcal{P}$$

where

$$\mathsf{delay}(p) = \sum_{\ell \in p} \mathsf{delay}(\ell) \qquad p \in \mathcal{P}.$$

In other words, $\mathcal{P}_{sd} = \{p \in \mathcal{P} : p$ is a path from v_s to v_d such that $\mathsf{delay}(p) \leq \mathsf{delay\_max}\}$.

- Satisfy all requests, i.e., sum of all bandwidth fractions is equal to 1

$$\sum_{p \in \mathcal{P}_{sd}} x_p^1 = 1 \qquad (v_s, v_d) \in \mathcal{SD}.$$

- For each pair of source and destination nodes, overall bandwidth on the primary paths is equal to overall bandwidth on the backup paths

$$\sum_{p \in \mathcal{P}_{sd}} x_p^1 = \sum_{p \in \mathcal{P}_{sd}} x_p^2 \qquad (v_s, v_d) \in \mathcal{SD}.$$

- Bandwidth allocated to primary and backup paths on each link

$$\sum_{(v_s,v_d) \in \mathcal{SD}} T_{sd} \sum_{p \in \mathcal{P}_{sd}} \delta_\ell^p x_p^1 + \sum_{(v_s,v_d) \in \mathcal{SD}} T_{sd} \sum_{p \in \mathcal{P}_{sd}} \delta_\ell^p x_p^2 \leq \mu u_\ell \qquad \ell \in L.$$

- A given path can be either unused, or used as a primary path, or used as a backup path, but it cannot be used both as a primary and a backup path for

the same connection.

$$y_p^1 + y_p^2 \leq 1 \qquad p \in \mathcal{P} \tag{4.1}$$

$$x_p^1 \leq y_p^1 \qquad p \in \mathcal{P} \tag{4.2}$$

$$x_p^2 \leq y_p^2 \qquad p \in \mathcal{P} \tag{4.3}$$

$$y_p^1 \leq M x_p^1 \qquad p \in \mathcal{P} \tag{4.4}$$

$$y_p^2 \leq M x_p^2 \qquad p \in \mathcal{P}. \tag{4.5}$$

Inequality (4.1): a path p cannot be used both as a primary and a backup path for the same request.

Inequality (4.2): if p is not used as a primary path, then it carries no traffic with respect to primary.

Inequality (4.3): if p is not used as a backup path, then it carries no traffic with respect to backup.

Inequality (4.4): if p is used as a primary, it must carry nonzero traffic for primary; but if it carries no traffic, it cannot be considered as a primary path.

Inequality (4.5): if p is used as a backup path, it must carry nonzero traffic for backup; but if it carries no traffic, it cannot be considered as a backup path.

- Limit on the delay differences between the primary and the backup paths

$$|\mathsf{delay}(p) - \mathsf{delay}(p')| \leq \lambda\, \mathsf{delay}(p) \qquad p \in \mathcal{P}_{sd}, p' \in \mathcal{P}_{sd}, (v_s, v_d) \in \mathcal{SD}$$

that is equivalent to

$$-\lambda\,\mathsf{delay}(p) \leq \mathsf{delay}(p) - \mathsf{delay}(p') \leq \lambda\,\mathsf{delay}(p)$$

$$p \in \mathcal{P}_{sd}, p' \in \mathcal{P}_{sd}, (v_s, v_d) \in \mathcal{SD}$$

which can be rewritten

$$-(\lambda+1)\mathsf{delay}(p)\,y_p^1 + \mathsf{delay}(p')\,y_{p'}^2 \leq M(1-y_p^1)$$

$$p \in \mathcal{P}_{sd}, p' \in \mathcal{P}_{sd}, (v_s, v_d) \in \mathcal{SD} \quad (4.6)$$

$$(1-\lambda)\mathsf{delay}(p)\,y_p^1 - \mathsf{delay}(p')\,y_{p'}^2 \leq M(1-y_{p'}^2)$$

$$p \in \mathcal{P}_{sd}, p' \in \mathcal{P}_{sd}, (v_s, v_d) \in \mathcal{SD} \quad (4.7)$$

where

$$\mathsf{delay}(p) = \sum_{\ell \in p} \mathsf{delay}(\ell) \qquad p \in \mathcal{P}.$$

We list in the Table 4.1 all the four cases that can arise with inequalities (4.6) and (4.7). We can easily deduce from this list that those two equations are valid, either they are redundant or imposing the delay constraints when both paths p and p' are used, i.e., $p \in \mathcal{P}_{sd}, p' \in \mathcal{P}_{sd}$.

y_p^1	$y_{p'}^2$	Enequality (4.6)	Inequality (4.7)
0	0	$0 \leq M$	$0 \leq M$
0	1	$\mathsf{delay}(p') \leq M$	$-\mathsf{delay}(p') \leq 0$
1	0	$-(\lambda+1)\,\mathsf{delay}(p) \leq 0$	$(1-\lambda)\,\mathsf{delay}(p) \leq M$
1	1	$-(\lambda+1)\,\mathsf{delay}(p) + \mathsf{delay}(p') \leq 0$	$(1-\lambda)\,\mathsf{delay}(p) + \mathsf{delay}(p') \leq 0$

Table 4.1: Validation of delay inequalities

4.3.4 Objective function

- First component: minimize the delay on the primary paths

$$f_1^{\text{OBJ}} = \sum_{(v_s,v_d)\in\mathcal{SD}} \sum_{p\in\mathcal{P}_{sd}} \mathsf{delay}(p)\, x_p^1$$

- Second component: minimize the failure

$$f_2^{\text{OBJ}} = \sum_{(v_s,v_d)\in\mathcal{SD}} \sum_{p'\in\mathcal{P}_{sd}} \left(x_{p'}^2 \sum_{p\in\mathcal{P}_{sd}} f_{pp'} y_p^1 \right).$$

This last objective function is quadratic and can be linearized with the introduction of continuous variables $z_{pp'} = x_{p'}^2 y_p^1$ such that

$$0 \leq z_{pp'} \leq x_{p'}^2 (\leq 1) \qquad p,p' \in \mathcal{P}_{sd}, (v_s,v_d) \in \mathcal{SD} \tag{4.8}$$

$$z_{pp'} \leq x_{p'}^2 \qquad p,p' \in \mathcal{P}_{sd}, (v_s,v_d) \in \mathcal{SD} \tag{4.9}$$

$$z_{pp'} \leq y_p^1 \qquad p,p' \in \mathcal{P}_{sd}, (v_s,v_d) \in \mathcal{SD} \tag{4.10}$$

$$x_{p'}^2 + y_p^1 - 1 \leq z_{pp'} \qquad p,p' \in \mathcal{P}_{sd}, (v_s,v_d) \in \mathcal{SD} \tag{4.11}$$

One can easily check that the above linearization is valid considering all possible

$x^2_{p'}$	y^1_p	$z_{pp'}$	Other constraints
0	0	0 due to constraint (4.9)	(4.8), (4.10) and (4.11) are satisfied or redundant
0	1	0 due to constraint (4.9)	(4.8), (4.10) and (4.11) are satisfied or redundant
> 0	0	0 due to constraint (4.10)	(4.8), (4.9) and (4.11) are satisfied or redundant
> 0	1	$x^2_{p'}$ due to constraint (4.11)	(4.8), (4.9) and (4.10) are satisfied or redundant

Table 4.2: Validation of the linearization.

cases. They are enumerated in the Table 4.2.

4.4 Mathematical Formulation: A Two-step Model

In this section we present mathematical formulations for the two–step heuristic. We give an MILP, and LP formulations for the step two of the heuristic.

4.4.1 Step One: Definition of the Primary Paths (S1)

4.4.1.1 Variables

- Traffic flows

x^1_p = fraction of bandwidth flow of T_{sd} (assuming $p \in \mathcal{P}^1_{sd}$) that goes over p, with p being a primary path

$$0 \leq x^1_p \leq 1 \qquad p \in \mathcal{P}^1_{sd}, (v_s, v_d) \in \mathcal{SD}.$$

4.4.1.2 Parameters

- $\delta_\ell^p = \begin{cases} 1 & \text{link } \ell \text{ is part of path } p \\ 0 & \text{otherwise} \end{cases}$
- μ = a portion of the link bandwidth u_ℓ available to real-time traffic

4.4.1.3 Constraints

- Delay constraint on the paths; we will limit the set of potential paths such that

$$\mathsf{delay}(p) \leq \mathsf{delay\_max} \qquad p \in \mathcal{P}$$

where

$$\mathsf{delay}(p) = \sum_{\ell \in p} \mathsf{delay}(\ell) \qquad p \in \mathcal{P}.$$

In other words, $\mathcal{P}_{sd}^1 = \{p \in \mathcal{P} : p \text{ is a path from } v_s \text{ to } v_d \text{ such that } \mathsf{delay}(p) \leq \mathsf{delay\_max}\}$.

- Satisfy all requests, i.e., sum of all bandwidth fractions is equal to 1

$$\sum_{p \in \mathcal{P}_{sd}^1} x_p^1 = 1 \qquad (v_s, v_d) \in \mathcal{SD}.$$

- Bandwidth capacity on each logical link for the primary paths

$$\sum_{(v_s, v_d) \in \mathcal{SD}} T_{sd} \sum_{p \in \mathcal{P}_{sd}^1} \delta_\ell^p x_p^1 \leq \mu\, u_\ell \qquad \ell \in L.$$

4.4.1.4 Objective

Minimize the delay on the primary paths

$$f_1^{\text{OBJ}} = \sum_{(v_s,v_d)\in\mathcal{SD}} \sum_{p\in\mathcal{P}_{sd}^1} \mathsf{delay}(p)\, x_p^1$$

4.4.2 Step Two: Definition of the Backup Paths – MILP Formulation

It is possible to formulate the step two of the problem by using the same potential set of paths $\mathcal{P}_{sd}^1$ to define the problem variables as in step one. In that case the relative delay requirement described in Section 3.5 can be introduced through a set of constraints. That approach forces us to include discrete variables in our formulation which leads us to a mixed integer linear programming problem (MILP). In this section we give the MILP formulation. Another approach is to define problem variables using a potential set of paths that includes all the paths from $\mathcal{P}_{sd}^1$ with the relative delay within the given bounds. This leads us to a linear programming problem formulation which we present in Section 4.4.3.

4.4.2.1 Variables

- x_p^2 = fraction of bandwidth flow of T_{sd} (assuming $p \in \mathcal{P}_{sd}^1$) that goes over p, with p being a backup path

$$0 \leq x_p^2 \leq 1 \qquad p \in \mathcal{P}_{sd}^1, (v_s, v_d) \in \mathcal{SD}.$$

- $y_p^2 \in \{0, 1\}$, $x_p^2 \leq M y_p^2$ for all $p \in \mathcal{P}_{sd}^1, (v_s, v_d) \in \mathcal{SD}$.

4.4.2.2 Parameters

- Primary traffic flows

  x_p^1 = fraction of bandwidth flow of T_{sd} (assuming $p \in \mathcal{P}_{sd}^1$) that goes over p, with p being a primary path as determined in the first phase.

- $y_p^1 \in \{0, 1\}$, $x_p^1 \leq M y_p^1$ for all $p \in \mathcal{P}_{sd}^1, (v_s, v_d) \in \mathcal{SD}$.

- $f_{p'}$ = failure probability factor that reflects the probability of simultaneous failure of a backup path p' and the corresponding primary set.

4.4.2.3 Constraints

- For each pair of source and destination nodes, overall bandwidth on the primary paths = overall bandwidth on the backup paths

$$\sum_{p \in \mathcal{P}_{sd}^1} x_p^1 = \sum_{p \in \mathcal{P}_{sd}^1} x_p^2 \qquad (v_s, v_d) \in \mathcal{SD}.$$

- Bandwidth capacity on each logical link for the backup paths

$$\sum_{(v_s,v_d)\in\mathcal{SD}} T_{sd} \sum_{p\in\mathcal{P}^1_{sd}} \delta^p_\ell x^2_p \leq u^R_\ell = \mu u_\ell - \sum_{(v_s,v_d)\in\mathcal{SD}} T_{sd} \sum_{p\in\mathcal{P}^1_{sd}} \delta^p_\ell x^1_p \qquad \ell \in L$$

Note that $\mu u_\ell - \sum_{(v_s,v_d)\in\mathcal{SD}} T_{sd} \sum_{p\in\mathcal{P}^1_{sd}} \delta^p_\ell x^1_p$ corresponds to the residual bandwidth for the backup paths, once the primary paths have been defined.

- Limit on the delay differences between the primary and the backup paths

$$|\mathsf{delay}(p) - \mathsf{delay}(p')| \leq \lambda\, \mathsf{delay}(p) \qquad p, p' \in \mathcal{P}^1_{sd}, (v_s, v_d) \in \mathcal{SD}$$

that is equivalent to

$$-\lambda\, \mathsf{delay}(p) \leq \mathsf{delay}(p) - \mathsf{delay}(p') \leq \lambda\, \mathsf{delay}(p) \qquad p, p' \in \mathcal{P}^1_{sd}, (v_s, v_d) \in \mathcal{SD}$$

which can be rewritten

$$-(\lambda+1)\mathsf{delay}(p)\, y^1_p + \mathsf{delay}(p')\, y^2_{p'} \leq M(1 - y^1_p)$$

$$p, p' \in \mathcal{P}^1_{sd}, (v_s, v_d) \in \mathcal{SD} \quad (4.12)$$

$$(1-\lambda)\mathsf{delay}(p)\, y^1_p - \mathsf{delay}(p')\, y^2_{p'} \leq M(1 - y^2_{p'})$$

$$p, p' \in \mathcal{P}^1_{sd}, (v_s, v_d) \in \mathcal{SD} \quad (4.13)$$

where

$$\mathsf{delay}(p) = \sum_{\ell \in p} \mathsf{delay}(\ell) \qquad p \in \mathcal{P}$$

4.4.2.4 Objective

$$f_2^{\text{OBJ}} = \min \sum_{(v_s,v_d)\in\mathcal{SD}} \sum_{p'\in\mathcal{P}^1_{sd}} f_{p'} x^2_{p'}$$

4.4.3 Step Two: Definition of the Backup Paths – LP Formulation (S2)

Here we define problem variables using a potential set of paths that includes all the paths from $\mathcal{P}^1_{sd}$ with the relative delay within the given bounds. This effectively removes a need for discrete variables $y^1_{p'}$ and $y^2_{p'}$, and constraints (4.12) and (4.13). It becomes possible to formulate the problem using only continuous variables which leads to a linear programming problem.

4.4.3.1 Variables

- x^2_p = fraction of bandwidth flow of T_{sd} (assuming $p \in \mathcal{P}^2_{sd}$) that goes over p, with p being a backup path

$$0 \le x^2_p \le 1 \qquad p \in \mathcal{P}^2_{sd}, (v_s, v_d) \in \mathcal{SD}.$$

4.4.3.2 Parameters

- Primary traffic flows

  x_p^1 = fraction of bandwidth flow of T_{sd} (assuming $p \in \mathcal{P}_{sd}^1$) that goes over p, with p being a primary path as determined in the first phase.

- $y_p^1 \in \{0, 1\}$, $x_p^1 \leq M y_p^1$ for all $p \in \mathcal{P}_{sd}^1, (v_s, v_d) \in \mathcal{SD}$.

- $\overline{d}_{sd} = \sum_{p \in \mathcal{P}_{sd}^1} y_p^1 \mathsf{delay}(p) \,/\, \sum_{p \in \mathcal{P}_{sd}^1} y_p^1$.

- $f_{p'}$ = failure probability factor that reflects the probability of simultaneous failure of a backup path p' and the corresponding primary set of paths.

4.4.3.3 Constraints

- Delay constraint on the backup paths; we will limit the set of potential paths for the backup paths such that

$$|\mathsf{delay}(p) - \overline{d}_{sd}| \leq \lambda\, \overline{d}_{sd} \qquad p \in \mathcal{P}_{sd}^1.$$

  where

$$\mathsf{delay}(p) = \sum_{\ell \in p} \mathsf{delay}(\ell) \qquad p \in \mathcal{P}.$$

  In other words, $\mathcal{P}_{sd}^2 = \{p \in \mathcal{P}_{sd}^1 : p$ is a path from v_s to v_d such that $|\mathsf{delay}(p) - \overline{d}_{sd}| \leq \lambda\, \overline{d}_{sd}\}$.

- For each pair of source and destination nodes, overall bandwidth on the primary paths = overall bandwidth on the backup paths

$$\sum_{p \in \mathcal{P}^1_{sd}} x^1_p = \sum_{p \in \mathcal{P}^2_{sd}} x^2_p \qquad (v_s, v_d) \in \mathcal{SD}$$

- Bandwidth capacity on each logical link for the backup paths

$$\sum_{(v_s,v_d) \in \mathcal{SD}} T_{sd} \sum_{p \in \mathcal{P}^2_{sd}} \delta^p_\ell x^2_p \leq u^R_\ell = \mu u_\ell - \sum_{(v_s,v_d) \in \mathcal{SD}} T_{sd} \sum_{p \in \mathcal{P}^1_{sd}} \delta^p_\ell x^1_p \qquad \ell \in L$$

Note that $\mu u_\ell - \sum_{(v_s,v_d) \in \mathcal{SD}} T_{sd} \sum_{p \in \mathcal{P}^1_{sd}} \delta^p_\ell x^1_p$ corresponds to the residual bandwidth for the backup paths, once the primary paths have been defined.

4.4.3.4 Objective

$$f_2^{\text{OBJ}} = \min \sum_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$$

4.5 Problem Complexity

We have presented so far three formulations of the problem: an integrated model, a two step heuristic with an MILP formulation of the second set, and a two step heuristic with an LP formulation of the second step. For the step one of the heuristic we will use the acronym S1 and for the LP formulation of the step two S2.

The integrated model belongs to the MIQP class of problems that is known

to be NP-hard [21]. It involves continuous variables x_p^1 and x_p^2, and discrete variables y_p^1 and y_p^2. The total number of variables is $4 \times |\bigcup_{sd \in \mathcal{SD}} \mathcal{P}_{sd}|$. We provided an MILP formulation of the problem by introducing additional continuous variables $z_{pp'}$. The total number of variables for the MILP formulation is $4 \times |\bigcup_{sd \in \mathcal{SD}} \mathcal{P}_{sd}| + |\bigcup_{sd \in \mathcal{SD}} \mathcal{P}_{sd}|^2$. MILP problems can be solved by standard software such as CPLEX but the efficiency of the algorithm depends on the number of variables and constraints. Since the number of variables in our case is a polynomial function of the number of paths in the potential set, it would not be possible to solve it efficiently for a typical backbone network. This has prompted us to search for a heuristic solution.

The first step of our two step heuristic is formulated as an LP problem. It has $|\bigcup_{sd \in \mathcal{SD}} \mathcal{P}_{sd}^1|$ variables and $(|\mathcal{SD}| + |L| + 1)$ constraints. The size of the $\bigcup_{sd \in \mathcal{SD}} \mathcal{P}_{sd}^1$ set depends on the topology of the network as well as on the delay_max constraint. LP problems can be solved in polynomial time. However, in practice, the simplex algorithm although an exponential algorithm is widely used to solve most LP problems. The simplex algorithm in general performs well for hundreds thousands of variables and constraints when used to solve structured LP problems such as the multicommodity network flow problem [22]. When applied to a backbone network we expect the number of variables in the step one and the step two models to be in this range which makes the models applicable to practical scenarios, see Section 5.2.

For the second step of the solution we offered an MILP and an LP formulation. The MILP formulation has a larger number of variables: $2 \times |\bigcup_{sd \in \mathcal{SD}} \mathcal{P}_{sd}^1|$ and

constraints: $(2 \times |\mathcal{SD}| \times |\bigcup_{sd \in \mathcal{SD}} \mathcal{P}^1_{sd}| + |L| + |\mathcal{SD}| + 1)$ then the LP formulation with $|\bigcup_{sd \in \mathcal{SD}} \mathcal{P}^2_{sd}|$ variables and $(|\mathcal{SD}| + |L| + 1)$ constraints. The LP formulation is a simplification of the MILP problem, thus a preferred choice and the one we implement in our simulation model.

4.6 Jointness Metric

In Section 3.7 we define the failure probability factor $f_{pp'}$ to be a measure of link-node jointness of two paths where p is a primary and p' is a backup path. We use this definition in the integrated model. For the two-step heuristic we introduce a different one. We define the failure probability factor as a measure of jointness of a backup path p' and a primary set of paths.

Primary tunnels are the input to the second step of our model. The goal is to find backup sets maximally disjoint from the primary sets. Thus, we need to find a path p' (or multiple paths if needed) from the potential set of paths $\bigcup_{sd \in \mathcal{SD}} \mathcal{P}^2_{sd}$ so that p' shares the minimal number of resources (nodes/links) with the whole primary set. We introduce $f_{p'}$ to be a "cost" of choosing a path p' as a backup for the primary set. The higher the jointness, the higher is the cost. We achieve our goal by minimizing the overall "cost" of the solution as expressed in the objective function of the second step defined in Section 4.4.3.4. Note that it is not possible to use this approach for the integrated model, because both primary and backup tunnels are calculated in the same step. We show the objective function here:

$$f_2^{\text{OBJ}} = \min \sum_{(v_s,v_d)\in\mathcal{SD}} \sum_{p'\in\mathcal{P}_{sd}^2} f_{p'}x_{p'}^2$$

The function is a sum of non–negative factors $f_{sd} = \sum_{p'\in\mathcal{P}_{sd}^2} f_{p'}x_{p'}^2$ over all sd pairs. To minimize the sum the algorithm will minimize each of the factors. For each factor f_{sd} the values $f_{p'}$ can be sorted in the ascending order. Let's define $f_{p'}^{min}$ to be the first member of the ordered set. Then the value of the i–th member of the set can be expressed as $f_{p'}^{min} + K_i$ where K_i is a positive constant. Now f_{sd} can be written as a function of $f_{p'}^{min}$ and K_i:

$$\sum_{p'\in\mathcal{P}_{sd}^2} f_{p'}x_{p'}^2 = \sum_{p'\in\mathcal{P}_{sd}^2} f_{p'}^{min}x_{p'}^2 + \sum_{p'\in\mathcal{P}_{sd}^2} K_i x_{p'}^2$$

After taking into account the bandwidth constraint (Section 4.4.3.3):

$$\sum_{p\in\mathcal{P}_{sd}^2} x_p^2 = 1$$

the equation becomes:

$$\sum_{p'\in\mathcal{P}_{sd}^2} f_{p'}x_{p'}^2 = f_{p'}^{min} + \sum_{p'\in\mathcal{P}_{sd}^2} K_i x_{p'}^2$$

It can be easily seen that in order to minimize f_{sd} the algorithm will try to reserve the bandwidth on a path p' with the minimal value of $f_{p'}$ (the first path in the ordered set). If there is no enough bandwidth it will use the next one from the set and so on.

Figure 4.4 shows a set of primary paths p_1, p_2, p_3 and a set of potential backup paths p'_1, p'_2, p'_3, p'_4 with their corresponding failure probability factors $f_{p'_1}$, $f_{p'_2}$, $f_{p'_3}$, $f_{p'_4}$. Paths p'_1, p'_2, p'_3 are link and node disjoint from the primary set. Path p'_4 shares one link and two nodes with the primary set. This means that we need to assign "costs" so that $f_{p'_1} = f_{p'_2} = f_{p'_3} < f_{p'_4}$. This gives us an ordered set: $(f_{p'_1}, f_{p'_2}, f_{p'_3}, f_{p'_4})$. The minimal value for f_{sd} is $f_{p'_1}$. The algorithm will try to route the demand for the sd pair over the paths p'_1, p'_2, p'_3 (they have equal cost). If there is no enough bandwidth on them, it will use path p'_4.

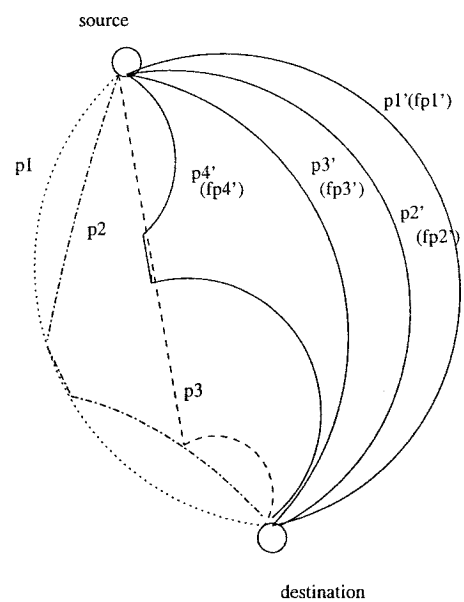

Figure 4.4: Jointness

Here we propose a concrete jointness metric $J(Sn, Sl)$. Let's assume that a path p has ℓ links. We define $f_{p'}$ such that

$$f_{p'} = J(Sn, Sl), \qquad Sn = \sum_{p \in \mathcal{P}^1_{sd}} sn_p, \qquad Sl = \sum_{p \in \mathcal{P}^1_{sd}} sl_p$$

where sn_p represents the number of nodes (excluding v_s and v_d) a path p shares with p' $(0 \leq sn_p \leq \ell - 1)$ and, sl_p represents the number of links the path p shares with p', $(0 \leq sl_p \leq \ell)$.

In our simulation model we use the following assumptions:

a) nodes represent Points of Presence (POPs),

b) links are inter-POP links,

c) multiple link failures are caused by router related failures within POPs,

d) router-related failure events are evenly distributed across different POPs,

e) single link failure events are evenly distributed over all links.

Because of the assumption c), if two IP links do not connect to the same routers then the probability of them failing simultaneously is minimal. Because of the assumption d) and e) the mapping $(Sn, Sl) \rightarrow J(Sn, Sl)$ does not depend on the choice of nodes/links, but only on their number.

Typically in an ISP network POPs contain a small number of fully meshed backbone routers and a large number of access routers. Adjacent POPs are connected by a number of parallel logical links terminating at different core routers so that a single router failure cannot cause a disconnect between the POPs [15]. For a given

pair of ingress/egress POPs the set of POP-disjoint paths is a subset of the link-disjoint paths. The number of POP-disjoint paths cannot be greater than the number of link-disjoint paths. POP-disjoint paths represent the highest level of availability. Since 70% of all failure events are single link failures [12], and parallel inter-POP links mostly connect to different routers, link-disjoint paths may provide a reasonable level of availability. We incorporate this conclusion in the design of the jointness metric.

A primary set of paths can contain a maximum of Sn_{max} nodes and Sl_{max} links. Let's define bif_{sd} to be the number of primary paths per sd pair. Then $bif_{max} = \max_{(v_s,v_d)\in\mathcal{SD}} bif_{sd}$. Let's $\overline{\ell}$ be the limit on the number of links any path can have. Then $Sl_{max} = bif_{max} * \overline{\ell}$ and $Sn_{max} = bif_{max} * (\overline{\ell} - 1)$. A path p' can share with a primary set $0 \leq Sl \leq (Sn_{max} + 1)$ links and $0 \leq Sn \leq Sn_{max}$. There are number of (Sl, Sn) combinations. The jointness metric assigns a value to each combination. The value depends only on Sl and Sn. We do not distinguish between different cases that result in the same Sl and Sn. We want the optimization algorithm to choose a backup path so that Sl is minimized. Out of two paths that have the same value of Sl the one with a smaller Sn will be chosen. This is expressed in the following inequalities

$$J(Sl, Sn1) < J(Sl + 1, Sn2) \quad \forall\, Sn1, Sn2 \in [0, Sn_{max}]$$

$$J(Sl, Sn) < J(Sl, Sn + 1) \quad \forall\, Sl \in [0, Sl_{max}], \quad Sn \in [0, Sn_{max}]$$

$$J(0, Sn) = Sn * C_1 \qquad Sn \in [0, Sn_{max}]$$
$$J(0)min = J(0,0) = 0$$
$$J(0)max = J(0, Sn_{max}) = Sn_{max} * C_1$$

$$J(Sl, Sn) = J(Sl)min + (Sn - Sl + 1) * C_1 \qquad Sl \in [1, Sn_{max}], \quad Sn \in [Sl, Sn_{max}]$$
$$J(Sl)min = \lfloor J(Sl-1)max/C_2 \rfloor * C_2 + C_2$$
$$J(Sl)max = J(Sl, Sn_{max})$$
$$J(Sl, Sn) = J(Sl)min \qquad Sl \in [Sn_{max}, Sn_{max} + 1]$$

where C_1, C_2 are positive integers and $C_1 \ll C_2$

Figure 4.5: Jointness metric

Based on these inequalities we propose a mapping $(Sn, Sl) \rightarrow J(Sn, Sl)$ that favors link disjoint paths (figure 4.5).

Chapter 5

Implementation and Results

The two-step LP heuristic described in the previous chapter has been implemented in C++, using the CPLEX-LP [19] libraries. The implementation has three modules, namely Input Preparation, Phase One and Phase Two. The Input Preparation module generates an IP topology and traffic demand matrices. Phase One implements the first, and Phase Two the second step of the heuristic. In this chapter we will discuss each module, its input and output data and the results of the heuristic.

5.1 Input Preparation

Figure 4.1 shows the input to the VPN RT–TE problem. It consists of an IP topology, an aggregate real–time traffic demand matrix, and parameters μ, λ, and delay_max. The IP topology corresponds to the topology of a backbone network operated by one NSP. The aggregate traffic demand matrix is a sum of the real–time

traffic demand matrices provided by VPN customers. NSPs consider information about the IP topology of their networks and traffic demands of their customers confidential and proprietary. Consequently, there is no real–world data readily available to the research community that we could use in our simulation scenarios. Faced with this problem, researchers often use random topologies and matrices to validate their algorithms. In the absence of real world data, we create a simulation model for the LP heuristics that more closely approximates a real world scenario. To accomplish this we use two sources: the reference transport network scenarios provided in [16], as well as the information available on the Sprint US backbone network [15].

5.1.1 IP Topology

A typical IP backbone contains a number of POPs interconnected with parallel IP links. The POPs contain backbone and access IP routers. Backbone routers that belong to the same POP are fully meshed. Each POP is connected with a subset of other POPs. In our simulation model, we assume that nodes in V represent POPs and links in L are inter–POP IP links (sets V and L are defined in Section 4.2). This assumption is obviously a simplification of a backbone topology. However, POPs are designed to be reliable. The backbone routers within a POP are fully meshed and access routers connect to more than one backbone router. This architecture makes the faster local restoration possible. Also, because routers within a POP are collocated, the restoration does not impact the end–to–end delay [14]. This is why our focus is

on inter–POP link failures. We assume that multiple inter–POP link failures are due to a router failure within a POP. We also assume that a node failure is a multiple link failure event where all the inter–POP links for the POP have failed. To create an IP topology, with these assumptions in mind, the following data has to be defined: a) POPs, b) for each POP the set of physically connected neighboring POPs, c) number of parallel IP links between adjacent POPs, their capacity and propagation delay.

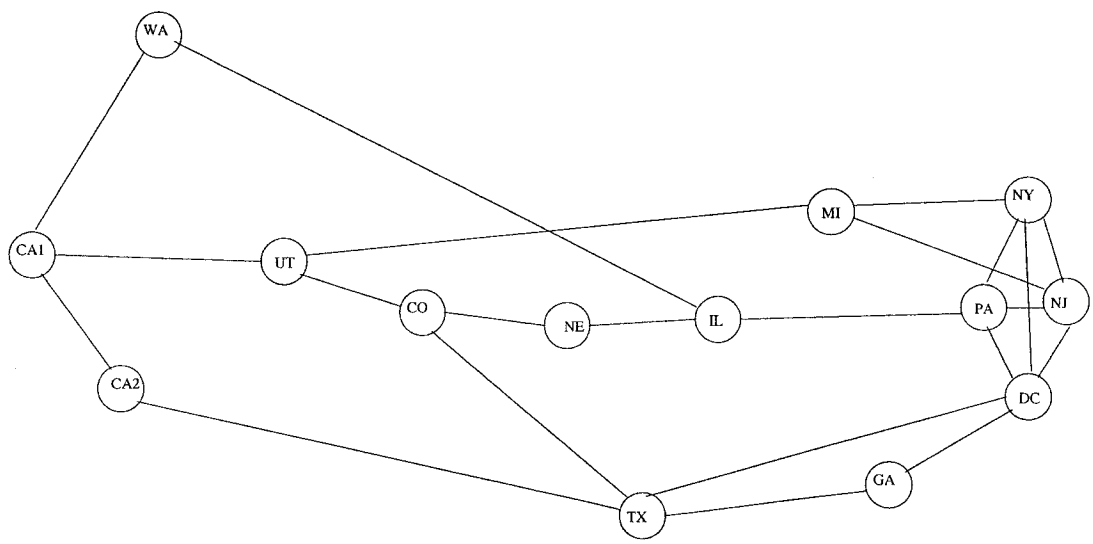

Figure 5.1: NSF optical network

Most real–world backbone networks are based on Dense Wavelength Division Multiplexing infrastructure. An IP network can be obtained from an optical network, by mapping a set of logical links (IP links) on the physical links in the optical network. This mapping is obtained as a solution to the Grooming, Routing and Wavelength Assignment (GRWA) problem [17]. Each logical link is assigned one wavelength or a set of wavelengths [15]. This is the approach we take (we assign one wavelength to

one IP link). We use the NSF optical network topology shown in Figure 5.1, and the total US traffic demand matrix for the NSP network given in [16] as a starting point. With this as the input information we solve the GRWA problem implemented by the GRWA with Tabu Search tool (GRWABOU tool) [17]. By mapping OC–48/OC–192 wavelengths obtained as the output to the GRWA problem to 2.5 Gbps/10Gbps directional IP links we obtain an IP topology based on the NSF optical network. As a result the nodes in the optical NSF network are POPs in our IP network. Optical node adjacencies are POP adjacencies. IP links are wavelengths obtained by the tool. Their capacity is determined by the type of the wavelength which is either OC-48 or OC-192. Propagation delay on IP links that connect adjacent POPs equals the the propagation delay on the corresponding fibers. The GRWABOU tool takes as an input a number of parameters. The one relevant to us is the maximum available number of wavelengths on a fiber in each direction. The number of wavelengths between adjacent nodes and their type OC–48/OC–192 obtained by the solution (that is the number and the type of parallel unidirectional IP links) will depend on this parameter as well as on the traffic demand matrix we use as an input. The solution to the problem studied in [15] was validated using the Sprint US backbone data. In their simulation model (a simplified version of the Sprint US backbone), the adjacent POPs are connected with a minimum of two, a maximum of six, and an average of three parallel bidirectional IP links. This is the guideline we used when we decided to set the maximum available number of wavelengths on a fiber to six. Also to obtain

the close numbers for the minimum and average number of parallel IP links we had to scale the elements of the total US traffic demand matrix by 50. We ended up with an IP topology that has 14 POPs whose physical connection is shown in Figure 5.1 . Table 5.1 gives the details on the IP topology we obtained. It is important to note that what we refer to as "links" in the table are unidirectional IP links.

Number of POPs	14
Total number of links	169
Total number of "OC-48" links	123
Total number of "OC-192" links	46
Average number of links between adjacent POPs	8
Minimum number of links between adjacent POPs	5
Maximum number of links between adjacent POPs	12
Minimum number of links with the same direction between adjacent POPs	2
Maximum number of links with the same direction between adjacent POPs	6

Table 5.1: IP topology

5.1.2 Traffic Matrix

Little information is available today on the dynamics of the traffic in an IP backbone. Often in simulation environments researchers assume a traffic matrix. A common approach is to pick, for a given source/destination pair, a traffic demand at random from a given range [1]. In our simulation model we use traffic matrices provided in [16]. The matrices are derived based on the statistical data such as distance between the cities/regions, population of the regions, number of employees, number of hosts etc. Matrices are provided for three types of data: voice, transaction data and IP data. A voice demand between two nodes (POPs) i and j is computed as a function

of the population of the regions Pi, Pj and their distance Dij: T_{voice} $(Pi * Pj)/Dij$. Transaction data demands are computed as a function of the number of employees Ei, Ej of the region and their distance Dij: T_{data} $(Ei * Ej)/\sqrt{Dij}$. IP data demands are computed as a function of the number of hosts Hi and Hj of the corresponding regions: T_{ip} $Hi * Hj$. Although neither of these matrices represents a real–time traffic matrix of a typical VPN customer, we found them useful as they provide different POP to POP demand distributions. The voice matrix has the largest range of POP–to–POP demand values. The ratio between the largest demand (element) in the voice matrix and the smallest demand (element) in the voice matrix is 221. This is because the demands are inversely proportional to the distance. For short distances and populous regions demands have large values, where for long distances and less populous regions demands are small. The data transaction matrix has somewhat more uniform distribution, because its demands are inversely proportional to the square root of the distance. The ratio between the largest and the smallest matrix element here is 34. And finally the IP data matrix has the most uniform distribution of the demands with the demand ratio 18. We scaled the elements in the three matrices to bring the sum of all the demands (all the elements in the matrix) for each of the matrices to approximately the same value as shown in Table 5.2. In this way, we ended up for all three traffic matrices with the same value of the total demand but different distributions of POP-to-POP demand values.

Total demand scaled by 50	265067
Total voice demand scaled by 100	35622
Total data demand scaled by 57	35527
Total IP demand scaled by 216	35654
Demand[Mb/s]	

Table 5.2: Total demand

5.1.3 Input Parameters

In Section 3.6 we defined parameter μ as a portion of the link capacity available to primary/backup O–LSPs. In our simulation scenarios we define μ as a global parameter, that applies to all the links. By changing its value we change the overall network capacity available to route primary and backup paths. In Section3.5 we introduced parameter λ as the maximum allowable difference in propagation delay between primary and backup O–LSPs defined as a percentage of the delay on the primary O–LSP. We change the values of μ and λ in the simulation scenarios and explore their impact on the quality of the solution. We give more detailed discussion in the remaining sections of this chapter.

In Section 3.4 we introduce parameter delay_max as a limit on the propagation delay of the potential paths. We set this parameter to 55ms based on the information available on the Sprint US backbone [15]. We showed in Section 4.5 that the number of the variables is a function of the size of the set of potential paths. For delay_max $= 55ms$ the number of paths in the set is in the order of million. To limit the size of the set we introduce parameter $lmax$ as the maximum number of links a potential path can have. We set this parameter to four for all the scenarios. Another

way would be to calcualte for each sd pair the shortest path and set the maximum number of links to be the length of the shortest path plus a constant.

5.2 Phase One

In the Phase One module we implement step one of the LP heuristic (S1), formulated in 4.4.1. Figure 4.2 depicts S1 in terms of its input and output data. In this section we present the simulation scenarios we have executed, and discuss the results we have obtained. All the simulation scenarios share the following input data: a) IP topology defined in Section 5.1, b) parameters delay_max and $lmax$ defined in Table 5.3. In the scenarios we will refer to the common input data as the common input.

delay_max [ms]	55
Maximum number of links in a path $= lmax$	4
Number of source/dest pairs with $T_{sd} > 0$	$\|V\|(\|V\| - 1) = 182$

Table 5.3: Phase One parameters

5.2.1 Simulation Scenarios and Results

Simulation Scenario 1:

INPUT: a) the common input b) traffic demand matrix, c) parameter $\mu = 0.5$

OUTPUT: primary paths computed for each sd pair with $T_{sd} > 0$.

Simulation Scenario 2:

INPUT: a) the common input b) traffic demand matrix, c) parameter $\mu = 0.8$

OUTPUT: primary tunnels (O–LSPs) computed for each sd pair with $T_{sd} > 0$.

We apply scenario 1 and scenario 2 to three different demand matrices defined in Section 5.1. The results of the execution for voice, transaction data and IP data traffic matrix are summarized in Tables 5.4, 5.6, and 5.5. Each table has the same format. The rows in the Tables are as follows:

The size of the set of potential paths $\bigcup_{sd \in \mathcal{SD}} \mathcal{P}^1_{sd}$. The size of the set depends on the IP topology and on the value of delay_max and $lmax$ parameters. Since for all the phase one scenarios we use the same IP topology, the maximum delay value 55ms, and the maximum number of links in the path 4, the size of the set is the same and equal to 105740. Note that the number of variables in the S1 equals the size of the set of potential paths for S1 (P1), see Section4.5.

Total number of primary tunnels. The minimum number of tunnels required in a solution equals the number of demands (elements) in the input traffic matrix whose bandwidth is greater then 0. In all three traffic matrices we have $Tsd > 0$ for all the sd pairs. The number of such pairs equals $|V|(|V| - 1) = 182$. Thus the minimal number of primary O–LSPs in our simulation scenario is 182. For a solution with 182 tunnels demand for each sd pair is routed over one primary O–LSP. If there is no enough bandwidth to place the whole Tsd demand on one of the potential sd paths from P1 and obtain the optimal solution, the algorithm will route the Tsd demand over multiple tunnels. In this case the total number of primary tunnels will be greater

then $|V|(|V|-1) = 182$.

Non-bifurcation level [%] . We define non–bifurcation level as a ratio between the minimal number of tunnels a solution can have, and the total number of primary tunnels. Non–bifurcation level serves as an indication of the management complexity of the solution. The complexity of the solution will be higher with the larger number of O–LSPs. Non-bifurcation level is inversly proportional to the number of O–LSPs. Values closer to 100% indicate better solution from the management complexity perspective.

Percentage of s/d pairs with the demand routed over multiple tunnels. Apart from the non–bifurcation level we are also interested in the number of *sd* pairs whose demands are routed over multiple O–LSPs. It has been proved in [1] that the total number of tunnels a solution has on top of the minimal number of tunnels is given by the number of links whose bandwidth is utilized to the allowed maximum in the optimization problem, i.e. bottleneck links. The number of s/d pairs with multiple tunnels will thus depend on the number of such links. For those links the inequality

$$\sum_{(v_s,v_d)\in\mathcal{SD}} T_{sd} \sum_{p\in\mathcal{P}^1_{sd}} \delta^p_\ell x^1_p \leq \mu\, u_\ell \qquad \ell \in L$$

becomes an equality in the solution. We can see that the number of LSPs in the solution can be controlled by the value of the parameter μ. An increase in its value for the bottleneck links will result in a decrease in the number of LSPs, i.e. it will increase the non–bifurcation level of the solution. In this way the heuristic provides a

trade–off between the link utilization (service quality experienced by the traffic) and the management complexity.

Minimum/maximum average delay. In heuristic S2 we define $\overline{d}_{sd}$ and use it to create the set of potential paths $\mathcal{P}^2_{sd}$, see Section 4.4.3. In all of the experiments we execute in the phase one we get the same values for the minimum and maximum path delay. This is because for each sd pair the shortest path is used to route primary tunnels. Also for the sd pairs with multiple tunnels the delay difference between the tunnels was zero.

Total link capacity used to route demands and total number of links over which demands routed. We calculate the total bandwidth on links reserved for primary tunnels as a measure of the cost of the solution. The same applies to the total number of links used on the primary paths.

Demand type = VOICE	$\mu = 0.5$	$\mu = 0.8$
$\lvert \bigcup_{sd \in \mathcal{SD}} \mathcal{P}^1_{sd} \rvert$	105740	105740
Total number of primary tunnels	189	183
Non-bifurcation level [%]	96.30	99.45
Percentage of s/d pairs with the demand routed over multiple tunnels	3.30	0.55
Minimum number of tunnels per s/d pair	1	1
Maximum number of tunnels per s/d pair	3	2
Minimum average delay $= \overline{d}_{sd}$ **[ms]**	1.04	1.04
Maximum average delay $= \overline{d}_{sd}$ **[ms]**	18.21	18.21
Total link capacity used to route demands [Mbps]	61714	61714
Total number of links over which demands routed	136	128

Table 5.4: Phase One results. Demand type VOICE

Demand type = DATA	$\mu = 0.5$	$\mu = 0.8$
$\lvert \bigcup_{sd \in \mathcal{SD}} \mathcal{P}^1_{sd} \rvert$	105740	105740
Total number of primary tunnels	187	183
Non-bifurcation level [%]	97.32	99.45
Percentage of s/d pairs with the demand routed over multiple tunnels	2.20	0.55
Minimum number of tunnels per s/d pair	1	1
Maximum number of tunnels per s/d pair	3	2
Minimum average delay $= \bar{d}_{sd}$ **[ms]**	1.04	1.04
Maximum average delay $= \bar{d}_{sd}$ **[ms]**	18.21	18.21
Total link capacity used to route demands[Mbps]	71694	71694
Total number of links over which demands routed	132	129

Table 5.5: Phase One results. Demand type DATA

Demand type = IP	$\mu = 0.5$	$\mu = 0.8$
$\lvert \bigcup_{sd \in \mathcal{SD}} \mathcal{P}^1_{sd} \rvert$	105740	105740
Total number of primary tunnels	191	184
Non-bifurcation level [%]	95.29	98.91
Percentage of s/d pairs with the demand routed over multiple tunnels	4.95	1.10
Minimum number of tunnels per s/d pair	1	1
Maximum number of tunnels per s/d pair	2	2
Minimum average delay $= \bar{d}_{sd}$ **[ms]**	1.04	18.21
Maximum average delay $= \bar{d}_{sd}$ **[ms]**	1.04	18.21
Total link capacity used to route demands [Mbps]	80166	80166
Total number of links over which demands routed	134	132

Table 5.6: Phase One results. Demand type IP

5.2.2 Analysis of the Results

By analyzing the data provided in Tables 5.4, 5.6, and 5.5 we can see that the non–bifurcation level depends on the value of μ. We already explained this tradeoff, and the results confirm our analysis. For $\mu = 0.5$ there are more O–LSP paths. Also the percentage of s/d pairs with the demand routed over multiple tunnels increases. When we use different traffic matrices we get a slightly different outcome in terms of

the non–bifurcation level and the total cost of the solution. All the traffic matrices place the same total demand on the network, however they differ in the distribution of the POP–to–POP demand values. IP data matrix has the most uniform distribution, which results in a slightly higher number of tunnels and a higher cost. The voice matrix has the highest level of variation in the value of demands, which results in the smallest overall cost and a smaller number of tunnels. The minimum/maximum average delay values are the same for all three matrices. This is because in all the scenarios there is enough bandwidth to place the demands onto the shortest delay paths.

5.3 Phase Two

In the Phase Two module we implement step two LP optimization problem (S2) formulated in Section 4.4.3. Figure 4.3 depicts the problem in terms of its input and output data. In this ection we present the simulation scenarios we execute in phase two, and discuss the results of execution. All the S2 simulation scenarios share the following input data: a) the residual IP topology, b) parameters delay_max and $lmax$ defined in Table 5.3. In the scenarios we will refer to the common input data as the common input – phase two. The residual IP topology is obtained from the IP topology used in S1 when we remove all the links whose residual capacity equals 0. The residual capacity of a link equals the capacity of the link u_l minus the total bandwidth reserved on the link for primary tunnels. The solution S1 (the primary

tunnel set) is used as the input data for the phase two module. We calculate backup tunnels using the jointness metric defined in Section 4.6 for values $C_1 = 10$ and $C_2 = 1000$.

5.3.1 Simulation Scenarios and Results

Simulation Scenario 1 – phase two:

INPUT: a) the common input – phase two b) traffic demand matrix, c) parameter $\mu = 0.5$, d) primary tunnel set, e) parameter $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 1, 100\}$

OUTPUT: secondary tunnels computed for each sd pair with $T_{sd} > 0$.

Simulation Scenario 2 – phase two:

INPUT: a) the common input – phase two b) traffic demand matrix, c) parameter $\mu = 0.8$, d) primary tunnel set, e) parameter $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 1, 100\}$

OUTPUT: secondary tunnels computed for each sd pair with $T_{sd} > 0$.

We apply simulation scenario 1 – phase one to three different matrices: voice, transaction data and IP data. For each of the primary sets obtained as a result we calculate the residual IP topology. The residual IP topology, the traffic matrix and the primary set then become the input to the simulation scenario 1 – phase two. We run the scenario 1 – phase two for different values of λ and for each obtain a backup set of tunnels. We repeat the same procedure for the simulation scenario 2 – phase one and the simulation scenario 2 – phase two combination. All the results are

summarized in Tables 5.7 to 5.12. Each table has the same format. The first part of the table contains the same type of data as calculated for phase one. The tables contain the following data:

The size of the set of potential paths $\bigcup_{sd \in \mathcal{SD}} \mathcal{P}^2_{sd}$. The set of potential paths in step two (P2) is a subset of the set of potential paths defined as the input to step one (P1). It contains all the paths that satisfy the relative delay criteria defined in 3.5. In other words, $\mathcal{P}^2_{sd} = \{p \in \mathcal{P}^1_{sd} : p \text{ is a path from } v_s \text{ to } v_d \text{ such that } |\mathsf{delay}(p) - \overline{d}_{sd}| \leq \lambda\, \overline{d}_{sd}\}$. Obviously the size of the set increases with the increase of λ. Also $\mathcal{P}^2_{sd}$ is calculated for the residual IP topology. Paths from $\mathcal{P}^1_{sd}$ that satisfy relative delay criteria but contain links with 0 residual capacity cannot be used to route backup tunnels. We do not include these paths when calculating the size of P2.

Total number of primary tunnels. The same definition applies as in Section 5.2.

Non-bifurcation level [%] .The same definition applies as in Section 5.2.

Percentage of s/d pairs with the demand routed over multiple tunnels. The same definition applies as in Section 5.2.

Maximum relative delay. In this row we provide the maximum value for the relative delay between primary and backup paths calculated in ms.

Total link capacity used to route demands and total number of links over which demands routed. The same definition applies as in Section 5.2.

Minimum/Maximum sd jointness. We define *sd* jointness as a measure of jointness of the backup tunnels and the primary set used to route a Tsd demand. We

calculate the minimum/maximum value of sd jointness over all sd pairs. In other words minimum sd jointness $= \min_{(v_s,v_d)\in\mathcal{SD}} \sum_{p'\in\mathcal{P}^2_{sd}} f_{p'}x^2_{p'}$ and maximum sd jointness $= \max_{(v_s,v_d)\in\mathcal{SD}} \sum_{p'\in\mathcal{P}^2_{sd}} f_{p'}x^2_{p'}$.

Total jointness. We define total jointness as a sum of sd jointness over all sd pairs. In other words total jointness $= \sum_{(v_s,v_d)\in\mathcal{SD}} \sum_{p'\in\mathcal{P}^2_{sd}} f_{p'}x^2_{p'}$. Total jointness is a measure of the reliability of the solution. It is a measure of the amount of resources shared by the primary and backup tunnels. Total jointness of zero indicates that all the primary/backup sets are link/node disjoint. This is the solution we desire. If it cannot find a backup tunnel node/link disjoint from the primary set S2 will find the maximally disjoint one. In this case total jointness will be higher then 0. In general a higher value of jointness indicates that there is more link/node sharing on primary/backup tunnels, thus the reliability of the solution is lower.

Percentage of s/d pairs with link/node disjoint primary and backup tunnels. Another value we define to quantify the quality of the solution of S2 is the percentage of sd pairs for which a link/node disjoint primary/backup tunnels were found.

Number of backup tunnels with 0/1 shared links. Our jointness metric is designed to give preference to link disjoint paths in case the algorithm cannot find link/node disjoint paths, see Section 4.6. We calculate the number of tunnels with 0/1 shared links. For the input data we use in our simulation scenarios the maximum number of links shared between a backup tunnel and a primary set is 1.

Demand type = VOICE $\mu = 0.5$							
λ	0	0.2	0.4	0.6	0.8	1	100
$\lvert \bigcup_{sd \in \mathcal{SD}} \mathcal{P}^2_{sd} \rvert$	I	24807	35507	46482	51473	55682	88853
Total number of backup tunnels	I	189	189	190	190	191	191
Non-bifurcation level [%]	I	96.30	96.30	95.79	95.79	95.28	95.28
Percentage of s/d pairs with the demand routed over multiple tunnels	I	3.85	3.23	4.40	4.40	4.94	3.30
Minimum number of tunnels per s/d pair	I	1	1	1	1	1	1
Maximum number of tunnels per s/d pair	I	2	3	2	2	2	4
Maximum relative delay [ms]	I	2.86	6.11	9.05	9.23	11.49	28.83
Total link capacity used to route demands [Mbps]	I	62346	62648	63074	65464	64790	71368
Total number of links over which demands routed	I	140	149	147	149	152	152
$\min_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	0	0	0	0	0	0
$\max_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	30	30	20	20	20	10
$\sum_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	1609.49	1040.11	697.34	601.459	456.671	80
Percentage of s/d pairs with link and node disjoint primary and backup tunnels	I	43.96	59.34	69.23	73.08	78.57	95.6
Number of backup tunnels with 0 shared links	I	189	189	190	190	191	191
Number of backup tunnels with 1 shared link	I	0	0	0	0	0	0

Table 5.7: Phase Two results. Demand type VOICE. Portion of a link capacity $\mu = 0.5$.

Demand type = VOICE $\mu = 0.8$							
λ	0	0.2	0.4	0.6	0.8	1	100
$\lvert \bigcup_{sd \in \mathcal{SD}} \mathcal{P}^2_{sd} \rvert$	I	29184	42963	55436	61190	66344	102974
Total number of backup tunnels	I	185	184	186	186	186	184
Non-bifurcation level [%]	I	98.37	98.91	97.85	97.85	97.85	98.91
Percentage of s/d pairs with the demand routed over multiple tunnels	I	1.65	1.09	2.20	2.20	2.20	1.09
Minimum number of tunnels per s/d pair	I	1	1	1	1	1	1
Maximum number of tunnels per s/d pair	I	2	2	2	2	2	2
Maximum relative delay [ms]	I	2.86	6.11	9.05	9.29	11.49	28.83
Total link capacity used to route demands [Mbps]	I	62346	62648	63074	63376	63114	68956
Total number of links over which demands routed	I	148	148	154	150	157	151
$\min_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	0	0	0	0	0	0
$\max_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	30	30	20	20	20	10
$\sum_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	1560	1000	660	580	440	80
Percentage of s/d pairs with link and node disjoint primary and backup tunnels	I	43.96	60.44	70.33	73.63	79.12	95.60
Number of backup tunnels with 0 shared links	I	185	184	186	186	186	184
Number of backup tunnels with 1 shared link	I	0	0	0	0	0	0

Table 5.8: Phase Two results. Demand type VOICE. Portion of a link capacity $\mu = 0.8$.

Demand type = DATA $\mu = 0.5$							
λ	0	0.2	0.4	0.6	0.8	1	100
$\lvert \bigcup_{sd \in \mathcal{SD}} \mathcal{P}^2_{sd} \rvert$	I	25944	38486	49399	54934	59631	93310
Total number of backup tunnels	I	198	196	198	193	198	196
Non-bifurcation level [%]	I	91.91	92.86	91.91	94.3	91.91	92.86
Percentage of s/d pairs with the demand routed over multiple tunnels	I	8.79	7.69	7.69	4.95	7.14	5.49
Minimum number of tunnels per s/d pair	I	1	1	1	1	1	1
Maximum number of tunnels per s/d pair	I	2	2	3	3	3	4
Maximum relative delay [ms]	I	2.86	6.71	9.05	9.29	11.49	28.83
Total link capacity used to route demands [Mbps]	I	73372	73826	74658	75456	76050	82844
Total number of links over which demands routed	I	152	152	153	154	158	155
$\min_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	0	0	0	0	0	0
$\max_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	196.28	30	20	20	20	10
$\sum_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	1756.27	1010	670	590	450	80
Percentage of s/d pairs with link and node disjoint primary and backup tunnels	I	43.956	60.44	70.33	73.62	79.12	95.60
Number of backup tunnels with 0 shared links	I	197	196	198	193	198	196
Number of backup tunnels with 1 shared link	I	1	0	0	0	0	0

Table 5.9: Phase Two results. Demand type DATA. Portion of a link capacity $\mu = 0.5$.

Demand type = DATA $\mu = 0.8$							
λ	0	0.2	0.4	0.6	0.8	1	100
$\lvert \bigcup_{sd \in \mathcal{SD}} \mathcal{P}^2_{sd} \rvert$	I	29536	43180	55669	61578	66706	103344
Total number of backup tunnels	I	186	186	186	189	189	185
Non-bifurcation level [%]	I	97.85	97.85	97.85	96.30	96.30	98.38
Percentage of s/d pairs with the demand routed over multiple tunnels	I	2.19	2.19	2.19	3.30	3.30	1.64
Minimum number of tunnels per s/d pair	I	1	1	1	1	1	1
Maximum number of tunnels per s/d pair	I	2	2	2	3	3	2
Maximum relative delay [ms]	I	2.86	6.11	9.05	9.29	11.49	28.83
Total link capacity used to route demands [Mbps]	I	73196	73738	74444	75590	75712	82518
Total number of links over which demands routed	I	148	150	155	158	156	155
$\min_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	0	0	0	0	0	0
$\max_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	30	30	20	20	20	10
$\sum_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	1570	1010	670	590	450	80
Percentage of s/d pairs with link and node disjoint primary and backup tunnels	I	43.95	60.44	70.33	73.62	79.12	95.64
Number of backup tunnels with 0 shared links	I	186	186	186	189	189	185
Number of backup tunnels with 1 shared link	I	0	0	0	0	0	0

Table 5.10: Phase Two results. Demand type DATA. Portion of link capacity $\mu = 0.8$.

Demand type = IP $\mu = 0.5$							
λ	0	0.2	0.4	0.6	0.8	1	100
$\lvert \bigcup_{sd \in \mathcal{SD}} \mathcal{P}^2_{sd} \rvert$	I	23847	35877	45195	50519	55079	86251
Total number of backup tunnels	I	197	202	200	197	200	204
Non-bifurcation level [%]	I	92.38	90.09	91.00	92.38	91.00	89.21
Percentage of s/d pairs with the demand routed over multiple tunnels	I	7.14	9.89	9.34	7.69	8.24	10.43
Minimum number of tunnels per s/d pair	I	1	1	1	1	1	1
Maximum number of tunnels per s/d pair	I	3	3	3	3	3	4
Maximum relative delay [ms]	I	2.86	6.11	9.05	9.29	11.49	28.83
Total link capacity used to route demands [Mbps]	I	82871	83984	85260	86592	86897	96577
Total number of links over which demands routed	I	143	146	151	149	153	152
$\min_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	0	0	0	0	0	0
$\max_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	1010	1000	1000	1000	1000	10
$\sum_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	3620	2050	1710	1630	1470	80
Percentage of s/d pairs with link and node disjoint primary and backup tunnels	I	43.40	59.89	69.78	73.07	78.57	95.60
Number of backup tunnels with 0 shared links	I	195	201	199	196	199	204
Number of backup tunnels with 1 shared link	I	2	1	1	1	1	0

Table 5.11: Phase Two results. Demand type IP. Portion of link capacity $\mu = 0.5$

Demand type = IP $\mu = 0.8$							
λ	0	0.2	0.4	0.6	0.8	1	100
$\lvert \bigcup_{sd \in \mathcal{SD}} \mathcal{P}^2_{sd} \rvert$	I	28676	42236	54141	60002	65126	100836
Total number of backup tunnels	I	189	187	188	187	188	190
Non-bifurcation level [%]	I	96.29	97.32	96.80	97.32	96.8	95.78
Percentage of s/d pairs with the demand routed over multiple tunnels	I	3.29	2.74	3.29	2.74	3.29	4.39
Minimum number of tunnels per s/d pair	I	1	1	1	1	1	1
Maximum number of tunnels per s/d pair	I	3	2	2	2	2	2
Maximum relative delay [ms]	I	2.86	6.11	9.05	9.29	11.49	28.83
Total link capacity used to route demands [Mbps]	I	82754	83658	84776	85182	86430	94992
Total number of links over which demands routed	I	145	151	150	153	157	149
$\min_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	0	0	0	0	0	0
$\max_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	30	30	20	20	20	10
$\sum_{(v_s,v_d) \in \mathcal{SD}} \sum_{p' \in \mathcal{P}^2_{sd}} f_{p'} x^2_{p'}$	I	1560	1000	660	580	440	80
Percentage of s/d pairs with link and node disjoint primary and backup tunnels	I	43.95	60.44	70.33	73.62	79.12	95.60
Number of backup tunnels with 0 shared links	I	189	187	188	187	188	190
Number of backup tunnels with 1 shared link	I	0	0	0	0	0	0

Table 5.12: Phase Two results. Demand type IP. Portion of link capacity $\mu = 0.8$.

5.3.2 Analysis of the Results

We define total jointness as an indicator of the reliability of the solution to the problem of routing of primary/backup real–time O–LSP tunnels. The question we are interested in is: What does total jointness depend on? Figures 5.2, 5.3, 5.4 show the dependency between total jointness and parameter λ (relative delay) for voice, transaction data and IP matrices respectively. Two curves are shown for two different values of μ. We can see that total jointness decreases for larger values of λ and μ.

Larger values of λ result in a larger value of the maximum allowable propagation delay difference between a primary O–LSP and a backup O–LSP, see Section 3.5, and consequently in a larger size of P2 set. Primary tunnels are routed over the shortest paths. Often paths that are link/node disjoint from the shortest path include larger number of links, which results in a longer propagation delay. This of course depends on the network topology. In general the more we relax the relative delay constraint, the higher is the number of paths in P2 set that share smaller number of nodes/links with the shortest path. As a consequence of this the solution to S2 for a larger value of λ will have a set of backup tunnels that are less joint with the corresponding primary sets.

We can also observe that the value of μ has the impact on total jointness. For the same value of parameter λ we get smaller total jointness for larger values of μ. Parameter μ defines the portion of the link capacity available to route primary/backup paths. Obviously for larger μ, the residual capacity of the links will be larger. P2

set is defined on the residual IP topology, as a subset of P1. For the same value of λ and the same traffic matrix, we will get the same P2 set regardless of the value of μ. However, the maximal bandwidth that can be allocated on the paths that belong to P2 set will be smaller for smaller μ. In that case potential paths with lower failure cost may not have enough bandwidth to route a demand. In order to route a demand, paths with higher failure cost have to be used, which results in larger total jointness.

For the input data we use in the simulation, the minimal total jointness we are able to achieve is 80. We get this value for $\lambda = 100$. We get the same value for all three traffic matrices. For such a large value of λ the P2 set includes all of the paths from P1 set that contain links with residual capacity greater than zero. Still, we were unable to find link/node disjoint primary/backup sets for all *sd* pairs. The reason is the limit on the number of links $lmax = 4$. There are 8 *sd* pairs for which there are no link/node disjoint paths that contain the number of links smaller of equal to 4. An example is the source/destination pair CA1 – MI. The shortest path for this pair has two links. All link/node disjoint paths have more than 4 links. This is the reason we could not achieve total jointness of zero. However, the percentage of *sd* pairs for which we were able to obtain a link/node disjoint solution is 95.8. For 8 out of 182 *sd* pairs, the backup paths are joint in one node with the primary set.

In Figures 5.5 to 5.8 we show on the same plot total jointness curves for all three traffic matrices. We can see that the worst results we obtained for IP data. IP data has the most uniform distribution and as a result the total bandwidth on links

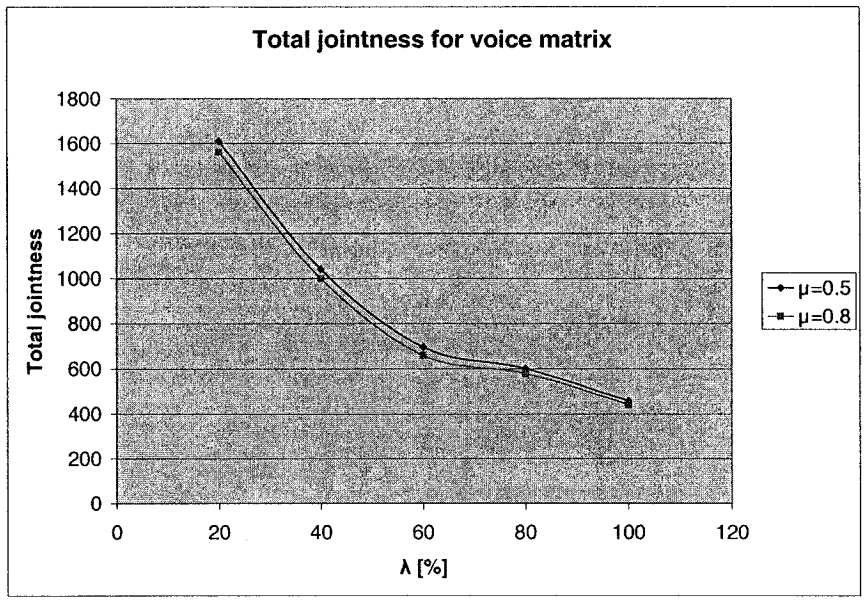

Figure 5.2: Total jointness curves for voice matrix: $\mu = 0.5$ and $\mu = 0.8$.

reserved for primary tunnels is the largest. This makes the residual capacity on links smaller than for other traffic types, and consequently total jointness larger.

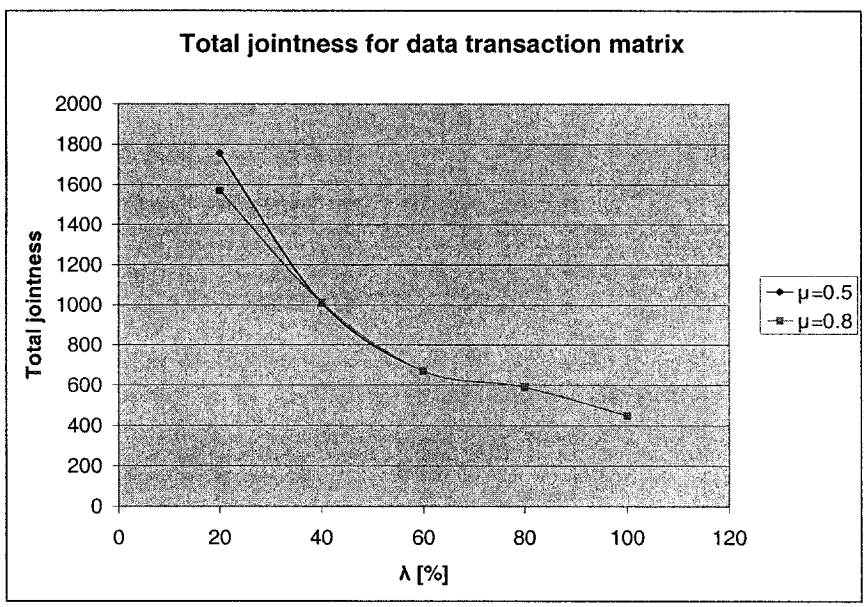

Figure 5.3: Total jointness curves for transaction data matrix: $\mu = 0.5$ and $\mu = 0.8$.

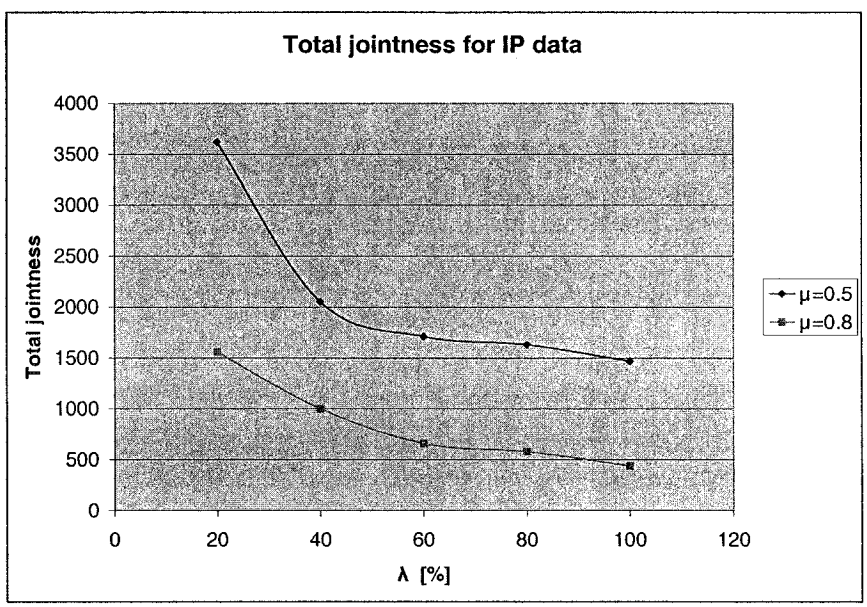

Figure 5.4: Total jointness curves for IP data matrix: $\mu = 0.5$ and $\mu = 0.8$.

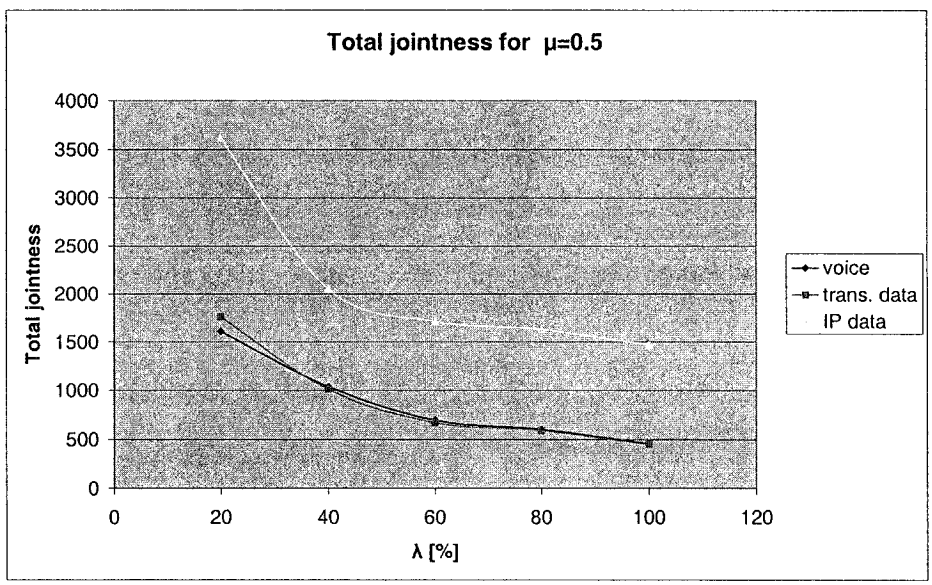

Figure 5.5: Total jointness curves for $\mu = 0.5$ and voice, transaction data and IP data matrices.

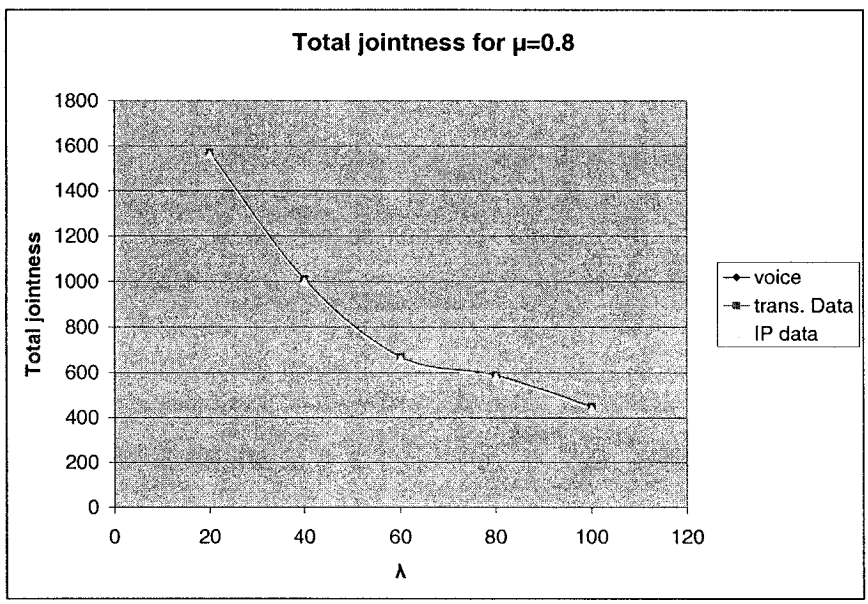

Figure 5.6: Total jointness curves for $\mu = 0.8$ and voice, transaction data and IP data matrices.

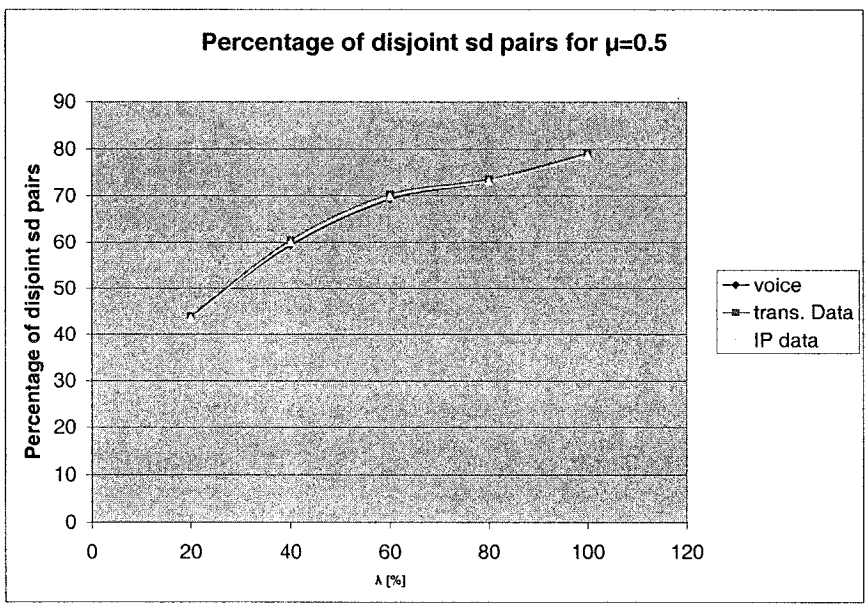

Figure 5.7: The percentage of link/node disjoint sd pairs for $\mu = 0.5$ and voice, transaction data and IP data matrices.

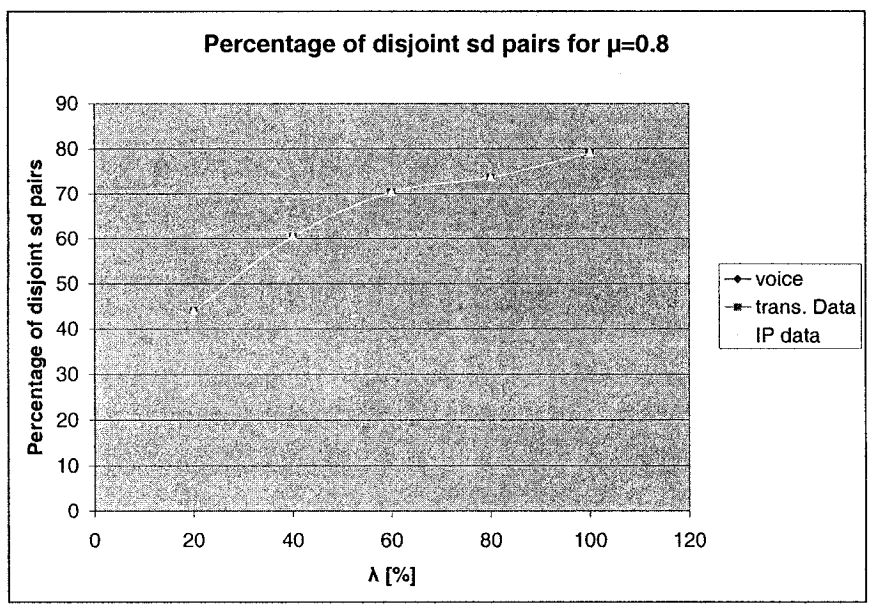

Figure 5.8: The percentage of link/node disjoint sd pairs for $\mu = 0.8$ and voice, transaction data and IP data matrices.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

We have proposed a solution to the problem of routing real–time VPN demands in DiffServ/MPLS VPN network (VPN RT–TE problem). We assume that a path protection mechanism is used to provide a reliable service for real–time applications. Our solution encompasses the problem of routing both primary and backup MPLS tunnels. We have formulated an MILP optimization problem that takes into account all the VPN RT–TE problem requirements in one step – the integrated model. The integrated model is NP-complete and has a large number of variables and constraints. We have proposed a two–step heuristic that can be solved in a context of a backbone MPLS VPN network. Each step of the heuristic is an LP optimization problem. The LP formulation of the VPN RT–TE problem leads to routing of the VPN demands over multiple paths between ingress – egress node pairs. Each VPN demand is routed over one path, however different VPN demands that have the same ingress and egress

nodes may be routed over different paths. Multipath routing allows for a more efficient use of the resources, at the cost of the increase in management complexity. This tradeoff is controllable. The number of MPLS tunnels in the solution depends on a parameter of the heuristic, namely the maximum allowable bandwidth that can be reserved on a link for primary/backup tunnels.

We focused on the minimization of the propagation delay of primary MPLS tunnels, and the minimization of the network resources shared by the primary and backup MPLS tunnels. We have also taken into account the relative delay and maximum ingress/egress delay requirement for the primary/backup tunnels. We introduced the following assumptions: a restoration/protection mechanism (such as SONET protection) is available that takes care of the failures at the layers below IP, and link and node failures are uniformly distributed across all links/nodes. Under these assumptions we developed a jointness metric as a measure of jointness at the IP level of a backup tunnel and a primary set of tunnels. The result of the two–step heuristic are primary/backup tunnel sets that satisfy relative and maximum delay requirements and are maximally disjoint. To our knowledge we are the first to formulate the VPN RT–TE problem and to provide a solution in a form of a two–step LP heuristic.

We have created a simulation model that approximates a real–world backbone network. We have applied the heuristic to the model IP backbone network based on the NSF optical network topology. We dimensioned the IP network in terms of

the number of IP links based on the model of the Sprint US backbone network from [15]. In our simulation scenarios we used demand traffic matrices with different demand distributions, that are the scaled version of the US traffic demand matrices provided in [16]. We have demonstrated the relative delay – jointness tradeoff of the solution. To obtain better reliability of the solution (smaller jointness) we had to relax the relative delay constraint (decrease the performance quality of the solution). We also showed that with an increase of the available link capacity the management complexity (number of MPLS primary/backup tunnels) of the solution decreases and reliability of the solution increases. We demonstrated that the quality of the solution depends on the distribution of the demands in a traffic matrix and on the network topology. For the IP network model we created, with the limit on the maximum number of links in a path set to 4, we were unable to find link/node disjoint primary/backup tunnels for all ingress/egress pairs, even when we removed the relative delay constraint. However, we were able to find link/node disjoint primary/backup tunnels for 95.6% of ingress/egress pairs. We feel that the simulation scenarios we carried out produced insightful results.

The problem of routing primary and backup tunnels can arise in overlay networks other then MPLS VPNs. Therefore, the two–step heuristic is applicable in other settings.

6.2 Future Work

In the future we would like to:

- Solve the VPN RT–TE problem for other backup bandwidth allocation schemes. We have formulated VPN RT–TE problem so that the bandwidth reserved on backup tunnels equals the bandwidth allocated to corresponding primary tunnels. A different scheme can be used. The resources allocated for a backup tunnel can be shared by another backup tunnel if their corresponding primary tunnels do not share any resources. We would like to extend the VPN RT–TE problem to include this shared bandwidth allocation scheme.

- Develop a new jointness metrics assuming a non–uniform distribution of link/node failures over all links/nodes in the network. Link/node failures are typically not uniformly distributed over all links/nodes in a network. There are some links/nodes that fail more often than others.

- Develop a new jointness metrics relaxing the assumption on the existence of a protection/restoration mechanism at the layers below IP. In this case the failures at the layers below IP are visible at the IP layer for the links that share some lower layer infrastructure (i. e. traverse a common fiber). This has to be taken into account by the jointness metric.

Bibliography

[1] X. Xiao, A. Hannan, B. Bailey, L.M. Ni, "Traffic engineering with MPLS in the Internet" *IEEE Network*, 14(2), pp.28–33,2000.

[2] M.K. Girish, Z. Bei, J. Hu, "Formulation of the traffic engineering problems in MPLS based IP networks", *ISCC* 2000.

[3] Y. Wang, Z. Wang, "Explicit routing algorithms for Internet traffic engineering" *Proceedings of the 8th International Conference on Computer Communications and Networks*, pp.582–588, 1999.

[4] F. Le Faucheur, W. Lai, *Requirements for Support of Differentiated Services–aware MPLS Traffic Engineering*, RFC 3564, July 2003.

[5] C. T. Chou, "Traffic engineering for MPLS-based virtual private networks", *Computer Networks*, Volume 44 , Issue 3, pp. 319 - 333 ,February 2004.

[6] Y. Lee, C. Kim, "A constrained multipath traffic engineering scheme for MPLS networks", *IEEE International Conference on Communications*, pp. 2431-2436, 2002.

[7] W. Cui, I. Stoica, R. H. Katz, "Backup path allocation based on a correlated link failure probability model in overlay networks", *ICNP Proceedings of the 10th IEEE International Conference on Network Protocols*, pp.236, 2002.

[8] Murali S. Kodialam, T. V. Lakshman "Dynamic Routing of Bandwidth Guaranteed Tunnels with Restoration", *INFOCOM*, pp.902-911,2000.

[9] I. Pepelnjak J. Guichard, *MPLS and VPN Architectures*, Cisco Press, 2001.

[10] D.D. Clark, K. Solins, J. Wroclawski, T. Faber, "Addressing Reality: An Architectural Response to Real-World Demands on the evolving Internet", *ACM SIGCOMM 2003 Workshops, Krlsruhe, Germany*, August, 2003.

[11] R. Chandra, D. Katz, Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 2283, February 1998.

[12] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, C. Diot, "Characterization of Failures in an IP Backbone", *in Proc. of. IEEE Infocom*, 2004.

[13] F. A. Tobagi, A. P. Markopoulou, M. J. Karam, "Is the Internet ready for VoIP?", *In Proc. IWDC, Capri, Italy*, p. 49-57, Sept.,2002.

[14] G. Iannaccone, C.Chuah, S. Bhattacharyya, C. Diot, "Feasibility of IP Restoration in a Tier-1 Backbone", *IEEE Network Magazine (Special Issue on Protection, Restoration and Disaster Recovery)*, March 2004.

[15] F. Giroire and A. Nucci and N. Taft and C. Diot, "Increasing the Robustness of IP Backbones in the Absence of Optical Level Protection", *IEEE Infocom. San Francisco*, March 2003.

[16] A. Betker, C. Gerlach, R. Hulsermann, M. Jager, M. Barry, S. Bodamer, J. Spath, C. Gauger, M. Kohn, "Reference transport network scenarios", *Technical report, BMBF-Projekt MultiTeraNet*, 2003.

[17] A. C. Houle, B. Jaumard, Y. Solari, "Addressing the GRWA Problem in WDM Networks with a Tabu Search Algorithm", *CCECE/CCGEI, Saskatoon* , May, 2001.

[18] Y. Solari , "Une métaheuristique pour le problème d'affectation de longueurs d'onde, de groupage et de routage du trafic dans les réseaux optiques WDM", MSc Thesis, University of Montreal, submitted.

[19] ILOG, ILOG CPLEX 7.0 Reference Manual, 2000.

[20] ILOG. http://www.ILOG.com/products/cplex/

[21] T. H. Cormen, C. E. Leirserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, MIT Press, 2 edition, 2001.

[22] M. S. Bazaraa, J.J. Jarvis, H. D. Sherali, *Linear Programming and Newtork Flows,chapter 8*, 2nd ed., John Wiley and Sons Inc. , 1990.