Provisioning virtual private networks in the hose model

Wang, Haibo

2007

Wang, H. (2007). Provisioning virtual private networks in the hose model. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/3456>

<https://doi.org/10.32657/10356/3456>

Nanyang Technological University

Downloaded on 26 Apr 2023 20:28:00 SGT

Provisioning Virtual Private Networks in the Hose Model

Wang Haibo

School of Electrical & Electronic Engineering

A thesis presented to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2007

I hereby certify that the content of this thesis is the result of work done by me and has not been submitted for a higher degree to any other University or Institution.

.....
Date

.....
Wang Haibo

Acknowledgments

First I owe great thanks to my supervisor Associate Professor Poo Gee Swee. If not for his stringent requirement for academic standards, I would not have achieved this much as I currently have. He transformed me from a know-nothing-of-research student to a researcher.

Secondly I heartily thank my fellow students in both Comms IV lab and NTRC, particularly Mr. Yin Yongning, Mr. Chen Sheng, Mr. He Xuzhou, Mr. Zhou Yin Zhu, and Mr. Yong Keen Mum. Your friendship and sincere help make my life a lot easier and colorful. I really appreciate the time spent with you, and it will be a precious and sweet memory for me. My thanks also extend to the technicians in both labs. Without the necessary hardware and software environment you have provided, my research work would remain academic ideas.

In particular, I cannot express my thanks enough to Dr. Tong Ming Sze and Mr. Gu Jian. Conversations with them often turn out to be enlightening and encouraging. Having true friends around is really a big fortune in life.

Most importantly, I dedicate this work to my family, especially my wife, Li Xiaoling. In order for me to pursue a PhD degree abroad, they had to bear a lot of hardship at home. Their endless love, great sacrifice, and full support are essential ingredients of this work.

At last, I thank Nanyang Technological University for providing me with this great opportunity and scholarship for my PhD study.

Table of Contents

Acknowledgments.....	i
Summary.....	vii
List of Figures.....	ix
List of Tables.....	xiii
Acronyms.....	xv
Notations.....	xvii
Chapter 1 Introduction.....	1
1.1 Background.....	2
1.1.1 VPNs.....	2
1.1.2 The Pipe Model and the Hose Model	3
1.2 Motivations	6
1.3 Objectives	7
1.4 Methodologies and Approaches.....	8
1.5 Major Contributions of the Thesis	9
1.6 Organization of the Thesis	11
Chapter 2 Literature Review.....	13
2.1 VPN Solutions	13
2.1.2 IP/MPLS VPNs.....	17
2.1.3 Optical VPNs	20
2.2 Related Works.....	24
2.2.1 IP/MPLS VPNs.....	24
2.2.2 Optical VPNs	27
2.2.3 The Hose Model.....	29
2.2.4 Blocking Probability Calculation.....	33

2.2.5	Availability Guaranteed Services	34
2.2.6	Load Balancing	36
2.3	Fundamentals of the Hose Model	39
2.3.1	Formal Definition of the Hose Model.....	39
2.3.2	Minimum Link Capacity Calculation in the Hose Model.....	39
2.3.3	Minimum Link Capacity Calculation in Tree Routing	41
2.3.4	Multi-Path Routing Linear Program	42
2.4	Summary	43
Chapter 3	Blocking Performance of the Pipe Model and the Hose Model	45
3.1	Introduction.....	45
3.2	Analytical Models.....	46
3.2.1	Incorporating Traffic Bounds into the Analytical Model	50
3.2.2	Applying the Analytical Model to the Pipe Model.....	51
3.2.3	Applying the Analytical Model to the Hose Model.....	52
3.3	Discrete Event Simulations.....	53
3.4	Numerical Results and Discussions	56
3.4.1	Performance of the Pipe Model	56
3.4.2	Performance of the Hose Model	57
3.4.3	Performance Comparison between the Two Models	58
3.4.4	Average Network Blocking Probabilities for the Two Models	60
3.5	Conclusion	63
Chapter 4	Tree Routing and Multi-Path Routing for the Hose Model	67
4.1	Introduction.....	67
4.2	Bandwidth Efficiency of Tree Routing and Multi-Path Routing.....	69
4.2.1	Algorithm for Tree Routing	69

4.2.2	Algorithm for Multi-Path Routing	70
4.2.3	Sub-Provisioning and Admission Control	72
4.3	Blocking Performance for Tree Routing and Multi-Path Routing.....	74
4.3.1	General Assumptions	74
4.3.2	Approximation in Calculating $P_{u,v}^o(m, k)$	79
4.4	Experiments and Numerical Results	80
4.4.1	Bandwidth Requirements of Two Routing Schemes when $F=1$	83
4.4.2	Bandwidth Requirement of Multi-Path Routing When Varying F	87
4.4.3	Blocking Performance of Multi-Path Routing and Tree Routing.....	91
4.4.4	Constraining the Number of Paths	98
4.5	Conclusions.....	102
Chapter 5	Providing Availability Guarantee in Hose Model VPNs	105
5.1	Introduction.....	105
5.2	Preliminaries of Availability Guaranteed Service	106
5.2.1	Joint Availability Calculation	106
5.2.2	Preference of Link-Disjoint paths.....	109
5.3	Two Multi-Path Routing Heuristics.....	110
5.3.1	Problem Formulation	110
5.3.2	Availability-Centric Approach.....	112
5.3.3	Bandwidth-Centric Approach	113
5.4	Admission Control.....	118
5.5	Simulations and Numerical Results	121
5.5.1	Joint Availabilities of Tree Routing and Multi-Path Routing.....	123
5.5.2	Effects of Three Types of Blocking.....	125
5.5.3	Total Blocking	129

5.5.4	Comparison between the Two Heuristic Approaches.....	131
5.6	Conclusion	134
Chapter 6	Load Balancing in Hose Model VPNs with Multi-Path Routing	137
6.1	Introduction.....	137
6.2	Load Balancing in Hose Model VPNs.....	137
6.3	Load Balancing with Multi-Objective Multi-Path Routing Algorithm	140
6.3.1	Problem Formulation	140
6.3.2	The Multi-Objective Multi-Path Routing Algorithm.....	141
6.3.3	Multiple Path Guarantee with the MFTP constraint.....	143
6.4	Other Hose Model VPN Provisioning Algorithms	144
6.4.1	Tree Routing Algorithm.....	144
6.4.2	Single Objective Multi-Path (SOMP) Routing Algorithm	144
6.5	Numerical Results and Discussions.....	145
6.5.1	Typical Traffic Distribution.....	146
6.5.2	Service Provider Level Load Balancing without MFTP Constraint..	149
6.5.3	Service Provider Level Load Balancing with MFTP Constraint.....	150
6.5.4	Bandwidth Overprovisioning for Load Balancing.....	152
6.5.5	Multiple Paths for VPN Customer Level Load Balancing	154
6.6	Summary	156
Chapter 7	Conclusions and Recommendations	159
7.1	Conclusions.....	159
7.2	Recommendation for Future Work	162
Author's publications.....		165
Bibliography.....		167

Summary

Virtual private networks (VPNs) can provide customers with cost-effective, reliable, and secure services. Currently there are two resource management models for provisioning VPN services, the pipe model and the hose model. The hose model only requires customers to specify an aggregate ingress and egress bandwidth capacities per-endpoint, instead of an exact traffic demand per endpoint pair as with the pipe model. This thesis investigates hose model VPN provisioning algorithms on the following important but neglected aspects.

Firstly the blocking performance of the hose model and the pipe model in provisioning bandwidth-guaranteed connections are analyzed and compared, using both analytical models and discrete-event simulations. This study shows that the hose model has a blocking performance better than the pipe model by several orders of magnitude under normal network operating loads, contributing to the justification of using the hose model as the preferred resource management model for VPN service provisioning.

Secondly the hose model can be implemented in tree routing, single-path routing and multi-path routing. In this part, a thorough study of different routing schemes for the hose model is presented, from the cost as well as the blocking performance perspectives. In addition, a new concept of sub-provisioning is proposed to overcome the bandwidth overprovisioning problem associated with the hose model. The results show that compared with tree routing, multi-path routing only brings in marginal overprovisioning at rare combinations of network topologies and hose parameters, but it delivers a much better blocking performance at a fixed amount of provisioning.

Thirdly, two multi-path routing heuristics are proposed for the provisioning of availability-guaranteed services in hose model VPNs, the bandwidth-centric approach and the availability-centric approach. The two approaches both employ multiple link-disjoint paths to provide higher joint availabilities; they differ in the way to locate these paths. Compared with tree routing or single-path routing, the proposed multi-path routing heuristics can reduce the blocking probabilities in provisioning connection requests with availability requirements by a few orders of magnitude, and also make the provisioning process fast and straightforward.

Finally, a multi-objective multi-path (MOMP) routing linear programming approach is proposed for load balancing at both service provider level and VPN customer level. Service provider level load balancing significantly reduces the bandwidth reservation on the most loaded link, where congestions are most likely to occur; VPN customer level load balancing ensures that there are multiple paths between each VPN endpoint pair, thus allowing load balancing within the VPN.

In summary, this Thesis has demonstrated the hose model as the preferred resource management model for VPN service provisioning, and further established multi-path routing as the preferred routing scheme. It provides a comprehensive solution to VPN service provisioning in the hose model, heeding practical concerns of reliability and traffic engineering. The research works presented make good use of both theoretical analysis and discrete-event simulations, which render the thesis concrete and solid.

List of Figures

Figure 1.1 A VPN example.....	3
Figure 1.2 The pipe model and the hose model.....	4
Figure 2.1 VPN Reference Models.....	15
Figure 2.2 BGP/MPLS VPN example	19
Figure 2.3 Min-cost flow formulation	40
Figure 3.1 Illustrative example	48
Figure 3.2 Example on node-splitting.....	51
Figure 3.3 A pipe example.....	51
Figure 3.4 The NSFNET topology.....	53
Figure 3.5 Best-cost tree for the hose model	56
Figure 3.6 Analytical vs simulation results for the pipe model	57
Figure 3.7 Analytical vs simulation results for the hose model.....	58
Figure 3.8 Performance comparison for a single endpoint pair.....	59
Figure 3.9 Average class blocking probabilities for the pipe model	60
Figure 3.10 Average class blocking probabilities for the hose model.....	61
Figure 3.11 Average class blocking probabilities for the two models.....	61
Figure 3.12 Network blocking probabilities for the two models.....	63
Figure 4.1 Sparse network topology with 15 nodes.....	81
Figure 4.2 Dense network topology with 50 nodes	81
Figure 4.3 Average bandwidth requirements.....	85
Figure 4.4 Overprovisioning factor vs number of nodes	87
Figure 4.5 Bandwidth increase with sparse topologies.....	89
Figure 4.6 Bandwidth increase with dense topologies.....	89
Figure 4.7 Average bandwidth increase.....	90

Figure 4.8 Overprovisioning factor vs F	90
Figure 4.9 Blocking probabilities with full provisioning, $F=0.9$	91
Figure 4.10 Blocking probabilities with 90% provisioning.....	92
Figure 4.11 Blocking probabilities with 50% provisioning.....	92
Figure 4.12 Load vs provisioning percentage for fixed blocking probability	94
Figure 4.13 Blocking probabilities vs F	95
Figure 4.14 Number of paths between a VPN endpoint pair	96
Figure 4.15 Analytical vs simulation blocking probabilities	97
Figure 4.16 Path aggregation algorithm to remove excessive paths.....	100
Figure 4.17 Maximum and average number of paths	101
Figure 4.18 blocking probability before and after aggregation	101
Figure 5.1 Overlapping path example.....	107
Figure 5.2 Complete bandwidth-centric approach.....	117
Figure 5.3 Joint availabilities.....	124
Figure 5.4 Various types of blockings for tree routing.....	125
Figure 5.5 Various types of blockings for bandwidth-centric approach.....	127
Figure 5.6 Various types of blockings for availability-centric approach.....	127
Figure 5.7 Total blocking of tree routing and two heuristics with $b=1$	129
Figure 5.8 Total blocking of tree routing and two heuristics with $b=5$	130
Figure 5.9 Blocking probabilities versus scaling factors, $b=1$	132
Figure 5.10 Blocking probabilities versus scaling factors, $b=5$	132
Figure 5.11 Overprovisioning of two multi-path approaches.....	133
Figure 6.1 an example for load distribution.....	138
Figure 6.2 Traffic distribution with tree routing.....	147
Figure 6.3 Traffic distribution with SOMP routing.....	147

Figure 6.4 Traffic distribution with MOMP routing.....	148
Figure 6.5 Maximum link bandwidth reservation.....	149
Figure 6.6 Bandwidth reduction of MOMP routing to tree routing versus F	150
Figure 6.7 Bandwidth reduction of MOMP routing to SOMP routing versus F	152
Figure 6.8 Maximum and average number of paths	155

List of Tables

Table 3.1 Traffic matrix for the pipe model	54
Table 3.2 Traffic parameters for the hose model	54
Table 4.1 Hose ingress and egress parameters.....	82
Table 4.2. Bandwidth savings of multi-path routing	83
Table 4.3. F for multi-path routing vs scaling factor for tree routing.....	95
Table 4.4 Bandwidth increase after path aggregation.....	101
Table 6.1 Bandwidth Reduction to SOMP Routing and Tree Routing.....	150
Table 6.2 Overprovisioning to Tree Routing and SOMP routing.....	153
Table 6.3 Effects of aggregation on bandwidth capacities	156

Acronyms

ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
CE	Customer Edge Devices
CPI	Customer Port Index
DiffServ	Differentiated Service
GMPLS	Generic MultiProtocol Label Switching
IETF	Internet Engineering Task Force
IntServ	Integrated Service
IP	Internet Protocol
IPCC	IP Control Channel
IPSec	IP Security
L1VPN	Layer-1 VPN
L2VPN	Layer-2 VPN
L3VPN	Layer-3 VPN
LDM	Load Distribution to Multipath
LSP	Label Switched Path
LSR	Label Switched Router
LP	Linear Program
MFTP	Maximum Fraction of Traffic on a Path
MOMP	Multi-Objective Multi-Path
MP-BGP	BGP MultiProtocol Extension
MPLS	MultiProtocol Label Switching
OADM	Optical Add-Drop Multiplexer
O-E-O	Optical-Electrical-Optical
OVPN	Optical Virtual Private Network
OXC	Optical Cross-Connect
PE	Provider Edge Devices
PN	Private Network
PPI	Provider Port Index
PPVPN	Provider Provisioned VPN
RSVP	Resource ReserVation Protocol
RSVP-TE	RSVP Traffic Engineering Extension
RT	Route Target
SLA	Service Level Agreement
SOMP	Single-Objective Multi-Path
SP	Service Provider
VPN	Virtual Private Network
VR	Virtual Router
WDM	Wavelength Division Multiplexing

Notations

$G=(V,E)$	Graph with nodes V and bidirectional edges E
Q	The set of VPN endpoints, $Q \in V$
$d_{u,v}$	Traffic demand between endpoint (u,v)
D	A valid traffic matrix, $d_{u,v} \in D$
$\mathcal{D}$	The set of all valid traffic matrix, $D \in \mathcal{D}$
x_e	Bandwidth reservation on edge e
$x_{\max}$	Maximum bandwidth reservation among all the network links
c_e	Cost of per-unit bandwidth on edge e
F	The maximum fraction of traffic on a path (MFTP) constraint
$f_{u,v}^e$	Fraction of traffic between endpoint (u,v) routed on edge e
$\Gamma^+(v)$	The set of outgoing edges of endpoint v
$\Gamma^-(v)$	The set of incoming edges of endpoint v
$b^+(v)$	Egress bandwidth requirement of endpoint v
$b^-(v)$	Ingress bandwidth requirement of endpoint v
K	The set of classes supported by the network, $k \in K$
J	The set of links in the network, $j \in J$
C_j	Bandwidth reservation on link j when dimensioning the VPN with the pipe model or the hose model
b_k	Bandwidth requirement of class- k connections
$\lambda_{u,v}(k)$	Arrival rate of class- k connections between endpoint u and v
μ_k	Mean holding time of class- k connections
M	The number of paths between a VPN endpoint pair.
$\mathcal{P}_{u,v}$	The set of alternative paths between endpoints (u,v)
$p_{u,v}(m)$	The m -th path in $\mathcal{P}_{u,v}$
$p_{u,v}$	The single path between endpoint (u,v) in tree routing
$\rho_{u,v}(k)$	Offered load of class- k connections between endpoint (u,v)
$\rho_{u,v}(m,k)$	Offered load of class- k connections to $p_{u,v}(m)$
$\rho_{u,v}(m,k,j)$	Offered load on link j due to the load of class- k connections of $p_{u,v}(m)$, $j \in p_{u,v}(m)$
$\rho(k,j)$	Offered load of class- k connections on link j
$P_{u,v}^a(m,k)$	Probability of a connection of class- k will be admitted on $p_{u,v}(m)$
$P_{u,v}^o(m,k)$	Probability of a connection of class- k will be offered to $p_{u,v}(m)$
$P_{u,v}(k)$	Blocking probability of class- k connections between endpoint u and v
$P_j(k)$	Blocking probability of class- k connections on link j
$q(n)$	The distribution that exactly n units of bandwidth are occupied

Chapter 1 Introduction

Traditionally a company or an organization has to maintain a private network (PN) established using leased lines to provide fast, reliable and secure communications among their geographically dispersed facilities. A PN has obvious advantages over a public network when it comes to performance, reliability and security. However, maintaining a PN with leased lines is very costly and often raises a concern as the facility grows. As a result, in recent years there has been much interest in offering virtual private network (VPN) service.

VPNs are expected to provide a service comparable to that offered by a private network established with leased lines. Proper resource management is essential for this purpose. The newly proposed hose resource management model [1] abstracts the interface between a customer site and the service provider network into a “hose”, which allows VPN customers to specify traffic requirement per VPN endpoint with an aggregate ingress and egress bandwidth parameters. This is in sharp contrast to the conventional pipe model [2], where traffic demands have to be specified per endpoint pair, causing scalability problems.

As the hose model offers customers many advantages over the pipe model, it is desirable that the hose model is adopted in provisioning VPN services. This approach will bring together the cost-efficiency of VPN and the flexibility of the hose model. However, the hose model makes the resource management task more challenging, as it explicitly allows the traffic from a VPN endpoint to be arbitrarily distributed to other endpoints, and allows the traffic distribution to be dynamically changed.

This thesis investigates provisioning VPNs in the hose model. We begin with a thorough and comprehensive literature review on VPNs and the hose model. Next we focus on some important issues in provisioning VPNs in the hose model, including the blocking performance of the pipe model and the hose model, bandwidth-efficiency and blocking performance among different routing schemes for the hose model, availability guaranteed service provisioning and load balancing in the hose model.

The rest of this chapter is organized as follows. Section 1.1 presents an overview on VPN, the pipe model and the hose model, followed by motivations, objectives, methodologies and approaches, contributions, and organizations of the thesis in Section 1.2 to 1.6 respectively.

1.1 Background

1.1.1 VPNs

VPN [3] is a private network established over a shared public network infrastructure to provide cost-effective, quality-assured, and secure communications between a set of geographically dispersed branch offices and facilities of an organization. Figure 1.1 is a VPN example where two VPNs, namely VPN A and VPN B, share the same public network infrastructure. Instead of using dedicated connections such as leased lines, a VPN uses “virtual” connections routed through public network to connect the remote sites, and the management tasks may be outsourced to VPN service providers, therefore the communication cost of VPN customers can be significantly reduced. In addition, MultiProtocol Label Switching (MPLS) [4], Differentiated Service (DiffServ) [5], and Resource ReserVation Protocol (RSVP) [6] technologies have enabled VPNs to provide customers with quality-of-service (QoS) guaranteed

services; IPSec [7] suite protocols have strengthened VPNs to be as secure as traditional PNs.

Figure 1.1 A VPN example

The following terms have specific meanings in the context of VPN.

- User: An individual human using a computer
- Site: A collection of users in a local network that is attached to a CE
- VPN customer: An enterprise or organization controlling multiple sites.
- VPN service provider: An organization offering VPN services.

1.1.2 The Pipe Model and the Hose Model

The pipe model [2] and the hose model [1] are the two eminent resource management models in the context of VPN. They play a vital role in providing QoS guaranteed service.

In the pipe model, customer sites (sub-networks at branch offices, thereafter they are referred to as endpoints) are interconnected through a set of “pipes” (Figure 1.2(a)), which are inter-site tunnels or connections. Resources reserved for a pipe are to be

used exclusively by the two pipe endpoints. Traffic entering a pipe gets certain QoS guarantees, e.g., guaranteed minimum bandwidth. In this model, customers are required to specify the exact traffic demand between each pair of VPN endpoints, and the QoS requirements are conditioned on a pair-wise basis. But as the number of VPN endpoints becomes large, it is very difficult or even impossible for customers to predict the traffic matrix—traffic demands for all the VPN endpoint pairs.

In the hose model, the interface between a customer site and a service provider network is abstracted into a “hose” (Figure 1.2(b)), comprising an aggregate ingress and egress bandwidth parameters to specify the bandwidth requirements of this site. The ingress parameter specifies the maximum allowed incoming traffic from all the other endpoints to this endpoint, while the egress parameter specifies the maximum allowed outgoing traffic from this endpoint to all the other endpoints. Customers only need to specify the traffic demand per endpoint, and the QoS requirements are conditioned on the aggregated traffic.

Figure 1.2 The pipe model and the hose model

Compared to the pipe model, the hose model provides customers with the following advantages [1, 8]:

- **Ease of Specification.** Only a pair of ingress and egress bandwidth requirements per endpoint needs to be specified, instead of an exact traffic matrix.
- **Flexibility.** The hose model can accommodate various traffic matrices, thus allowing traffic from one endpoint to be dynamically and arbitrarily distributed to other endpoints, while the pipe model is restricted with a fixed traffic matrix known in priori.
- **Multiplexing gain.** Due to statistical multiplexing, the hose capacity could be less than the aggregate capacity of a set of pipes.
- **Ease of characterization.** Hose requirements are easier to characterize, because the statistical variability of the individual source-destination pair is smoothed by the aggregation into a hose.

The pipe model is implemented by setting up pair-wise bandwidth-guaranteed tunnels. The hose model could be implemented in a number of routing schemes, including tree routing [8], single-path routing [9] and multi-path routing [10]. With tree routing, all the VPN endpoints are connected via a VPN tree, and all traffic from a source endpoint to a destination endpoint follows the unique path in the tree. With single-path routing, there is also a single path joining every pair of endpoints, but the union of these paths is not necessarily a tree. With multi-path routing, for each pair of distinct endpoints, the traffic from a source endpoint to a destination endpoint is split among multiple paths together with a specification of the fraction of traffic on each path.

Besides sharing the common advantages of the hose model, different routing schemes have their own characteristics.

- The pros of tree routing and single-path routing are that they permit simple routing and restoration. The cons include sensitivity to failures, bigger delay due to longer routing paths for some VPN endpoint pairs, potential congestion problems caused by uneven traffic distribution.
- The advantages of multi-path routing are that it has built-in fault protection mechanism and admits efficient algorithms for optimal multi-path routing with general hose parameters. The weaknesses of multi-path routing stem from the increased maintenance cost and complexity associated with multiple paths.

1.2 Motivations

Provisioning a VPN is to identify a subset of the underlying public network and reserve necessary bandwidth on the physical links to meet the VPN's traffic demand and QoS requirements. However, previous studies on the hose model [8-10] are confined to providing bandwidth guarantees, i.e., provisioning the VPN with the minimum amount of bandwidth capacity, such that all traffic can be admitted so long as the hose capacity constraints are not violated. Though bandwidth guarantee is important in provisioning VPNs in the hose model, many other issues deserve equal considerations.

- Provisioning VPNs in the hose model generally consumes twice as much bandwidth as that in the pipe model [11]. More bandwidth will bring about higher cost, which is undesirable. Thus is this overprovisioning justified by some other benefits from the hose model, e.g., better blocking performance?

- The hose model can be implemented with tree routing [8], single-path routing [9] and multi-path routing [10]. Which routing scheme is superior from the cost (total bandwidth requirement) and blocking performance perspectives?
- Reliability guarantee is an important QoS, which is often described in terms of service availability in the presence of network failures. VPNs without availability guarantee subject to frequent service disruptions, and customers will be unwilling to take such services. Thus, how to provide service availability guarantee in the VPN provisioning?
- Traffic engineering is the aspect of network engineering dealing with performance evaluation and performance optimization. Load balancing is a special case of traffic engineering, aiming at distributing the traffic more evenly in the network to avoid congestion problems and to improve the overall network utilization. Then, how to provide load balancing in the VPN provisioning?

These issues are essential for the successful deployment of VPNs in the hose model. Nevertheless, these issues are not covered in previous research works. As a result, these have motivated the present studies in this thesis.

1.3 Objectives

The overall objective of this research is to study how to provision VPNs in the hose model. Specifically, it covers the following scope:

1. Study the resource management models in the context of VPN, and identify the preferred one by comparing their blocking performance in provisioning bandwidth guaranteed connections.

2. Study different routing schemes for the preferred resource management model, and identify the favorable one by comparing their bandwidth efficiency and blocking performance.
3. Study service availability issues in provisioning a VPN such that the VPN can provide service availability guarantee as well as bandwidth guarantee.
4. Study load balancing issue in VPN service provisioning, such that congestion problems could be alleviated and the network resources could be better utilized.

1.4 Methodologies and Approaches

The following approaches are adopted in the research works presented in this thesis:

1. Study IP/MPLS VPNs and Optical VPNs to familiarize the topics.
2. Review other researchers' works on VPNs, focusing on service models; analyze and compare the advantages and disadvantages of existing studies.
3. Improve the hose model by presenting thorough performance evaluation and incorporating availability-guarantee and load-balancing into it.

The research studies employ both mathematical analysis and discrete-event simulations. Queuing theory is used for performance evaluation, and linear programming for bandwidth resource optimizations. Simulations are carried out to examine the proposed analytical models with 95% confidence interval with width of 0.001. All Linear Programs are solved using state-of-the-art CPLEX [12] optimization package from ILOG, while simulations are run with the OPNET [13] simulator and some self-developed code in C++.

In the study, the network is modeled as a bi-directional graph $G = (V, E)$, where V is the set of network nodes, and E the set of links. In a bi-directional network, every link acts as two unidirectional links. The network links are assumed to have large residual capacity, so that link bandwidth capacity is not a constraint in the calculation. This allows us to concentrate on the specific research problem. However, link capacities could be easily incorporated into the analytical framework by adding corresponding constraints into the linear program.

Traffic is abstracted as client requests for connection, which follow Poisson distributions. Both existing commercial network topology and randomly generated network topologies of different sizes are used in the study. In addition, a large set of randomly generated hose parameters (ingress and egress capacities) are used to remove possible dependencies of the result on a particular set of parameter.

1.5 Major Contributions of the Thesis

The major contributions of this thesis are as follows:

1. The blocking performance of the pipe model and the hose model in provisioning bandwidth-guaranteed connections is studied. Two approaches are presented, including both queuing modeling and discrete-event simulations. The study shows that the hose model has blocking performance better than that of the pipe model by several orders of magnitude under normal network operating loads. Thus the bandwidth overprovisioning associated with the hose model is justified. Combined with the other advantages of the hose model, such as ease of specification, flexibility, and multiplexing gain, the hose model is better than the pipe model as the preferred resource management model for VPN service provisioning.

2. A thorough study on bandwidth efficiency and blocking performance of different routing schemes for the hose model is presented. On a wide range of randomly generated network topologies, and together with a large number of randomly generated hose parameters, the total bandwidth requirement and blocking performance of multi-path routing and tree routing (including single-path routing) are examined and compared. In addition, in order to deal with the overprovisioning problem in the hose model, a new concept of sub-provisioning is proposed. The results show that dimensioning hose model VPNs with multi-path routing causes only marginal bandwidth increase at rare combinations of network topologies and hose parameters. However, multi-path routing requires significant less provisioning to achieve a certain blocking performance. In other words, it can deliver much better blocking performance than tree routing at a given provisioning amount. This study has established multi-path routing as the favorable choice for VPN service provisioning in the hose model.

3. Two multi-path routing heuristics are proposed for the provisioning of availability guaranteed as well as bandwidth guaranteed services in hose model VPNs, namely the bandwidth-centric approach and the availability-centric approach. The two approaches both employ multiple link-disjoint paths; they differ in the way to locate these paths. With multiple link-disjoint paths for each VPN endpoint pair, the proposed heuristics are able to provide a second path for a connection to increase the joint availability. Compared with

tree routing and single-path routing, the proposed heuristics reduce the blocking probabilities in provisioning connection requests with availability requirement by a few orders of magnitude. In addition, the heuristics avoids the too-many-path and overlapping-path problems encountered by other related studies, thus making the provisioning process fast and straightforward.

4. A multi-objective multi-path routing approach is proposed for load balancing at both service provider level and VPN customer level. Service provider level load balancing drastically reduces the bandwidth reservation on the most loaded network link, where congestion problems are most likely to occur. VPN customer level load balancing provides the guarantee to have multiple alternative paths between every VPN endpoint pair, thus allowing the possibility to do load balancing within the VPN. VPN customer level load balancing also helps to strengthen the effect of network level load balancing.

1.6 Organization of the Thesis

The rest of this thesis is organized as follows. Chapter 2 conducts a through literature review on IP/MPLS VPNs, optical VPNs, the hose model, and other related studies. Chapter 3 investigates the blocking performance of the pipe model and the hose model with tree routing. Chapter 4 deals with the bandwidth efficiency and blocking performance of different routing schemes for the hose model. Chapter 5 presents two multi-path routing heuristics for availability as well as bandwidth guaranteed services. Chapter 6 proposes a multi-objective multi-path routing approach for load balancing at both service provider level and VPN customer level. Chapter 7 gives the conclusions and recommendations for future works.

Chapter 2 Literature Review

This chapter presents a comprehensive literature review on IP/MPLS VPN, optical VPN, the hose model and other related publications. As the research on VPN covers a wide area, we adopt the following approach. First we discuss IP/MPLS VPN and optical VPN in Section 2.1 for two purposes: on one hand it summarizes numerous papers on how VPN works; on the other hand it lays the foundation for a better understanding of the research presented in this thesis.

Section 2.2 reviews important research papers by classifying them into different groups. Firstly papers on IP/MPLS VPN and optical VPN provide a general context for the VPN-related study. Next a symposium on the hose model gives specific information on the main topic of this thesis. Finally works pertinent to the specific research problems in later chapters are discussed, including blocking probability calculation, availability guaranteed service, and load balancing.

Section 2.3 gives a formal formulation of the hose model and techniques to calculate link capacities when dimensioning VPNs in the hose model. This is a key section in this chapter. Section 2.4 summarizes this chapter.

2.1 VPN Solutions

This section first presents VPN reference models and architectural constructs of PPVNS, followed by two IP/MPLS VPN examples and an optical VPN example. These examples serve the purpose to illustrate how VPN works without going into technical detail. Keen readers are referred to the corresponding references for more in-depth knowledge.

The term virtual private network is very broad, and often causes confusion among people with different backgrounds. This thesis focuses on provider provisioned VPNs (PPVPNs) [14], where service provider assumes the VPN management responsibilities. PPVPNs have been further classified according to where the VPN specific equipment and functions are located: customer edge (CE)-based VPNs vs. provider edge (PE)-based VPNs. Furthermore PE-based VPNs have been classified according to the service offered: L1 VPNs offer layer-1 services, L2 VPNs offer layer-2 services (Ethernet, ATM, or Frame Relay), and L3 VPNs offer layer-3 services. A comprehensive list of Internet Engineering Task Force (IETF) related drafts on L2 and L3 VPNs can be found in [15]. The introduction in this chapter is primarily on L3 VPNs and L1 optical VPNs.

2.1.1.1 VPN Reference Models

Figure 2.1 (a) and Figure 2.1 (b) are reference models of CE-based VPNs and PE-based VPNs respectively. In CE-based VPNs, CEs assume the specific functions of a VPN; PEs are not aware of the existence of the VPN—VPN traffic appears to them as ordinary network traffic. The situation is vice-versa in PE-based VPNs. The VPN specific equipment and functions reside in PEs; CEs have no knowledge that the private network in which they reside is sharing a public network infrastructure with others.

2.1.1.2 Architectural Constructs of PPVPN

PPVPNs are based on the following architectural constructs [16]. Different VPN solutions have different approaches to implement these constructs.

Figure 2.1 VPN Reference Models

CE: Customer edge devices
 PE: Service provider edge devices
 P: Service provider non-edge devices

- **VPN endpoint creation.** VPN endpoints refers to the entities representing specific VPNs in CEs/PEs, such as routing and forwarding instances in BGP/MPLS VPNs [17], Virtual Routers (VRs) in VPNs using VR solution [18]. In provisioning a VPN, service providers (SPs) need to create VPN endpoints in each of the CEs/PEs that have at least one customer site attached to it. Through the association of customer access connections to VPN endpoints, SPs configure which customer site belongs to which VPN.
- **Membership discovery.** Membership refers to the resources belonging to the same VPN. In provider provisioned CE-based VPNs, membership is the set of customer CEs; and in PE-based VPNs, it is the set of PEs (more specifically the VPN endpoints within the CEs/PEs). Membership discovery refers to the process that members of a VPN learn about one another belonging to the same VPN, and discover related information in order to set up inter-site connectivity. It is expected that members of a VPN may change dynamically,

so an automated membership discovery mechanism is more favorable to the static manual configuration.

- **Tunnel establishment.** VPN tunnels can be implemented in various ways, and can be established using either configuration or signaling. In PE-based VPNs, when a VPN packet is sent through the tunnel from the ingress PE to the egress PE across the SP's backbone, the ingress PE adds an additional encapsulation to the packet. This encapsulation serves two purposes: it guarantees the packet from a specific VPN to be forwarded to the correct egress PE, and indicates to which VPN this packet belongs. The egress PE will then strip the additional encapsulation and handle the data packet in correct VPN context. In CE-based VPNs, the tunnels are established directly between CEs; PEs are not aware of the tunnels.

Through VPN endpoint creation and the use of tunnels, VPN specific contexts are created and maintained across the shared SP's backbone, different VPNs thus can use overlapped private address spaces and transport their data opaquely through the SP's backbone network.

- **Reachability information distribution.** Reachability is the function used by VPN approaches to inform the VPN endpoints and their attached CEs about all the customer sites that exist in this VPN and how to reach them. In PE-based VPNs, the routing functions used to distribute reachability information consists of two parts: first is the routing a CE used to export its address information to the directly attached PE, and to learn from the PE address

information of other sites; second is the routing used by PEs to disseminate individual VPN routing information across the SP's backbone. In CE-based VPNs, reachability information is exchanged among the set of CEs, PEs do not actively participate in VPN reachability distribution.

The establishment of a VPN can be thought as comprising four steps. First is *VPN endpoint creation*, through which VPN specific context is created in the CEs/PEs that have at least one customer site attached to it. Next CEs/PEs learn one another through *membership discovery*, and determine the type of VPN to be set up. Subsequently tunnels are set up among the set of CEs/PEs belonging to the same VPN by invoking certain types of *tunneling* procedures. Finally *reachability information distribution* facilitates CEs to learn the addresses in other sites. By then, a fully functional VPN has been successfully established and traffic can start transmission.

2.1.2 IP/MPLS VPNs

In literature, there are extensive publications [14-21] on general VPN introduction. These works cover VPN concepts, service scenarios, technologies, requirements and standardizations. Two prominent VPN solutions are presented as follows:

2.1.2.1 BGP/MPLS VPNs

In BGP/MPLS VPNs [17], Border Gateway Protocol (BGP) [22] with multi-protocol extension (MP-BGP) [23] is used to disseminate the reachability information across the SP's backbone, and MultiProtocol Label Switching (MPLS) [4] is used to forward packets.

There are five key components in this type of VPN.

- **VPN routing and forwarding (VRF) instance.** VRFs are actually the VPN endpoints in the PEs. For each VPN that has at least one CE attached to a PE, the PE must maintain a VRF instance containing the VPN's route. A PE may contain multiple VRFs to support multiple VPNs, and the CE sites belonging to a VPN are associated to the VRFs corresponding to this VPN through CE-PE connection configurations.
- **VPN-IPv4 address family.** Different VPNs may use overlapped address spaces. To enable BGP to distinguish different VPN route with the same IPv4 address, the VPN-IPv4 address family is introduced. A VPN IPv4 address consists of an 8-byte route distinguisher (RD) and a 4-byte IPv4 address. The RD does not have any specific semantics; it is solely used to distinguish routes from different VPNs. The PEs are configured with a different RD for each VRF or interface connected to the VRF. When stored in the VRF, the RD is prepended to the IPv4 address, so that PEs do not mix routes from different VPNs with the same address.
- **Multiprotocol BGP.** Multiprotocol BGP (MP-BGP) defines extensions to the traditional BGP. It is used to propagate network-layer routing information for different address families. In BGP/MPLS VPNs, a single MP-BGP instance (with extensions to support VPN-IPv4 address family) is used to distribute VPN routes among the set of PEs across the service provider's backbone. Essentially MP-BGP distributes labels instead of raw addresses. These labels are used to identify a specific VPN attached to that PE.

- Route Target (RT).** The RT is a tag or attribute used by PEs to identify VPN routes belonging to a specific VPN. Each VRF is configured to import and export VPN routes with a specific set of RTs. When a PE exports a VPN route to MP-BGP, it attaches its “export RTs” to this route; and when a PE receives a VPN route through MP-BGP advertisement, it will check if the attached RTs match its “import RTs” or not. If yes, the VRF will import and install the route; otherwise the route will be discarded. A RT can be thought of as identifying a set of sites (more precisely a set of VRFs). Associating a particular RT with a route allows that route to be placed in the VRFs that are used for routing traffic which is received from the corresponding sites.
- MPLS.** A two-level label stack is used to transmit VPN packets across the SP’s network. The top level label is associated with the egress PE, used to label switch the packet across service provider network. The bottom level label is associated with a VRF in the PE, used to identify the specific VPN to which the egress PE should forward the packet.

Figure 2.2 BGP/MPLS VPN example

Figure 2.2 illustrates the route distribution process in BGP/MPLS VPNs. CE2 advertises address 10.1.2/24 to PE2. This may be via the use of RIP [24] or OSPF [25] protocol. PE2 then converts this address into a VPN IPv4 address by prepending a route distinguisher of 005 to it. The newly converted IPv4 address is stored in VRF3 with a (export) route target of “red”. Next MP-BGP advertises the address onto the provider backbone, with a RT of “red”, a label of “L2”, and the BGP next-hop address of 102.1.1.2, which is PE2’s loop-back address.

When the address advertisement reaches PE1, PE1 compares the RT values associated with the route to the (import) RT values stored in VRF1 and VRF2. Since VRF1 is configured to import routes with RT value equal to “red”, it imports the route and stores the VPN address, the label, and the BGP next hop. VRF2 will discard the route due to a mismatched RT value. PE1 then advertises the VPN route to CE1 with its own address as the next hop.

2.1.2.2 Virtual Router VPNs

A virtual router (VR) [18] emulates a physical router and its functions in the same way. In the VR approach, the PEs house a virtual router for each VPN it supports. It appears that each VPN has a set of designated routers for its service. VRs belonging to the same VPN exchange routing information, and the per-VPN context is created. VR solution is simple and straightforward. But as a PE must host an instance of VR for each VPN it supports, this solution often raises scalability concerns.

2.1.3 Optical VPNs

The technological advances in optical networks [26, 27], combined with the successful deployment of L2 and L3 VPN services, have motivated the provisioning

of VPN services over optical networks (OVPN). Optical VPN utilizes and enhances common control and management technologies from L2 and L3 VPNs, and applies them to L1 optical transport networks.

Compared to L2 and L3 VPN services, optical VPN works at L1 (the physical layer) and thus can support multiple types of transport technologies over its L1 service, such as IP, ATM, and Ethernet. In addition customers are given the privilege to modify their VPN topologies through control and management functions. These advanced features make optical VPN most attractive to customers who require large capacity as well as advanced control and management features, with a wide range of transport technologies to support. Two possible optical VPN service scenarios envisioned are as follows:

- Multiservice backbone: The service provider may partition its optical network into several optical VPNs, with each one supporting a different higher layer service.
- Carrier's carrier. A carrier relies on another carrier's optical VPN service to provide its own higher layer service.

In order to support optical VPNs, the current optical networks must be first enhanced with intelligent control and management functions, as optical VPN services, like dynamic connection creation, per-VPN management, and user-controlled topology modification, all depend on it. Generalized MultiProtocol Label Switching (GMPLS) [28] has emerged as the control and management plane protocols for the next-generation optical networks. Secondly, hardware-level support, like Optical Add-Drop Multiplexing (OADM) functions, is also critical. In view of a connection

typically requires a sub-wavelength bandwidth, traffic grooming techniques are employed to combine low-speed connections onto lightpaths to make more efficient use of wavelength resource. By combining several low-speed connections with the same source and destination nodes, OADM function allows them to bypass intermediate nodes to minimize the amount of costly electronic multiplexing equipment. This on one side improves the total network throughput, and on the other side reduces the network cost. More detailed study on traffic grooming can be found in [29-32].

General introduction to optical VPNs can be found in [33-38]. Among them, the paper [38] presented a comprehensive overview on optical VPNs, covering requirements, network models, possible future work, and standardization activities.

Just like L2 and L3 VPNs, optical VPN (L1 VPN) also needs to handle reachability (addresses/discovery) and connection setup functions [39]. Reachability information propagation can follow the VPN auto-discovery mechanisms in L3 VPNs, and connection setup is accomplished using GMPLS Resource Reservation Protocol-TE (RSVP-TE) [40] signaling.

- **Addressing.** A CE is identified by one or more TE links that connect the CE to the PE. TE links are control plane abstraction of physical resources. Each CE is assigned a unique identifier within a given VPN, the customer port index (CPI), and a unique identifier within the provider network, the provider port index (PPI). With the CPI-PPI association, the provider network can map a CPI to a PPI, which corresponds to a unique address in the provider network.

Between any CE-PE pair, there is at least one channel allowing IP connectivity. This channel is referred to as an *IP control channel* (IPCC), which is used for the intelligent control and management functions.

- **Auto-discovery mechanism.** Auto-discovery can be implemented using either centralized server-based techniques or distributed control plane-based techniques. Centralized server-based techniques require server(s) to be configured with a list of VPN membership and their corresponding list of CPI-PPI associations; CEs and PEs then access the server(s) to query the membership information in order to set up connections. Distributed control plane techniques piggyback VPN discovery information onto the client and provider-based control plane. For example optical VPN may also use the MP-BGP auto-discovery mechanism discussed in L3 VPNs.
- **Signaling Mechanism.** After a CE obtains the information about the remote CPI belonging to the same VPN, it can initiate connection setup request using GMPLS RSVP-TE signaling. RSVP-TE uses *Path* and *Resv* messages to set up a connection; the former is used in the direction from ingress to egress to send a request, and the latter is used in the reverse direction as a reply.

2.2 Related Works

2.2.1 IP/MPLS VPNs

2.2.1.1 Minimal Cost Design of VPN

Minimal cost design of VPN refers to provisioning a VPN such that the total cost is minimized. Cohen *et al.* [41] investigated the problem of determining the layout of tunnels in PE-based VPNs and CE-based VPNs, considering the cost of both links and routers involved in the tunnel setup. They defined related graph algorithm problems and showed that the problems were both *NP*-hard and hard to approximate, followed by two heuristic algorithms. The heuristics first constructed a CE-based solution, assuming core routers were not allowed to be tunnel endpoints; then the solution was improved by spending some relaxation cost on activating core routers as tunnel endpoints in strategic points. In [42], a network-flow model was proposed for the minimal-cost link allocation for the VPN tunnels. Only the cost of links was considered. Raghunath *et al.* [43] examined the trade-offs in operational efficiency in VPN provisioning among three network mechanisms, including *admission control*, *signaling-based per-link reservations*, and *traffic matrix estimation*. The main result was that traffic matrix estimation was the dominant factor in determining the utilization gain.

2.2.1.2 Quality of Service Issues

VPNs are viewed as a replacement for conventional private networks, so there is inherent requirement on VPN to provide QoS-guaranteed services. Studies in this field usually employ DiffServ, RSVP, and MPLS technologies. From a service provider perspective, Zeng *et al.* [44] examined various QoS enabling technologies and put them together to give a big picture on VPN QoS. Braun *et al.* [45] proposed a

management architecture for QoS enabled VPN, accompanied by a prototype implementation.

Zhang *et al.* [46] studied VPNs with IPv6 traffic allowing multi-class QoS using both queuing modeling and simulations. They modeled each output port of a VPN-enabled router as two priority queues, and multiple loss priorities were implemented by partitioning buffer of the two queues using threshold control mechanism. Garg [47] *et al.* proposed a stochastic fair sharing (SFS) scheme to carry out fair link sharing and fair sharing among Virtual Leased Lines (VLLs, tunnels or LSPs). In the link sharing environment, capacities allocated to different classes were dynamically adjusted as sessions arrived or departed, and the SFS admission control algorithm decided which session to accept or reject according to the current bandwidth utilization and the bandwidth allocation to different classes. In the VLL sharing environment, capacity resizing requests were handled by SFS admission control algorithm in the service provider network. Issacs *et al.* [48] presented prototype architecture for a VPN service that allowed each VPN to have guaranteed resource and customized control, so that QoS for each VPN was guaranteed. This architecture built and managed the VPN on demand, which meant the service creation and modification can be done quickly. Khalil *et al.* [49] proposed a two-layered model to provision VPN-DiffServ networks where the QoS requirements were specified as a range rather than a single value. The top layer was responsible for edge provisioning; it allocated the capacities to various QoS classes, and managed them efficiently to allow resource sharing in a dynamic and fair manner. The lower layer was driven by the top layer and responsible for interior provisioning to meet the QoS offered at edges.

2.2.1.3 VPN Protection and Restoration

Protection and restoration are two commonly used techniques to provide network reliability. They both work by switching interrupted traffic to alternate links or paths when failure occurs. Protection technique is proactive in that it reserves necessary resources early before failure occurs; restoration technique is reactive in that it finds resources after failure has occurred. Hegyi *et al.* [50] studied using shared path protection to enhance the reliability of a set of VPNs, where different VPNs can share protection bandwidth with each other. They presented a simulated allocation scheme to remove the dependency on the order in which the VPNs and their demands were processed. Han *et al.* [51] proposed a framework to support seamless network fail-over, where a set of minimally overlapping paths were pre-calculated and installed, so that they can be used as backup paths when the primary path failed. An application level module performed the fail-over process without re-connection and re-authentication.

2.2.1.4 Tunnels in VPN

Tunnels are an integral part of VPNs. Various tunneling techniques [52] are used to create per-VPN context and to separate traffic from different VPNs. In addition, IPSec [7] tunnels are the way for IP/MPLS VPNs to provide secure services. Yang *et al.* [53] presented an ordered-split approach to manage IPSec/VPN policies, with the objective of minimizing the number of tunnels. The proposed order-split approach was shown to be able to produce globally correct minimum set of policy rules, and performed significantly better in reducing the number of tunnels, compared to two existing off-line algorithms generating IPSec policies at endpoint-pair level. Chen *et al.* [54] proposed a new concept of “authority” to alleviate management overhead of IPSec tunnels by reducing the number of them. Tunnels established by higher

authority gateways can be used to relay packets originated from lower authority gateways and thus avoided the setting up of more tunnels. The ref. in [55] analyzed and evaluated the performance of encrypted tunneling models based on a corporate VPN on Windows and Novell platforms.

2.2.2 Optical VPNs

2.2.2.1 Optical VPN Design and Implementation

Takeda *et al.* [56] proposed network architectures and mechanisms for optical VPNs. They considered both centralized and distributed architectures. Possible mechanisms to implement these architectures consist of *multiple instances* (like the VR solution in L3 VPN), *BGP extension* (like the BGP/MPLS solution in L3 VPN), and *OSPF extension*. In [57] path computation algorithms were presented for dedicated and shared models for resource allocation in optical VPNs. With dedicated model, service provider's network resources were partitioned for different VPNs' exclusive use; with shared model, the whole provider network resource was shared among all VPNs. Simulation results showed that the proposed path computation algorithm achieved better performance with dedicated model.

In optical WDM networks, if data are routed to their respective destinations based on wavelengths, then the network is referred to as wavelength-routed WDM network. A lightpath is an optical trail from the source node to destination node, traversing several wavelength routed nodes. A virtual topology is a set of lightpaths connecting a set of access nodes. Zheng *et al.* [58] presented an integer programming approach for the virtual topology design and reconfiguration of optical VPNs over WDM networks, taking into account the average propagation delay, the maximum link load, and the

reconfiguration cost. As the problem was *NP*-hard, a heuristic called *Balanced Alternate Routing Algorithm (BARA)* based on a genetic algorithm was also presented. Chae *et al.* [59] presented a new method to implement multiple optical VPNs over a passive WDM network using cyclic arrayed waveguide grating (AWG). In [60] a framework was proposed to provide VPNs over optical WDM networks with QoS guarantee. Three kind of lightpaths, *Dedicated lightpath (DLP)*, *Shared lightpath (SLP)* and *Multi-Hop path (MHP)*, were used to accommodate traffic with different QoS requirements. In [61, 62] the problem of designing on-line VPNs over WDM networks was investigated in order to support different QoS. According to quality sensitivity, traffic streams within each VPN were classified into three types, *Constant Bit Rate (CBR)*, *Variable Bit Rate (VBR)*, and *Unspecified Bit Rate (UBR)*. Accordingly, three types of VPNs, *static*, *optical burst switching (OBS)*, and *alternate* VPNs, were allocated to meet the respective QoS requirements of different traffic streams.

2.2.2.2 Survivability in Optical VPNs

Survivability refers to the ability of a network to withstand and recover from failures [63]. Protection and restoration techniques are often employed to provide network survivability. As optical VPNs generally carry large amount of data, so survivability is an important consideration. Al Sayeed *et al.* [64] proposed to use a shortest pair (two link-disjoint paths with the minimum total cost) algorithm to combat the effect of link failures within optical VPNs. The pros of the algorithm were its simplicity and less computation overhead. Haque *et al.* [65] studied using $(M:N)^n$ protection scheme to provide survivability to a set of optical VPNs. Two integer linear programs were proposed with different strategies in partitioning protection groups, aiming at striking a balance between resource efficiency and computation complexity while preserving

the ability to meet the QoS requirements. Modiano *et al.* [66] explored the problem of embedding a logical topology into the underlying WDM network using survivable routing, i.e., the failure of any physical link would not disconnect the logical topology. They formulated the problem as an integer linear program and proposed a heuristic for it. Prommak *et al.* [67] extended [66] to include load distribution feature for optical VPNs. Vijaya Saradhi *et al.* [68] studied the use of different types of primary and backup lightpaths to provide different class of service to different optical VPNs.

2.2.3 The Hose Model

2.2.3.1 Bandwidth Efficiency of the Hose Model

Duffield *et al.* [1] presented two hose model VPN implementation scenarios—provisioned VPNs and resized VPNs. While provisioned VPNs make use of static bandwidth reservation, resized VPNs dynamically change the bandwidth reservation according to traffic measurement and prediction. Through trace-driven simulations, they examined the benefits of hoses on access links and network internal links using both provisioned and resized approaches. With provisioned VPNs, hoses achieved significant capacity savings over a set of statically allocated pipes on access links, but the savings on network internal links was small. With resized VPNs, hoses achieved significant capacity savings on access links as well as network internal links. The capacity savings on access links in both approaches carried a factor of two to three.

Jüttner *et al.* [11] compared bandwidth efficiency of the hose model to the pipe model. On a wide range of networks with different sizes and topologies, they calculated the bandwidth requirements of the hose model with various dimensioning

alternatives, including *provider-pipe* (shortest path routing), *hose-specific state*, *VPN-specific state*, and *tree routing*. These approaches differ in that they consider different amount of hose and VPN specific state information, from none to full state information. The ratio between the bandwidth requirements of the hose model to those of the pipe model was termed “overprovisioning factor”, which was used as an indicator of bandwidth efficiency of the various dimensioning alternatives. The main result was that tree routing was the most bandwidth efficient dimensioning technique for the hose model, carrying an overprovisioning factor around 2 to the pipe model.

Antal *et al.* [69] proposed to divide the network into clusters and using cluster-based traffic description to achieve an equilibrium between management complexity (the pipe model) and overprovisioning (the hose model). The network was divided into clusters of sites; pipe type or hose type traffic characterization can be applied to intra-cluster and inter-cluster traffic. Based on the cluster-based traffic description, a method to compute the necessary link capacities was also presented.

2.2.3.2 Routing Schemes for the Hose Model

Kumar *et al.* [8] presented an efficient polynomial algorithm for optimal tree routing with symmetric ingress and egress bandwidth values (i.e., the hose ingress bandwidth and egress bandwidth are the same) and infinite link capacities, and proved that the optimal tree routing for general ingress and egress bandwidth values, even with infinite link capacities, was already *NP-hard*. For the latter problem, they also gave a 10-approximation heuristic algorithm. Chim *et al.* [70] developed enhancement to the symmetric tree routing algorithm in [8] to include link capacities.

Gupta *et al.* [71] showed that it was *NP*-hard to find any feasible tree routing and single-path routing, even for the case of symmetric hose ingress and egress parameters and finite link capacities. Gupta *et al.* [9] considered the problem of optimal single-path routing for arbitrary hose parameters and infinite link capacities, and presented an algorithm that computed a tree routing with at most 5.55 times the cost of an optimal single-path routing. Italiano *et al.* [72] extended the study by showing that an optimal tree routing can be computed in polynomial time for general hose parameters and infinite link capacities, provided that the sum of hose egress parameters was the same as the sum of hose ingress parameters.

Erlebach *et al.* [10] investigated multi-path routing. They presented a linear programming approach for optimal bandwidth reservation in provisioning hose model VPNs with multi-path routing. The proposed multi-path routing linear program (LP) was shown to be a polynomial optimal algorithm with asymmetric hose parameters and finite link capacities, whereas such cases with tree routing and single-path routing were known to be computationally hard. On a set of randomly generated network topologies, they studied cost savings (bandwidth capacity reduction) and running times of the multi-path routing algorithm to that of tree routing and single-path routing. Their results showed that multi-path routing delivered on average 8.6% bandwidth saving on small network instances with the number of network nodes ranging from 3 to 5, as well as reduced the running time to reach an optimal solution.

Chu *et al.* [73] presented several linear programming approaches to find the optimal routing and to compute the maximum admissible traffic to a regular hose model MPLS network and a restorable network where every path was protected. Given a

network specified by its topology and link capacities, maximum admissible traffic was the maximum hose parameters that can be accommodated.

Liu *et al.* [74] studied two traffic engineering problems for hose model VPN provisioning. They extended the symmetric tree routing algorithm presented in [8], so as to maximize the number of VPNs established on the network backbone. In [75] online hose model VPN provisioning algorithms were proposed for the symmetric tree routing case to handle multiple VPN setup requests rapidly and to reduce the rejection ratio.

2.2.3.3 Protection and Restoration of the Hose Model

Italiano *et al.* [76] presented a restoration algorithm for VPNs in the hose model with tree routing. The assumptions were that the hoses were with symmetric parameters and at any time only one link could fail. Restoration algorithms select a set of backup edges and reserve necessary bandwidth on them. In case of link failure, traffic on the disrupted path is switched to the backup paths. They proved the problem to be *NP*-complete, and proposed a heuristic algorithm at most 16 times of the optimum. The proposed algorithm was based on two subsequent reductions to convert the original problem to that of adding minimum cost edges to the VPN tree so that the resulting graph was 2-connected.

The *Greedy Backup Paths Heuristic* proposed in [74] was shown to require much less protected bandwidth than the algorithm proposed in [76]. Balasubramanian [77] considered the bandwidth requirements for protecting a hose model VPN with link and path protection schemes, showing that path protection had lower bandwidth requirement.

2.2.3.4 Hose Model Extensions

In addition to provide bandwidth guarantee to meet customers' traffic requirement, papers in this section extended the hose model by incorporating other QoS considerations. Zhang *et al.* [78] presented provisioning algorithms for hose model VPNs with delay requirements. Applications using this VPN were classified into several groups; each group was characterized by the maximum delay allowed between every pair of endpoints. Three algorithms were presented, namely *Pipe Mesh*, *Multiple Source-Based Tree*, and *Shared Tree*; all algorithms employed certain delay-constraint routing algorithm to compute the VPN topology.

Wei *et al.* [79] investigated fair bandwidth allocation schemes in hose model VPNs. A fluid hose model VPN was proposed; and based on this model an idealized fluid fair bandwidth allocation scheme was employed to improve the VPN's performance. Simulation results showed that the proposed scheme was able to improve the overall throughput and enable the VPN customers to allocate the bandwidth according to their own requirements.

2.2.4 Blocking Probability Calculation

The blocking performance study employs the reduced-load approximation technique proposed by Kelly [80], which is based on the following assumptions:

- *Link independence assumption*: blockings occur independently from link to link.
- *Poisson rate assumption*: offered traffic to a link is Poisson and reduced by blockings on other links.

These assumptions lead to two mappings: one is from link arrival rates to blocking probabilities, and the other is from blocking probabilities to link arrival rates. These

mappings in turn generate a fixed-point equation, the solution to which is the blocking probabilities and arrival rates in equilibrium state.

The work in [80] was on single-rate loss networks, where the traffic demand of a connection was one circuit. Chung *et al.* [81] extended [80] for multi-rate loss networks, where a connection request can assume different traffic demands.

The above works were only applicable to the case where the total traffic between a source-destination pair was routed on a single path. Greenberg *et al.* [82] presented an approximation technique for the case where the traffic between a source-destination pair can be routed on multiple overlapping paths. Though [82] mainly dealt with reliability issues, the technique can be applied to calculate blocking probabilities as well.

Jüttner *et al.* [11] investigated the blocking performance of the hose model and the pipe model based on simulations on an IP telephony network dimensioned with the pipe model.

2.2.5 Availability Guaranteed Services

Availability refers to the probability that a resource is in correct operating state at any point of observation. Protection techniques, a proactive method that finds a backup path in advance during connection provisioning time, are usually employed to provide desired connection availability. Typically a working path and a link-disjoint backup path are required for a connection to survive from single link failure in the network by switching traffic promptly from the failed working path to the backup path. Protection techniques are commonly used in situations where high availability is mandatory,

such as wavelength-division multiplexing (WDM) networks [83] where a single fiber may carry a large amount of data, thus a single fiber link failure may cause large-scale service disruptions. A comprehensive survey of the protection as well as the reactive restoration schemes can be found in [84, 85] and the references therein.

Seok *et al.* [86] proposed a dynamic heuristic algorithm to find hop-count and path-count constrained multiple paths to satisfy a request's traffic demand, with the objective of minimizing the maximum of link utilization. Traffic split ratios were also obtained for routers to divide traffic between the same source and destination onto multiple paths.

Rai *et al.* [87] proposed to inversely multiplex a large-capacity connection onto multiple paths to satisfy a connection's availability requirement, with each path carrying a fraction of traffic proportional to its availability. Two heuristics were proposed for this problem, namely *Maximum Availability Heuristic* and *Maximum Flow Heuristic*.

Chakrabarti *et al.* [88] investigated constrained routing in QoS networks, where reliability was treated as one of the constraints. A new concept of partial protection was proposed, where backup paths were created only for a selected set of domains in the network to satisfy the reliability constraints. Three two-pass resource reservation schemes were proposed to implement partial protection, namely *Conservative*, *Optimistic*, and *Hybrid* schemes. The three schemes differed in when the backup path was created, i.e., during forward pass, reverse pass, or both.

Huang *et al.* [89] proposed an availability model to capture network failure characteristics and resource information for each link. Based on this model, they made use of a primary-backup path pair to provide required availability in wavelength-division-multiplexing (WDM) networks. They proposed an online algorithm, which temporarily set the availabilities of the links on the primary path to a very small value, so that the subsequent routing computation would favor a link-disjoint path as the backup path.

Tornatore *et al.* [90] presented a design technique for reliable optical transport network, which aimed at maximizing the availability level guarantee and minimizing the total number of fibers installed. The technique comprised two phases. In the first phase, the network was dimensioned for a given set of static protected optical connections on a network with unconstrained number of fibers on each link, with the objective to maximize the availability level of each connection. In the second phase, the number of installed fibers was minimized, while keeping the connection availability the same or within a pre-fixed margin.

2.2.6 Load Balancing

Load balancing is a particular feature of traffic engineering; it aims at distributing the traffic load evenly in the network, so as to eliminate congestion problems. Network congestion will result in large delay and packet loss in packet-oriented networks, or high blocking probability in connection-oriented networks. MPLS with its explicit routing capability has provided some basic mechanisms for handling traffic engineering problems. For example, load balancing can be achieved by distributing the total traffic to several explicitly routed Label Switched Paths (LSPs).

Dinan *et al.* [91] presented an analytical framework to adaptively map ingress traffic onto several parallel LSPs. Different models were employed for expedited forwarding and best-effort traffic to achieve different objectives. Elwalid *et al.* [92] proposed an adaptive approach to balance the load among multiple paths based on measurement and analysis of path congestion. Load balancing was achieved in [91, 92] by applying certain hashing methods on the packet fields: the traffic was first distributed into N bins, where the number of bins determines the minimum amount of traffic that could be shifted among the paths; and then the N bins were mapped onto multiple paths. Hashing based methods generally are more complex, and have possible packet re-ordering problem. Cao *et al.* [93] presented an extensive performance study of several hashing schemes that preserved the order of packets within a flow.

Song *et al.* [94] and Zhao *et al.* [95] studied flow-level load balancing schemes, which have the advantages of efficiency and free from packet re-ordering problem. The former proposed a dynamic Load Distribution to Multipath (LDM) algorithm, aiming at enhancing network utilization and performance (delay). The algorithm works by first setting up multiple Label Switched Paths (LSPs), among which some are chosen to form a candidate LSP set. For each incoming traffic flow, LDM selects an LSP from the candidate LSP set according to a probability distribution that is a function of both the length and the utilization of the LSP. The candidate LSP set could be expanded when its utilization level exceeds a predefined limit. The work [95] modeled each LSP as a M/G/1 processor-sharing (PS) queue, and showed that the difference of traffic (in terms of arrival rate) distributed on any two LSPs should be proportional to the difference of the average packet delays of the corresponding LSPs. Based on the observation, they proposed a heuristic load balancing algorithm using

packet round trip delay in lieu of the LSP delay. The algorithm only needs to be implemented in the ingress/egress LSRs.

While the above works focused on load balancing between a single source-destination node pair, the following researches concentrated on network design and provisioning. These research works assumed the average traffic demands between each node pair were known in priori. Wang *et al.* [96] presented two mathematical formulations for the case of allowing traffic bifurcation (splitting traffic demand between source-destination pair among multiple paths) and the case of disallowing traffic bifurcation. For the later case, they presented some heuristics. The mathematical formulations aimed at minimizing the maximum link utilization so that the congestion was minimized. Seok *et al.* [86] extended [96] by incorporating hop-count and path-count constraints through the use of heuristic algorithms. In addition to maximum hop-count constraint, the work of [97] further considered discrete split ratio and link/node affinity constraints (i.e., whether a node/link can be used for a specific request).

Prommak *et al.* [67] studied load balancing in the context of optical VPNs, where lightpaths (virtual connections) were established through the provider optical network to connect client edge devices. Connections originating or terminating on a node were defined as connections significant to that node. They presented a mathematical formulation aimed at minimizing the number of wavelengths and fiber cables used for the VPN. By introducing a penalty cost on subsequent connections routed on the same fiber as an earlier one did, the formulation also tried to minimize the number of interrupted lightpaths by distributing the connections significant to each node over different fiber links.

2.3 Fundamentals of the Hose Model

2.3.1 Formal Definition of the Hose Model

The network is modeled as a bi-directed graph, where V is the set of network nodes and E the set of edges. In a bi-directional network, $(i, j) \in E$ implies $(j, i) \in E$ as well. The set of VPN endpoints is denoted by Q , $Q \subseteq V$. For each $v \in Q$, a $b^+(v)$ and $b^-(v)$ is given to specify its egress and ingress bandwidth requirements. $b^+(v)$ and $b^-(v)$ are assumed to be non-negative integers.

Let $d_{u,v}$ denote the bandwidth requirement between endpoints u and v . A traffic matrix $D = (d_{u,v})_{u,v \in Q}$ is valid only if for each pair (u, v) of distinct endpoints in Q , there is a corresponding $d_{u,v}$ in D such that for every $v \in Q$, the bandwidth requirement $d_{u,v}$ satisfies

$$\sum_{u \in Q} d_{u,v} \leq b^-(v) \text{ and } \sum_{u \in Q} d_{v,u} \leq b^+(v).$$

2.3.2 Minimum Link Capacity Calculation in the Hose Model

Suppose the routing for an endpoint pair (u, v) is given in the form of a vector $f_{u,v} \in R^E$, where $f_{u,v}^e$ is a parameter, denoting the fraction of traffic between (u, v) that is routed on edge $e \in E$. Its value is assumed to have been given (i.e., known) with range $0 \leq f_{u,v}^e \leq 1$. Thus $x_{\min}(e)$, the minimum amount of bandwidth required on edge e to accommodate all the valid traffic matrices, could be calculated through the following linear program (LP) [10, 11]:

$$\begin{aligned}
 & \max \sum_{u,v \in Q} f_{u,v}^e \cdot d_{u,v} \\
 & s.t. \sum_{v \in Q} d_{u,v} \leq b^+(u), \quad u \in Q \\
 & \quad \sum_{v \in Q} d_{v,u} \leq b^-(u), \quad u \in Q \\
 & \quad d_{u,u} = 0, \quad u \in Q \\
 & \quad d_{u,v} \geq 0, \quad u, v \in Q, u \neq v
 \end{aligned}$$

The constraints of this linear program capture all the valid traffic matrices, so the objective value, maximized $\sum_{u,v \in Q} f_{u,v}^e \cdot d_{u,v}$, is just $x_{\min}(e)$. As the linear program has polynomially many constraints, usually an alternative min-cost flow method is exploited to reduce the computation time [10]. For every edge $e \in E$, a graph (Figure 2.3) $H_e = (N_e, A_e)$ is defined as follows: $N_e = \{s \cup t \cup V_1 \cup V_2\}$, $A_e = \{E_1 \cup E_2 \cup E_3\}$. V_1 and V_2 are two disjoint copies of every node $u \in Q$. E_1 contains the set of directed edges from s to every $u_1 \in V_1$ with capacity $b^+(u_1)$ and cost 0; E_2 contains the set of directed edges from $u_1 \in V_1$ to $v_2 \in V_2$ with infinite capacity and cost $-f_{u,v}^e < 0$; E_3 contains the set of directed edges from $v_2 \in V_2$ to t with capacity $b^-(v_2)$ and cost 0.

Figure 2.3 Min-cost flow formulation

A flow routed on the graph corresponds to the traffic routed on edge e with a valid traffic matrix, and the set of valid traffic matrices correspond to all the possible flows, thus the negative of the optimal objective is just the $x_{\min}(e)$ needed to handle the worst traffic load on edge e . The min-cost flow computation will be used to check if a constraint has been violated in the following multi-path routing linear program.

2.3.3 Minimum Link Capacity Calculation in Tree Routing

The technique presented in Section 2.3.2 for calculating minimum link capacity in the hose model is applicable to tree routing, single-path routing, and multi-path routing. Kumar *et al.* [71] presented a simpler and more efficient approach for calculating the minimum link capacity for tree routing, and this approach is used in this thesis when tree routing is used for the hose model.

The algorithm is briefly recapped as follows. Let T represents the customized VPN tree, and $b^-(v)$ and $b^+(v)$ represent the ingress and egress bandwidth for node v respectively. If link (i, j) were deleted from T , the tree will be divided into two connected components: $T_i^{(i,j)}$ represents the component of tree T containing node i while $T_j^{(i,j)}$ represents the component containing node j . Let $Q_i^{(i,j)}$ denote the VPN endpoints contained in $T_i^{(i,j)}$ and $Q_j^{(i,j)}$ denote the VPN endpoints in $T_j^{(i,j)}$. The bandwidth reservation on link (i, j) in the direction from i to j is given by

$$C_T(i, j) = \min \left\{ \sum_{v \in Q_i^{(i,j)}} b^+(v), \sum_{v \in Q_j^{(i,j)}} b^-(v) \right\} \quad (2.1)$$

This is because the only traffic that traverses link (i, j) from i to j is the traffic originating from endpoints in $Q_i^{(i,j)}$ and directed to endpoints in $Q_j^{(i,j)}$.

2.3.4 Multi-Path Routing Linear Program

The multi-path routing linear program (LP) from [10] will be frequently referred to in this thesis. For completeness purpose, it is presented below. Let c_e be the per unit bandwidth cost, x_e the amount of bandwidth reserved on edge e , C_e the capacity of edge e . $\mathcal{D}$ represents the set of all valid integral traffic matrices. $\Gamma^+(v)$ and $\Gamma^-(v)$ denote the set of outgoing and incoming edges of v , respectively.

$$\min \sum_{e \in E} c_e x_e \quad (2.2a)$$

$$\text{s.t. } x_e \leq C_e, \quad e \in E \quad (2.2b)$$

$$\sum_{u,v \in Q} d_{u,v} f_{u,v}^e \leq x_e, \quad D \in \mathcal{D}, e \in E \quad (2.2c)$$

$$\sum_{e \in \Gamma^+(u)} f_{u,v}^e = 1, \quad u, v \in Q, u \neq v \quad (2.2d)$$

$$\sum_{e \in \Gamma^-(u)} f_{u,v}^e = 0, \quad u, v \in Q, u \neq v \quad (2.2e)$$

$$\sum_{e \in \Gamma^+(v)} f_{u,v}^e = 0, \quad u, v \in Q, u \neq v \quad (2.2f)$$

$$\sum_{e \in \Gamma^-(v)} f_{u,v}^e = 1, \quad u, v \in Q, u \neq v \quad (2.2g)$$

$$\sum_{e \in \Gamma^+(w)} f_{u,v}^e - \sum_{e \in \Gamma^-(w)} f_{u,v}^e = 0, \quad (2.2h)$$

$$u, v \in Q, u \neq v, w \in V \setminus \{u, v\}$$

$$0 \leq f_{u,v}^e \leq 1, \quad u, v \in Q, u \neq v, e \in E \quad (2.2i)$$

$$x_e \geq 0, \quad e \in E \quad (2.2j)$$

The objective function (2.2a) aims to minimize the total cost in dimensioning the hose model VPN with multi-path routing. If c_e is set to 1, then the objective is the minimum bandwidth capacity needed to dimension the hose model VPN.

The constraint (2.2b) mandates that the bandwidth reservation on each edge cannot exceed the edge capacity. The constraint (2.2c) requires that all valid integral traffic matrices $\mathcal{D}$ must be satisfied. As it consists of exponentially many constraints, one for each valid traffic matrix, in implementation, it is incorporated into the LP by a

cutting-plane technique [98]. First the process begins without the constraint of (2.2c) to compute an optimal solution of $f_{u,v}^e$'s and x_e 's. Next, with the newly computed $f_{u,v}^e$'s as input, the min-cost flow method presented in Section 2.3.2 is applied to calculate $x_{\min}(e)$ for every edge $e \in E$. Subsequently $x_{\min}(e)$ is compared against x_e to see if $x_e \geq x_{\min}(e)$ for every edge $e \in E$. If no violation is found, an optimal solution has been obtained. Otherwise, constraint (2.2c) is violated for the traffic matrix D that generates $x_{\min}(e)$ on edge e . This traffic matrix can be obtained from the result of the min-cost flow computation—flow on edge (u_1, v_2) is the traffic demand between endpoint (u_1, v_2) . We then add the violated constraint $\sum_{u_1, v_2 \in Q} d_{u_1, v_2} \cdot f_{u_1, v_2}^e \leq x_e$ into the LP for every edge e that a violated constraint is found. Note that x_e and $f_{u,v}^e$ remain to be variables in the constraint, and the violation for different edge e may be caused by different traffic matrix. With the revised LP, the whole process repeats until an optimal solution is obtained.

Flow conservation constraints (2.2d) to (2.2h) guarantee $f_{u,v}^e$'s represent a flow from u to v . Finally constraint (2.2i) is the bound on $f_{u,v}^e$'s and constraint (2.2j) requires the bandwidth reservation must be non-negative.

2.4 Summary

This chapter presents a comprehensive literature review on VPN, the hose model, and some related topics. Research publications are categorized into groups to give a general context as well as specific knowledge on the research problems of this thesis.

First IP/MPLS VPN and optical VPN solutions are introduced to provide some basic knowledge of how VPN works, paving the way for a better understanding of the discussions that followed. Next, papers on IP/MPLS VPN and optical VPN are discussed and classified according to topics, including minimal cost design, protection and restoration, survivability, etc. Works in this group summarize the research activities in the large context of VPN. Subsequently we concentrate on the specific research topic of this thesis, the hose model. Papers in this group provide a thorough coverage of researches on the hose model, encompassing bandwidth efficiency, routing schemes, protection and restoration, and extensions of the hose model. Finally studies addressing blocking probability calculation, availability guaranteed services, and load balancing are shown respectively to furnish essential background knowledge to the specific research problems of this thesis.

The last part of this chapter consists of a formal formulation of the hose model, and methods to calculate link capacities when dimensioning VPNs in the hose model. This section lays the foundation for use in later chapters. It is presented in great detail.

Chapter 3 Blocking Performance of the Pipe Model and the Hose Model

3.1 Introduction

VPNs are expected to provide services with certain QoS guarantees, among which bandwidth guarantee is of great importance. Though other QoS metrics such as loss and delay can also be incorporated, routing mechanism taking these metrics into account is intrinsically hard and requires information difficult to acquire such as nodal load and delay characteristics. The most practical way of handling delay and losses is to convert such requirements into an effective bandwidth for the connection request [99]. Thus blocking probability of bandwidth guaranteed connections becomes an important QoS metric for VPNs. In addition, a lightpath is a natural bandwidth-guaranteed connection with its bandwidth granularity in terms of wavelengths. Thus the blocking performance study for bandwidth guaranteed connections applies to optical VPNs over WDM networks as well.

Traditionally the hose model is used with IP/MPLS VPNs; this work is the first of its kind to apply the hose model to optical VPNs over WDM networks. In order to concentrate on the relative merits of the hose model and the pipe model, we make following assumptions on optical networks: a connection requests one or more wavelength channels and all the network nodes are assumed to have full wavelength conversion capability.

Performance analysis of traffic grooming [29-32] is out of the scope of this thesis, therefore only a brief discussion on the difficulties and possible solutions is presented.

With traffic grooming, a connection requests sub-wavelength bandwidth, and network nodes do not have wavelength conversion capability. These bring new challenges. First of all, connections with different source-destination nodes traverse the same fiber link cannot share the same wavelength. Secondly, as the total bandwidth in a fiber link is allocated in discrete unit of wavelength, a fiber link cannot be considered as a single link whose capacity is the sum of capacities of all the wavelengths in it. As a result, performance analysis has to cross network domains, considering both IP/MPLS networks and optical networks. Xin *et al.* [100] studied performance analysis of traffic grooming in mesh WDM networks, and developed a theoretical analytical model. Their approach is to convert the two-layer traffic in the inter-domain architecture to the “fictitious” one-layer traffic, and conduct the analysis only in the optical network domain. Interested readers are referred to [100] for more detail.

This Chapter studies and compares the blocking performance of the pipe model and the hose model in provisioning bandwidth-guaranteed connections, employing both analytical models and discrete-event simulations. This is in sharp contrast to similar work [11] in literature which largely depends on simulations.

3.2 Analytical Models

Consider a bi-directional network with a link set J . Let the links in J be numbered through 1 to $|J|$, where link j has C_j units of bandwidth. The network supports K classes of connections, where class- k connection is defined by its bandwidth requirement $b_k \in \{1, 2, \dots\}$, $k \in K$. The path between endpoint (u, v) is denoted by $p_{u,v}$, $p_{u,v} \subseteq J$. If there is enough free bandwidth on the links of $p_{u,v}$, a connection between endpoint (u, v) will be accepted, and the bandwidth is assigned according to the class

for the duration of the connection. Otherwise the connection is blocked and lost. Class- k connections between endpoint (u,v) are assumed to arrive at endpoint u according to independent Poisson processes with rate $\lambda_{u,v}(k)$, and the holding time is of general distribution with mean $1/\mu_k$, thus the offered load is $\rho_{u,v}(k) = \lambda_{u,v}(k)/\mu_k$. The main performance metric considered is the connection request blocking probability.

Let $P_j(k)$ denote the approximate probability that represents “fewer than b_k units of bandwidth are available on link j ”, and hence a class- k connection will be blocked on link j . Suppose that these events are independent from one link to another, thus class- k connections will arrive at link j according to a Poisson process with offered load

$$\rho(k, j) = \sum_{(u,v): j \in p_{u,v}} \rho_{u,v}(k) \prod_{l \in p_{u,v} \setminus j} (1 - P_l(k)). \quad (3.1)$$

Figure 3.1 is an example showing how equation 3.1 works. There are four VPN endpoints labeled with 1, 2, 3, and 4 respectively. For illustration purpose, connections are only allowed between (1,3) and (4,2). Let $P_1(1)$, $P_2(1)$, $P_3(1)$, $P_4(1)$, and $P_5(1)$ denote the probability that a class-1 arrival will be blocked on link 1, 2, 3, 4, and 5 respectively. Consider the offered load of class-1 connections on link 2. As link 2 is shared by two paths where $p_{1,3} = \{\text{link 1, link 2, link 3}\}$, $p_{4,2} = \{\text{link 4, link 2, link 5}\}$, so the offered load on it should be:

$$\rho(1, 2) = \rho_{1,3}(1)(1 - P_1(1))(1 - P_3(1)) + \rho_{4,2}(1)(1 - P_4(1))(1 - P_5(1)).$$

Figure 3.1 Illustrative example

Let $w(n)$ denote the probability that exactly n units of bandwidth are occupied. Under the link independence assumption, we can calculate the probability of a class- k connection being blocked on link j using:

$$P_j(k) = 1 - \frac{\sum_{n=0}^{C_j-b_k} w(n)}{\sum_{n=0}^{C_j} w(n)}, \quad k \in K. \quad (3.2)$$

where $w(n) = \frac{1}{n} \sum_{k \in K} b_k \cdot \rho_j(k) \cdot w(n-b_k)$ and $w(0) = 1$, $n = 1, \dots, C_j$. The computation complexity of $P_j(k)$ is $O(C_j |K|)$, since the main work is to calculate $w(n)$; and for each $w(n)$, it involves a summation over all the possible class connections that utilize this link.

Equations 3.1 and 3.2 define a continuous mapping from a compact convex set $[0,1]^{|K||J|}$ into itself. Thus by the Brouwer fixed-point theorem [101], there exists a solution $(P_j(k), k \in K, j \in J)$ to equation 3.2. Once a solution to equation 3.2 is obtained, the blocking probability of a class- k connection between (u, v) can be calculated by

$$P_{u,v}(k) = 1 - \prod_{j \in P_{u,v}} (1 - P_j(k)). \quad (3.3)$$

This again is based on the assumption that blocking occurs independently on links. Repeated substitution could be used to get a solution to equation 3.2. In most cases, it will converge very fast. Here is a short description of the substitution process:

1. Assign an initial value for all the $P_j(k)$, using equation 3.1 to calculate $P_j(k)$.
2. Substitute those $P_j(k)$ into equation 3.2 and get a new set of $P_j'(k)$.
3. If $|P_j'(k) - P_j(k)|$ has reached the desired precision, then stop; otherwise go to step 2.

It is noted that although the Brouwer fixed-point theorem guarantees the existence of a fixed point, the fixed point might not be unique. It has been pointed out in [102] that in fully connected networks when alternate routes are used without admission control, there can be multiple fixed points. In particular, the network may enter a bi-stable region. This is because the use of the two-hop routes will consume more network resources, thus forcing more connections to take up the two-hop routes. Therefore the network might exhibit bi-stable behavior, where there are two equilibrium points, one stable and the other not stable. However, no adequate solution can be found in current literature to guarantee the convergence to a particular equilibrium point. In addition, Greenberg *et al.* [82] argue that general networks with arbitrary topology may lack the bi-stable behavior, even if alternate routes are used without any admission control. The reason is that the ratio of the number of links in the first path to the number of links in the second path is greater than 0.5, thus the increase in resource consumption of using the second path is not as drastically as that in fully-connected network. As a result, the blocking probability will grow smoothly as the network load increases,

which prevents the network to enter the bi-stable region. In all our experiments in this Thesis, our fixed point algorithm can reach steady state after initial period.

3.2.1 Incorporating Traffic Bounds into the Analytical Model

The pipe model is based on a known traffic matrix $D=(d_{u,v})_{u,v \in Q}$, where $d_{u,v}$ is the traffic demand between endpoints u and v , and Q is the set of VPN endpoints. The hose model associates each endpoint v with an ingress capacity $b^-(v)$ and an egress capacity $b^+(v)$. This thesis resolves the difference by mandating that the amount of traffic injected into (received from) the VPN by an endpoint in both models is the same. The total traffic an endpoint can inject into the VPN in the pipe model is the sum of outgoing traffic demand with this endpoint as source; and the total traffic an endpoint can receive from the network is the sum of incoming traffic demand with this endpoint as destination. The sum of outgoing (incoming) traffic demand for an endpoint in the pipe model is used as the hose egress (ingress) parameters for the corresponding endpoint in the hose model. That is for each $v \in Q$,

$$b^+(v) = \sum_{u \in Q} d_{v,u} \quad \text{and} \quad b^-(v) = \sum_{u \in Q} d_{u,v}.$$

In order to incorporate the traffic bounds ($b^+(v)$ and $b^-(v)$) into the analytical model, the “node-splitting” network transformation technique [103] is used to convert the b^+ and b^- bounds into link capacities. For each VPN endpoint v , an artificial node v' together with two artificial directional links, one from v' to v with capacity $b^+(v)$ and the other from v to v' with capacity $b^-(v)$, are introduced. This transformation is shown in Figure 3.2. The network link connecting v' and v is represented by a solid line; it consists of two directional links (the artificial links) represented by dashed lines with arrow heads. v' becomes the source and destination of a connection; v acts as an

intermediate node that neither contributes nor consumes any traffic. In this way, the traffic an endpoint injects into (receives from) the VPN is guaranteed not to violate the b^+ (b^-) bounds.

Figure 3.2 Example on node-splitting

3.2.2 Applying the Analytical Model to the Pipe Model

In the pipe model, a “pipe” is set up at the source and destination endpoint interfaces. Based on the traffic matrix, a common capacity $d_{u,v}$ is reserved to support the “pipe” between endpoints u and v ; the bandwidth capacity $d_{u,v}$ will be shared by all class connections. If the pipe consists of multiple links, then the same amount $d_{u,v}$ will be reserved on all the links. Figure 3.3(a) shows an example of a “pipe” which consists of two links with capacities of 6 and 7 respectively; 4 units of bandwidth is reserved on each link for the exclusive use of this “pipe”.

Figure 3.3 A pipe example

The above analysis reveals that, after converting the VPN endpoint’s ingress and egress bounds into link capacities, a pipe actually can be modeled as a three-link

system as shown in Figure 3.3 (b). With this transformation, the analytical model presented in Section 3.2 can be applied to study the blocking performance of the pipe model.

3.2.3 Applying the Analytical Model to the Hose Model

Tree routing [8] is used for the hose model. A detailed comparative study has been presented in [11] on the bandwidth requirement of the pipe model and the hose model based on a wide range of network sizes and topologies. The conclusion is that tree routing is the most bandwidth efficient routing scheme, carrying an overprovisioning factor around 2 over the pipe model. In addition, tree routing and the pipe model share the common characteristic of one single path for every endpoint pair. Thus tree routing is a good candidate for the comparative blocking performance study for the hose model. Best-cost shortest-path tree (with the minimum total bandwidth capacity) is used for the hose model. First the shortest-path tree for each network node is calculated, and then the tree structure with the best cost is chosen as the final routing result. When the construction is done, the path between a source and destination endpoint is fixed.

We make use of the method in [8] to calculate bandwidth reservation on tree links. A simple example is given in the following section. With tree routing, the two directional links directed into and out of the leaf nodes will have a bandwidth reservation equal to the leaf nodes' b^+ and b^- bounds. Under such circumstances, traffic from (to) the leaf nodes is already governed by the b^+ (b^-) bounds, so there is no need to introduce the artificial node and links for them.

3.3 Discrete Event Simulations

Discrete event simulations are carried out using OPNET simulator. The simulated physical network topology is the NSFNET (Figure 3.4) where all the nodes are used as VPN endpoints. The network links are assumed to have very large capacities so that link capacity is not a constraint in the study.

Figure 3.4 The NSFNET topology

A traffic matrix for the pipe model is randomly generated and shown in Table 3.1. In the traffic matrix each element represents the traffic demand between a specific source-destination endpoint pair. Note that traffic demand from u to v is not necessarily equal to that from v to u . All elements are uniformly distributed between 1 and 5. The traffic demand could be in any appropriate units. For example, in IP networks, the unit may be in Kbps; whereas in optical networks, the unit may be the number of wavelengths.

For the hose model, the ingress and egress parameters are derived by summing up all the elements in the corresponding column and row. For example, for endpoint 1, its egress parameter is $\sum_{v=2}^{13} d_{1,v} = 37$; and ingress parameter is $\sum_{v=2}^{13} d_{v,1} = 39$. The hose ingress and egress parameters are summarized in Table 3.2.

Table 3.1 Traffic matrix for the pipe model

nodes	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	−	4	5	2	2	4	1	5	2	5	2	1	1	3
2	4	−	2	3	3	5	2	3	2	4	1	2	3	3
3	5	4	−	1	4	3	2	4	3	1	3	4	5	5
4	5	5	3	−	4	4	3	5	3	3	5	5	2	4
5	3	1	3	4	−	3	2	4	5	4	3	2	4	1
6	1	1	2	5	1	−	5	2	5	2	1	5	1	4
7	1	4	4	5	4	3	−	5	1	4	5	3	4	2
8	3	1	4	1	5	3	2	−	2	4	3	5	3	5
9	2	1	5	1	3	4	1	5	−	1	3	3	2	2
10	3	1	1	5	5	4	5	2	1	−	5	3	4	3
11	1	3	5	1	1	3	4	3	5	1	−	4	2	5
12	3	3	1	5	5	2	3	1	2	4	4	−	1	4
13	5	5	4	4	3	1	3	1	3	3	4	5	−	4
14	3	1	1	4	4	5	2	1	2	2	2	3	3	−

Table 3.2 Traffic parameters for the hose model

Node 1		Node 2		Node 3		Node 4		Node 5		Node 6		Node 7	
In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out
39	37	34	37	40	44	41	51	44	39	44	35	35	45
Node 8		Node 9		Node 10		Node 11		Node 12		Node 13		Node 14	
In	Out	In	Out	In	Out	In	Out	In	Out	In	Out	In	Out
41	41	36	33	38	42	41	38	45	38	35	45	45	33

The network is assumed to support four class connections, where class- k connections requires k units of bandwidth, $k \in \{1, 2, 3, 4\}$. A connection request is equally likely to be with any of the four classes. Thus if the total network offered load is ρ , the total network offered load of class- k connections $\rho(k)$ is given by $\rho(k) = \rho/4$. The traffic injected by an endpoint into the network is proportional to the ratio of its egress

parameter to the sum of egress parameters of all VPN endpoints. Let $\rho_u(k)$ denote the offered load of class- k connections of endpoint u , then

$$\rho_u(k) = \rho(k) \times \frac{b^+(u)}{\sum_{m \in Q} b^+(m)},$$

where Q is the set of VPN endpoints, $m \in Q$. Similarly the traffic injected by an endpoint and directed to a specific destination is proportional to the ratio of the destination endpoint's ingress parameter to the sum of ingress parameters of all VPN endpoints (excluding that of the source endpoint). Thus the offered load of class- k connection between endpoint u and v is given by

$$\rho_{u,v}(k) = \rho_u(k) \times \frac{b^-(v)}{\sum_{m \in Q \setminus u} b^-(m)}.$$

Shortest path routing is used to dimension the pipe model. For example, endpoint 2 takes the path 2-3-6 (see Figure 3.4) to reach endpoint 6; and endpoint 6 takes the reverse path to reach endpoint 2. Knowing that the traffic demand between 2 and 6 is 5, so 5 units of bandwidth is reserved on links (2,3) and (3,6).

For the hose model with tree routing, the bandwidth reservation on each link is calculated using the algorithm proposed in [8]. Here is a simple example. Consider the link between endpoint 3 and 6 in the direction from 3 to 6. The only traffic traversing this link is that originating from endpoints 1, 2, and 3 and going to all the other endpoints. The maximum traffic that the endpoints 1, 2, and 3 can send is the sum over their egress parameters, giving $118 = 37 + 37 + 44$. As it is smaller than the maximum traffic the other endpoints could accept (i.e., the sum over their ingress parameters), so 118 units of bandwidth will be reserved on this link. The tree structure

is shown in Figure 3.5, and the hose ingress/egress parameters are also shown beside the endpoints. Note that the bandwidth reservation on the links directing out and into the leaf nodes equals the leaf nodes' egress and ingress parameters respectively.

In the simulation, the traffic arrivals follow a Poisson distribution and the service time is exponentially distributed. The mean service time is set to 20 seconds, and the simulation duration is 100 hours.

Figure 3.5 Best-cost tree for the hose model

3.4 Numerical Results and Discussions

3.4.1 Performance of the Pipe Model

For a typical example, first look at connections between endpoint 1 and 14, where the pipe capacity is 3 units of bandwidth (see Table 3.1). In the pipe model, when the total bandwidth demand exceeds the pipe capacity, the newly arrived connection requests will be blocked. As the pipe capacity is less than the bandwidth requirement of class-4 connections, thus all class-4 connection requests are blocked in this case. Figure 3.6 shows the blocking probabilities from both analytical modeling and discrete-event simulations for the four class connections under consideration. The analytical results are very close to that obtained from discrete-event simulations, with

the maximum deviation around 1%. This implies that the reduced-load multi-rate analytical model could well represent the pipe model.

3.4.2 Performance of the Hose Model

Connections between endpoint 1 and endpoint 6 (see Figure 3.5) are used as a typical example. The results are plotted in Figure 3.7. Because of the scalability problem [104], simulation techniques cannot produce accurate results when the blocking probability is very low. So for the hose model, a relatively high load region (from 90 to 225 Erlangs instead of that from 45 to 180 Erlangs for the pipe model) is used. The results of Figure 3.7 indicate that the analytical results are in general agreement with simulation results. The deviation is around 1% to 2%, so the proposed analytical method is applicable to the hose model as well.

Figure 3.6 Analytical vs simulation results for the pipe model

Figure 3.7 Analytical vs simulation results for the hose model

3.4.3 Performance Comparison between the Two Models

In this section, the blocking probabilities of the pipe model and the hose model are presented together for comparison using analytical results. The goal is to examine how the hose model improves the blocking performance over the pipe model in a multi-rate bandwidth requirement environment. The blocking probabilities of connections between endpoint 1 and endpoint 6 are shown in Figure 3.8. The corresponding pipe capacity is 4.

The results show that the hose model improves the blocking probabilities over the pipe model drastically, especially at low load region. The blocking probabilities for the pipe model are very high throughout the entire load range. When the network load is 45 Erlangs, the blocking probability for class-1 connections is about 5.5%, and that for class-4 connections is 21%; when the network load is 180 Erlangs, the blocking probability for class-1 connections is 30.5%, and that for class-4 connections is 55%.

The hose model improves the blocking probabilities over the pipe model by 4-5 orders of magnitude at low load region, and 2-3 times at high load region.

Figure 3.8 Performance comparison for a single endpoint pair

With the pipe model, the bandwidth reserved for a specific source-destination endpoint pair can only be used by that pair. If the total traffic demand between an endpoint pair has exceeded the corresponding pipe capacity, future connection requests have to be blocked, even when there are free capacities from other “pipes” that could be used by this endpoint pair. With the hose model, the bandwidth reservation on a link could be shared among all possible connections. Even when the traffic demand between a specific source-destination endpoint pair has exceeded the corresponding pipe capacity, newly arrived connections can take advantage of the free capacities from other “pipes”. The flexibility in sharing bandwidth resource allows the hose model to deliver a much better blocking performance than the pipe model. The good performance strongly demonstrates the superiority of the hose model.

3.4.4 Average Network Blocking Probabilities for the Two Models

In this section, the average blocking probabilities of the pipe model and the hose model over all possible source-destination pairs are obtained and presented. The average blocking probabilities are calculated as follows: first the blocking probabilities of connections in each class between all the possible source-destination pairs are calculated, and then the average value are taken over all the pairs.

Figure 3.9 and Figure 3.10 show the average class blocking probabilities from both the analytical results and simulation results for the pipe model and the hose model respectively. By and large, the simulation results and the analytical results agree well. This is particularly true for the pipe model. Figure 3.11 shows the comparison of average class blockings between the pipe model and the hose model with analytical results.

Figure 3.9 Average class blocking probabilities for the pipe model

Figure 3.10 Average class blocking probabilities for the hose model

Compared with Figure 3.8, for the pipe model, the blocking probability for connections in class-2, class-3, and class-4 are even higher, while that of connections in class-1 is lower. This is because there are many low-capacity pipes, which will drive up the blocking probabilities for connections with larger bandwidth demand. On the other hand, connections in class-1 benefit from the free bandwidth left behind by the high blockings of connections with larger bandwidth requirements, and thus have

Figure 3.11 Average class blocking probabilities for the two models

much lower blocking probability. The results again demonstrate the advantage of bandwidth sharing among different pipes in the hose model, especially for endpoint pairs with small-capacity pipes.

The above comparisons (Figure 3.9 through Figure 3.11) show the average blocking behavior on a per class basis. It is of interest to distinguish the average blocking behavior over all classes of the network. In order to make the result more convincing, we consider a new case where the pipe model and the hose model have the same amount of bandwidth capacity. In this study, the hose model consumes about 1.97 (1917/974) times as much bandwidth as the pipe model, so the bandwidth capacity of each pipe is scaled up by 1.97 to give a matching capacity to the hose model.

Figure 3.12 summarizes the results. It shows that the hose model exhibits large performance improvement over the pipe model. At low load region, the blocking probability of the hose model is better than that of the pipe model by about two orders of magnitude; at high load region, the improvement is around 2 times. The hose model still outperforms the pipe model significantly even when the pipe model has matching bandwidth capacity with the hose model. The large gain in blocking performance is attributed to bandwidth sharing capability among different source-destination pairs in the hose model.

We would like to point out that using other routing algorithms or network topologies will not affect the results. As discussed in Section 3.2.2, the pipe model could be modeled as a three-link system with fixed capacities, which are independent of the routing schemes and topologies being used. The hose model dictates that all traffic

shall be accepted so long as the hose capacity constraint is not violated, so the blocking performance of the hose model largely depends on the hose capacities, which are also independent of routing schemes and network topologies.

Figure 3.12 Network blocking probabilities for the two models

3.5 Conclusion

This chapter investigates the blocking performance of the hose model and the pipe model. The results are applicable to both IP networks and future WDM networks. Two approaches are presented. One is analytical modeling based on the multi-rate reduced-load approximation technique. The other is discrete-event simulation. The experimental results from the two approaches are very close, indicating that the multi-rate reduced-load technique is capable of predicting the blocking performance for the hose model and the pipe model.

A comparison of the blocking behavior of the two models shows that the hose model has superior blocking performance. By and large, the blocking probability of the hose model is better than that of the pipe model by a few orders of magnitude under normal

network operating loads, particularly at low load regions. The good performance is attributed to the flexibility in sharing bandwidth in the hose model.

Compared with the pipe model, the hose model possesses a number of advantages. For example, it allows dynamic changes in traffic patterns, thus freeing the customers from a rigid traffic matrix as with the pipe model. Due to statistical multiplexing, a hose can achieve as much as 3.57 times capacity savings compared to the aggregate capacity of a set of customer-pipes [1]. If used in conjunction with certain resizing technique, the multiplexing gain could be more phenomenal. In addition, the hose model makes the traffic specification and characterization easier, as now the traffic specification only need for every endpoint instead of every endpoint pair, and the statistical variability between a source node to all other nodes is smoothed by aggregating the traffic together.

However, the use of the hose model also implies certain complications. First of all, the hose model makes the resource management task more challenging to service providers, for they have to provide strong QoS guarantee (all traffic should be accepted so long as the hose capacities are not violated) based on a very weak specification (only the aggregate incoming and outgoing rates are known for every endpoint). Consequently, the provisioning algorithms for the hose model are much more complex than that for the pipe model. Secondly, the hose model is with intrinsic overprovisioning problem. For example, if the hose model VPN is implemented in tree routing, the overprovisioning to the pipe model is around two times. This amount of overprovisioning might be prohibitive. Thirdly, the hose model raises new problems, like how to capture the aggregate traffic for an endpoint. A loose

specification will result in large overprovisioning, while a too tight specification will be unable to satisfy the Service Level Agreement (SLA). The ref. [1] suggests starting from a simple specification and later refines it until the SLA is met.

On top of the advantages discussed in existing works, we believe the superior blocking performance shown in this chapter, and the methods of using sub-provisioning to alleviate the overprovisioning or using the overprovisioning to provide availability-guaranteed service presented in later chapters, not only have justified the bandwidth overprovisioning, but also made the hose model the preferred resource management model for VPN service provisioning.

Chapter 4 Tree Routing and Multi-Path Routing for the Hose Model

4.1 Introduction

This chapter studies the bandwidth efficiency and blocking performance of different routing schemes for the hose model, and explores the relationship between the two metrics. As tree routing and single-path routing both have the one-endpoint-pair-one-path property, discussions on tree routing often applies to single-path routing as well. So in this chapter we will concentrate on tree routing and multi-path routing.

Although the bandwidth efficiency of the hose model is a hot research topic, little has been done among different routing schemes. The bandwidth efficiency study of [11] did not consider multi-path routing. While Erlebach *et al.* [10] pioneered the work on multi-path routing, but the cost saving part (bandwidth efficiency of multi-path routing) of their work was confined to small networks with network node number ranging from 3 to 5.

In addition, previous bandwidth efficiency studies did not address the overprovisioning issues in the hose model. In order to give the strong guarantee that all traffic should be accepted as long as the b^+ and b^- bounds are not violated, the bandwidth provisioning of the hose model is based on the worst case traffic split, i.e., the traffic from each hose can be directed entirely to just one other endpoint [1]. This is costly, because worst-case traffic split is rare; most of the time the traffic split is not so extreme. This also means to a large extent, the network resources are not used efficiently. To alleviate the overprovisioning problem, Duffield *et al.* [1] proposed a

“resizing” technique. However, dynamic resizing is rather complex and unproven, not as reliable as static provisioning [11].

The above deficiencies have motivated the present study. Firstly, the bandwidth efficiency of tree routing and multi-path routing is studied. On a wide range of randomly generated network topologies and hose parameters, the bandwidth requirements of tree routing and multi-path routing are calculated. The range of network nodes studied varies from 15 to 50, much larger than that used by Erlebach *et al.* [10]. Then the bandwidth efficiencies of tree routing and multi-path routing are compared. This is a key contribution of this work.

Secondly, a new concept of sub-provisioning is proposed to handle the overprovisioning problem, and based on this concept the blocking performance of multi-path routing and tree routing using static reduced provisioning is studied. Instead of designing full provisioning based on b^+ and b^- bounds for the worst case, reduced provisioning (or sub-provisioning) is used to achieve the performance objectives. This is another key contribution of this work.

The blocking performance study employs both analytical modeling and discrete event simulation. Chapter 3 compared the blocking performance of the hose model with tree routing implementation to that of the pipe model, but did not cover different routing schemes for the hose model, such as tree routing and multi-path routing. Although the reduced-load approximation technique was employed in Chapter 3, the analytical model developed was not suitable for the cases where there are multiple overlapping paths between a VPN endpoint pair. A new formulation is needed.

4.2 Bandwidth Efficiency of Tree Routing and Multi-Path Routing

For the bandwidth efficiency study, the respective bandwidth requirements of various routing schemes are computed and then compared. The bandwidth requirements are dependent on network topologies and the set of hose ingress and egress parameters being used. In order to eliminate the dependency on network topologies, a number of randomly generated topologies are used. Moreover, in order to remove the dependency on hose parameters, the bandwidth requirements with a large set of randomly generated hose ingress and egress parameters are calculated with each topology. Since the per-unit bandwidth cost in both algorithms is set to 1, the total cost is essentially the total bandwidth required to dimension the hose model with tree routing and multi-path routing. The two algorithms are both based on a bi-directional network modeled as a graph $G = (V, E)$, where V is the set of nodes and E the set of edges.

4.2.1 Algorithm for Tree Routing

It has been shown in [8] that the optimal tree can only be obtained using polynomial algorithm when the hose parameters are symmetric. In this thesis, the best-cost shortest-path tree is used. For this purpose, the shortest-path tree (with link weight set to 1) for every node in the network is first calculated, and then the one with the best cost is chosen as the final routing result. The method for calculating the bandwidth reservation on each tree link follows the approach in [8]. Since any tree link partitions the VPN tree into two disjoint parts, the bandwidth required on that link is the minimum of:

- The sum of the egress capacity of the endpoints on the “source side” of the link;

- The sum of the ingress capacity of the endpoints on the “destination side” of the link.

4.2.2 Algorithm for Multi-Path Routing

The linear program (LP) for multi-path routing in [10] is adopted and implemented with the CPLEX [12] optimization package. In the LP formulation, F represents the maximum fraction of traffic between a VPN endpoint pair that can be routed on a path (MFTP). All other notations have the same meaning as defined in Section 2.3.4. The LP is given as follows:

$$\min \quad \sum_{e \in E} c_e x_e \quad (4.1a)$$

s.t.

$$\sum_{u,v \in Q} d_{u,v} \cdot f_{u,v}^e \leq x_e, \quad D \in \mathcal{D}, e \in E \quad (4.1b)$$

$$\sum_{e \in \Gamma^+(u)} f_{u,v}^e - \sum_{e \in \Gamma^-(u)} f_{u,v}^e = 1, u, v \in Q, u \neq v \quad (4.1c)$$

$$\sum_{e \in \Gamma^+(v)} f_{u,v}^e - \sum_{e \in \Gamma^-(v)} f_{u,v}^e = -1, u, v \in Q, u \neq v \quad (4.1d)$$

$$\sum_{e \in \Gamma^+(w)} f_{u,v}^e - \sum_{e \in \Gamma^-(w)} f_{u,v}^e = 0, \quad u, v \in Q, u \neq v, w \in V \setminus \{u, v\} \quad (4.1e)$$

$$0 \leq f_{u,v}^e \leq F, \quad u, v \in Q, u \neq v, e \in E \quad (4.1f)$$

$$x_e \geq 0 \quad e \in E \quad (4.1g)$$

The objective and constraints have been explained in Section 2.3.4. Only the constraint (4.1f) requires further explanation. The MFTP constraint F is an important parameter. It sets an upper limit on the maximum fraction of traffic a path can carry by mandating the fraction of traffic routed through each link on this path cannot exceed F , which is a constant less than 1. In order to satisfy the flow conservation constraint, the traffic between a VPN endpoint pair must be split onto multiple paths. In this way the multi-path routing LP guarantees there are multiple paths for each endpoint pair. If some of the paths between an endpoint pair traverse the same link

(i.e., overlapping paths), then the sum of fractions on those paths must not be greater than F .

The running time of this LP varies with the network topologies, the placement of the hose endpoints on the topology, and the hose parameters used. On a Pentium IV PC running Windows XP with 256 MByte memory, it runs from around 1 minute to nearly 4 hours for extreme sparse topologies and dense topologies respectively (please refer to Section 4.4 for descriptions on the network topologies). For the NSFNET topology used in the blocking performance study, the running time is around 10-15 minutes. The largest scenario it can handle is a network with 50 nodes, 250 edges, and 10 VPN endpoints, which requires about 4-hour computation time. When the network size grows even larger, such computing power fails to produce result within a reasonable amount of time.

Note that the link capacity constraint in [10] cannot provide true multiple-path guarantee. Depending on the set of link capacities, some endpoint pairs may still have only one path between them. One extreme case is that when all the links have very large bandwidth capacities, the result from the multi-path routing LP in [10] will most likely turn out to be single-path routing or even tree routing.

The above algorithms give the routing and link bandwidth reservations for tree routing and multi-path routing. Tree routing is simple, as all traffic between each (u,v) pair goes along the one and only one path in the tree. For multi-path routing, the above algorithm gives the result in arc flow form, i.e., a vector $f_{u,v} \in R^E$ for each (u,v) pair, with $f_{u,v}^e$ specifying the fraction of traffic from u to v traversing the edge

e. Another equivalent representation is the path flow form, i.e., a set $\mathcal{P}_{u,v}$ for each (u,v) pair specifying the set of alternative paths between the pair together with the fraction of traffic on each path. We have implemented a flow decomposition algorithm [103] to convert the arc flows to path flows. The latter will be used in calculating the blocking probabilities in the next section.

4.2.3 Sub-Provisioning and Admission Control

The primary objectives of this chapter are to study bandwidth efficiency with different routing schemes, and show that multi-path routing can provide more efficient bandwidth provisioning with regard to blocking probability as compared with tree routing.

To be consistent with the hose model, the hose parameters, b^+ and b^- values, should be observed. Adequate bandwidth can be reserved based on b^+ and b^- values so that all traffic matrices that do not violate the ingress and egress bounds can be admitted. This provisioning gives strong guarantee, as the bandwidth reserved is based on the worst case traffic split, i.e., the traffic from each hose can be directed entirely to just one other endpoint [1]. Thus, bandwidth provisioning based on b^+ and b^- values gives the largest amount of bandwidth to be reserved for the network. Under this circumstance, the b^+ and b^- admission control dominates. As shall be shown later, the blocking probability of multi-path routing will be the same as that for tree routing.

Though bandwidth provisioning based on b^+ and b^- values caters for the worst case traffic, the amount of bandwidth needed is at least twice (for tree routing) as much as that required for the conventional pipe model [11]. This is costly because worst case traffic split is not frequent; most of the time the traffic split is around the average

value. This means that to a large extent, the network resources are not used efficiently, which is undesirable.

Thus a new concept of sub-provisioning is proposed to handle the overprovisioning problem. Instead of designing full provisioning based on b^+ and b^- bounds for the worst case, reduced provisioning (or sub-provisioning) is used to achieve the performance objectives. The sub-provisioning is defined as providing the reserved bandwidth that is of a certain percentage of the full provisioning. The full provisioning is obtained by reserving the bandwidth according to b^+ and b^- bounds. For instance, 50% provisioning means half of the full provisioning. In this case, the provisioning would be similar to that used by the pipe model. In general, the provisioning could be designed at a certain percentage of the full provisioning depending on the need.

So, in the design, the hose model is first dimensioned based on the b^+ and b^- bounds, and then the link bandwidth reservation is reduced to a certain percentage of the full provisioning value. Two admission control schemes have been designed. The first one is the b^+ and b^- bounds and the second the reserved link bandwidth. Traffic input will be first constrained by b^+ and b^- values. Those admitted by b^+ and b^- values will be further constrained by link bandwidth availability. In the second scenario, the bandwidth constraint will be the key differentiating factor. Thus, multi-path routing is able to show much better blocking performance than tree routing. To achieve a certain blocking probability at a fixed load and F value, multi-path routing requires much less bandwidth provisioning as compared with tree routing.

On the other hand, sub-provisioning will cause some overheads in routing and admission control. Firstly, sub-provisioning usually works together with multi-path routing and some alternate routing mechanisms, so more path information has to be maintained, such as the number of paths and the available bandwidth on each path. Secondly, unlike full provisioning where the admission control only needs to consider the hose capacities, using sub-provisioning must also take the path bandwidth availability into account, rendering the admission process more complex.

Nonetheless, we believe the advantages of sub-provisioning outweigh its disadvantages. Using sub-provisioning not only has preserved the advantages of the hose model, such as flexibility, multiplexing gain, but also effectively alleviates the overprovisioning problem associated with the hose model. In addition, when used in combination with alternate routing, sub-provisioning is able to deliver very good blocking performance.

4.3 Blocking Performance for Tree Routing and Multi-Path Routing

4.3.1 General Assumptions

Consider a bi-directional network with a link set J , where link j has C_j units of bandwidth. A total of K classes of connections are supported, where class- k connection is defined by its bandwidth requirement, $b_k \in \{1, 2, \dots\}$.

Connection requests of class- k from source u to destination v are assumed to arrive at endpoint u according to a Poisson process with rate $\lambda_{u,v}(k)$, and the connection

holding time of class- k connections has an exponential distribution with mean $1/\mu_k$, so the offered load of class- k connections is given by $\rho_{u,v}(k) = \lambda_{u,v}(k)/\mu_k$.

With sub-provisioning, whether a connection can be admitted or not depends on not only the b^+ and b^- bounds but also the link bandwidth reservation. A connection request of class- k is admitted to the network if there are at least b_k units of bandwidth available in the egress and ingress capacities of the source and destination endpoints as well as on each link on its path. A connection request admitted by b^+ and b^- bounds could still be blocked due to insufficient bandwidth reservation on network links.

The routing policy employed is fixed-alternate routing, where each endpoint pair is assigned an ordered list $\mathcal{P}_{u,v}$ of M paths: $p_{u,v}(1), p_{u,v}(2), \dots, p_{u,v}(M)$. Note that different endpoint pairs generally will have different number of paths, depending on the outcome of the multi-path routing algorithm. A connection request is admitted on the first path where it is admissible as long as the b^+ and b^- bounds are not violated. If none of the paths can accommodate the connection request, it is blocked and lost. The main performance metric is the average blocking probability for each class. First the blocking probabilities for each endpoint pair and each class are calculated, and then the average value is taken for each class over all the endpoint pairs.

The alternative paths between a VPN endpoint pair are obtained by solving the LP equations in Section 4.2.2; the fractions associated with the paths provide some guidance to route a connection request. These alternative paths are sorted in descending order according to the fractions associated with them. When a connection

arrives, these alternative paths are attempted in the order from the one with the largest fraction to that with the smallest fraction. The rationale is that a path with larger fraction is more likely to have adequate resources to admit the connection.

The use of fixed-alternate routing instead of assigning traffic to paths according to the pre-calculated fractions is to make good use of the network resources in sub-provisioning. In fixed-alternate routing, a connection is allowed to explore available bandwidth more thoroughly in the network, so its chances of being admitted are enhanced. In this way, multi-path routing is able to reduce the blocking probability significantly when compared to tree routing. If the connection is routed according to the pre-calculated fractions, multi-path routing and tree routing will have about the same blocking performance. This means the potential of multi-path routing to improve the network utilization is not fully utilized.

Multi-path routing with different F values is examined. With each distinct value of F in the LP formulation, a new set of link reservations and a new set of paths for each VPN endpoint pair will be computed. And based on the new link reservations and the new set of paths, the blocking probabilities for the specific F value are re-calculated.

In fixed-alternate routing, a connection of class- k between endpoint u and v will be routed on $p_{u,v}(m)$ when and only when $p_{u,v}(m)$ is admissible while the first $m-1$ paths are in blocking conditions. In other words, if we let $S_{u,v}(m, k)$ and $\bar{S}_{u,v}(m, k)$ denote the events that $p_{u,v}(m)$ are in admissible and blocking states for connections

of class- k respectively, then the probability that a connection of class- k will be *admitted* on $p_{u,v}(m)$ is given by

$$P_{u,v}^a(m, k) = \Pr \left\{ S_{u,v}(m, k) \prod_{i=1}^{m-1} \bar{S}_{u,v}(i, k) \right\}.$$

Here the product indicates the intersection of events. And by conditioning on $p_{u,v}(m)$ to be admissible

$$\begin{aligned} P_{u,v}^a(m, k) &= \Pr \left\{ S_{u,v}(m, k) \prod_{i=1}^{m-1} \bar{S}_{u,v}(i, k) \right\} \\ &= \Pr \left\{ \prod_{i=1}^{m-1} \bar{S}_{u,v}(i, k) \mid S_{u,v}(m, k) \right\} \Pr \{ S_{u,v}(m, k) \}, \end{aligned}$$

we will get the probability that a connection of class- k will be *offered* to $p_{u,v}(m)$

$$P_{u,v}^o(m, k) = \Pr \left\{ \prod_{i=1}^{m-1} \bar{S}_{u,v}(i, k) \mid S_{u,v}(m, k) \right\}.$$

Thus the total traffic load offered to $p_{u,v}(m)$ is given by

$$\rho_{u,v}(m, k) = \rho_{u,v}(k) \cdot P_{u,v}^o(m, k).$$

By the link independence assumption, the offered load due to connections of class- k on $p_{u,v}(m)$ to link $j \in p_{u,v}(m)$ is just

$$\rho_{u,v}(m, k, j) = \rho_{u,v}(m, k) \cdot \prod_{l \in p_{u,v}(m) \setminus j} (1 - P_l(k)),$$

where $P_l(k)$ is the probability that a connection of class- k will be blocked on link l .

Hence the total offered load on link j by all the paths and all the (u, v) pairs is

$$\rho(k, j) = \sum_{u,v} \sum_m \rho_{u,v}(m, k, j). \quad (4.2)$$

With the offered load of class- k connections on link j , the probability that a connection of class- k will be blocked on it is exactly given by (see [105, 106]):

$$P_j(k) = 1 - \frac{\sum_{n=0}^{C_j-b_k} w(n)}{\sum_{n=0}^{C_j} w(n)}, \quad k \in K, \quad (4.3)$$

where $w(0) = 1$ and $w(n)$ is given recursively by

$$w(n) = \frac{1}{n} \sum_{k \in K} b_k \cdot \rho(k, j) \cdot w(n - b_k), \quad n = 1, \dots, C_j.$$

Equations (4.2) and (4.3) define a continuous mapping from a compact convex set into itself. Therefore by the Brouwer fixed-point theorem [101], there exists a solution $(P_j(k), k \in K, j \in J)$ to (4.3). Repeated substitution could be used to get this solution.

Since a connection request is blocked only when none of the paths can accommodate it, so the probability a connection of class- k between endpoint u and v is blocked is given by

$$\begin{aligned} P_{u,v}(k) &= 1 - \sum_{m=1}^M P_{u,v}^a(m, k) \\ &= 1 - \sum_{m=1}^M P_{u,v}^o(m, k) \cdot \Pr\{S_{u,v}(m, k)\} \\ &= 1 - \sum_{m=1}^M P_{u,v}^o(m, k) \cdot \prod_{j \in p_{u,v}(m)} (1 - P_j(k)). \end{aligned} \quad (4.4)$$

The above blocking probability is derived mainly for multi-path routing. However, the analytical formulation can also be used for tree routing. This is because tree routing is a special case of multi-path routing where there is only one path in the list, and the topology happens to be a tree structure.

4.3.2 Approximation in Calculating $P_{u,v}^o(m, k)$

The set of alternative paths produced by the multi-path routing algorithm may be overlapped on each other, which will make the state of a path depend on those preceding it in the list $\mathcal{P}_{u,v}$. This is the reason why conditional probabilities have to be used in calculating $P_{u,v}^o(m, k)$. Unfortunately, for general networks, calculating $P_{u,v}^o(m, k)$ and $P_{u,v}(k)$ is at least *NP*-hard [82]. Thus one more approximation proposed by Greenberg *et al.* [82] is adopted. The approximation has been shown to produce fairly good results:

Conditional Path Independence: the events $S_{u,v}(i, k)$, $i = 1, 2, \dots, m-1$ are statistically independent, where $S_{u,v}(i, k)$ is the event that $p_{u,v}(i)$ is admissible given that $p_{u,v}(m)$ is admissible.

With this approximation, $P_{u,v}^o(m, k)$ could be re-written as

$$P_{u,v}^o(m, k) = \prod_{i=1}^{m-1} \left(1 - \prod_{j \in p_{u,v}(i) - p_{u,v}(m)} (1 - P_j(k)) \right). \quad (4.5)$$

Now the whole process could be summarized as follows:

1. Assign an initial value for all the $P_j(k)$.
2. Compute $P_{u,v}^o(m, k)$, the probability that $p_{u,v}(m)$ is admissible given all the preceding $m-1$ paths are in blocking condition, using (4.5).
3. Calculate $\rho(k, j)$ using (4.2), and then substitute them into (4.3) to compute a new set of $P_j'(k)$.
4. If $|P_j'(k) - P_j(k)|$ has reached the desired precision, then stop; otherwise go to step 2.

In order to incorporate the hose ingress and egress capacities into the analytical model, the node-splitting technique is used to covert hose ingress and egress capacities into link capacities. A detailed explanation on how to apply this technique has been given in Section 3.2.

4.4 Experiments and Numerical Results

The bandwidth efficiency study is carried out on a variety of network topologies with the number of network nodes ranging from 15 to 50 with a step length of 5. For each case, three sets of network topologies are randomly generated using the Waxman model [107]. The three sets of topologies are referred to as extreme sparse, sparse, and dense topologies henceforth. Each set has 8 topologies. The link/node ratio is 2 for the set of extreme sparse network topologies, 3 for the set of sparse network topologies, and 5 for the set of dense network topologies. There are 10 VPN endpoints, randomly placed at the network nodes. The 15-node sparse network topology and the 50-node dense network topology are shown in Figure 4.1 and Figure 4.2 respectively. In these figures, the normal network nodes are depicted as gray dots while the VPN endpoints are depicted as black dots.

Two sets of hose ingress and egress parameters are randomly generated. In the first set, the parameters are uniformly distributed between 1 and 60; and in the second set, the parameters are uniformly distributed between 20 and 60. The first set is referred to as more-asymmetric parameters, and the second set less-asymmetric parameters. For both tree routing and multi-path routing, the bandwidth capacity needed to dimension the hose model are calculated on each topology with 100 sets of different more-asymmetric parameters and less-asymmetric parameters respectively.

Figure 4.1 Sparse network topology with 15 nodes

Figure 4.2 Dense network topology with 50 nodes

For the blocking performance study, the 14-node NSFNET (please refer to Figure 3.4) is used as the physical topology. All nodes are VPN endpoints and the hose ingress and egress parameters are shown in Table 4.1. These parameters give the maximum bandwidth requirement of a hose, and can be set in any appropriate units. For example, in IP networks, the unit could be in x Kbps; and in optical networks, the units could be in y Mbps or a number of wavelengths. First the hose model is dimensioned with tree routing and multi-path routing to get the set of link bandwidth reservations and the routing results, and then the routing results are converted from arc flow form to path flow form with the flow decomposition algorithm. Next the technique developed in Section 4.3 is used to calculate the blocking probabilities. The

set of paths for a specific (u, v) pair produced by the multi-path routing algorithm are first sorted in descending order according to the fractions associated with them.

Table 4.1 Hose ingress and egress parameters

N1		N2		N3		N4		N5		N6		N7	
in	out	in	out	in	out	in	out	in	out	in	out	in	out
34	56	20	60	46	60	56	32	40	30	24	58	24	46
N8		N9		N10		N11		N12		N13		N14	
in	out	in	out	in	out	in	out	in	out	in	out	in	out
48	24	28	32	44	48	52	22	34	26	50	60	56	26

The network supports two class connections with bandwidth requirement of 1 and 2 units respectively. A connection request is equally likely to be with any of the two classes. Thus if the total network offered load is ρ , the total network offered load of class- k connections $\rho(k)$ is given by $\rho(k) = \rho/2$. The traffic injected by an endpoint into the network is proportional to the ratio of its egress parameter to the sum of egress parameters of all VPN endpoints. Similarly the traffic injected by an endpoint and directed to a specific destination is proportional to the ratio of the destination endpoint's ingress parameter to the sum of ingress parameters of all VPN endpoints (excluding that of the source endpoint). Detailed mathematical formulation is given in Section 3.3 of Chapter 3.

OPNET simulator is used for the discrete-event simulations. Traffic arrivals follow independent Poisson distributions and the holding time is exponentially distributed. The mean holding time is 20 seconds, and the simulation duration is set to 50 hours. When the total network load is 15 Erlangs, there are about 2.7×10^6 connection

requests; and when the total network load is 225 Erlangs, there are about 4.05×10^7 connection requests. Traffic load varies with the connection arrival rate.

4.4.1 Bandwidth Requirements of Two Routing Schemes when $F=1$

Our study reveals that the bandwidth requirement of multi-path routing is always less than or equal to that of tree routing, but for most of the cases they have the same bandwidth requirements. Let BW_T and BW_M denote the bandwidth requirement of tree routing and multi-path routing respectively. The bandwidth saving (%) is defined as follows:

$$\text{bandwidth saving}(\%) = \frac{BW_T - BW_M}{BW_T}.$$

Among all the cases evaluated, the maximum bandwidth saving achieved is 3.2%. The number of cases with bandwidth savings is summarized in Table 4.2, and the average savings of these cases are shown in parenthesis.

Table 4.2. Bandwidth savings of multi-path routing

	Less-asymmetric parameters	More-asymmetric parameters
Sparse topologies	4(0.2%)	37(0.93%)
Dense topologies	2(0.75%)	10(2%)

The result is not surprising as tree routing is a special case of multi-path routing with more stringent constraints. Thus, dimensioning hose model with tree routing will certainly require more bandwidth. On the other hand, if the entire traffic between a VPN endpoint pair could be routed on a single path ($F=1$), multi-path routing will try to find the single optimal path for each VPN endpoint pair to minimize the total bandwidth requirement. Under such circumstances, multi-path routing most often will

turn out to be single path routing or even tree routing, though the topology might not be the same. This explains why the two routing schemes have the same bandwidth requirements for a majority of the cases studied.

When the network topology is sparse, tree routing is more likely to find longer paths. Thus, most of the significant bandwidth saving cases are with this type of topologies. When the network is dense, tree routing will have a higher chance to find a better tree structure, and at the meantime multi-path routing always tries to find a tree structure in order to minimize the total bandwidth required, as a tree structure uses the least number of links. Thus the resulting topologies of the two tend to converge, and the cases with bandwidth savings are reduced. Experiment is also done on a set of extreme sparse topologies, where the link/node ratio is 2, implying the topology is nearly to be a tree. With the extreme sparse topologies, multi-path routing is left with no choice but to find the same tree structure as tree routing. Thus for all the cases studied, multi-path routing is unable to reduce the bandwidth requirements.

Close examination on the hose parameters where multi-path routing has bandwidth savings reveals that the difference between the sum of ingress parameters and the sum of egress parameters ($\sum_{v \in Q} b^+(v) - \sum_{v \in Q} b^-(v)$) is quite large. Obviously the more asymmetric the hose parameters, the larger the difference is likely to be. So, most of the bandwidth-saving cases are with the more-asymmetric hose parameters. The individual topology also plays a part. For the sparse topologies, respectively 11 and 17 out of the 37 cases with bandwidth savings are from the 25-node and the 40-node topologies. For the dense topology, 8 out of the 10 cases are with the 35-node topology. Thus without any restriction on the maximum fraction of traffic on a path

(MFTP), multi-path routing only delivers small bandwidth savings at certain rare combinations of topologies and hose parameters.

Tree routing and multi-path routing both use less bandwidth when the network is dense in terms of a higher link/node ratio. With dense networks, the two routing schemes are more likely to find better paths and therefore reduce the total bandwidth required. With the same network density given by a link/node ratio, generally the more network nodes, the more resources are required for both routing schemes. The reason is that as there are more network nodes, the paths between a VPN endpoint pair tends to be longer in terms of hops, thus more links will be involved and more bandwidth is needed. The average bandwidth requirements of multi-path routing and tree routing with the less-asymmetric hose parameters versus the number of network nodes are plotted in Figure 4.3. The overlapped curves indicate the average bandwidth requirements of multi-path routing with sparse (dense) network topologies are very close to that of tree routing with sparse (dense) network topologies. The result shows that individual network topology also plays a large part in the dimensioning cost.

Figure 4.3 Average bandwidth requirements

The fluctuations in Figure 4.3 need more explanation. The properties of the hose model VPNs largely depend on the network topologies, the placement of VPN endpoints on the topology, and the hose parameters used. For a set of VPN endpoints with given hose parameters, if the VPN endpoints are placed further to each other on a network topology, then the total bandwidth requirement may well be greater than the case where the VPN endpoints are placed nearer to each other on a larger network topology (with more nodes but the same link/node ratio). As the network topologies are randomly generated and the VPN endpoints are randomly placed on the topology, the distance between VPN endpoints could vary drastically in different cases; as a result, the total bandwidth requirement fluctuates instead of increasing smoothly as a function of the network size, though the general trend is the larger the network size the greater the total bandwidth requirement. The bandwidth requirement of the tree-dense curve peaks at the 35-node topology simply because the VPN endpoints are placed far apart in this case. The same phenomenon repeats on Figure 4.5 and Figure 4.6.

For all the topologies studied, the bandwidth overprovisioning factors of the multi-path routing relative to the conventional pipe model are also calculated. Adopting the method used in [11], first a traffic matrix $D=(d_{u,v})_{u,v \in Q}$ for the pipe model is randomly generated, where $d_{u,v}$ is the bandwidth demand between VPN endpoints u and v , and Q is the set of VPN endpoints. The corresponding hose egress and ingress parameters are calculated with $b^+(v) = \sum_{u \in Q \setminus v} d_{v,u}$ and $b^-(v) = \sum_{u \in Q \setminus v} d_{u,v}$ respectively. The hose parameters obtained in this way are the minimum ones to accommodate the traffic matrix D . Let BW_P represent the bandwidth requirement of the pipe model, the overprovisioning factor is defined as:

$$\text{overprovisioning factor} = \frac{BW_M}{BW_P}.$$

The result gives the average value over 100 sets of traffic matrices. It shows that multi-path routing also carries an overprovisioning factor around 2 as indicated in Figure 4.4, even though this factor varies with different topologies. Thus the overprovisioning factor of multi-path is similar to that of tree routing [11].

The results in this section reveal that the bandwidth requirement of tree routing is very close to multi-path routing, which actually is the minimum bandwidth requirement to dimension the hose model with all routing schemes.

Figure 4.4 Overprovisioning factor vs number of nodes

4.4.2 Bandwidth Requirement of Multi-Path Routing When Varying F

Without restricting the F value, multi-path routing often turns out to be single path routing. The advanced features of multiple paths, such as load balancing and fault tolerance, are lost with single path. By imposing F to be less than 1 (constraint (4.1f))

in the multi-path LP formulation), a path is restricted not to carry all the traffic between a VPN endpoint pair. In order to satisfy the flow conservation constraint, the entire aggregate traffic must be split onto multiple paths. Note that the F value only sets an upper limit on the fraction of traffic on each path; it does not mandate that the maximum fraction of traffic on a path should be F , nor does it mandate the number of paths should be minimum. The traffic between a VPN endpoint pair may be split onto more paths with each one carrying a smaller fraction of traffic. A very small value of F will result in too many paths between a VPN endpoint pair, (e.g., more than 10). Larger number of paths will incur higher cost in storing and processing them, and cause much complexity in maintaining them, which is undesirable. Hence, in this study we only vary F values from 0.5 to 1 with a step size of 0.1.

When F takes a value less than 1, some traffic between a VPN endpoint pair has to take less optimal paths than the case with $F=1$, thus the total bandwidth requirement will increase. For each distinctive F value, the bandwidth increase is calculated as follows:

$$\text{bandwidth increase} = \frac{BW_M(F \neq 1) - BW_M(F = 1)}{BW_M(F = 1)}.$$

The results with the set of less-asymmetric parameters are plotted in Figure 4.5 and Figure 4.6 respectively. The bandwidth increase varies greatly with individual topologies. For some topologies the alternative paths between a VPN endpoint pair may have to take significantly longer routes, thus the amount of bandwidth consumed will correspondingly increase.

Figure 4.5 Bandwidth increase with sparse topologies

Figure 4.6 Bandwidth increase with dense topologies

The average bandwidth increase with 50 sets of less-asymmetric hose parameters are shown in Figure 4.7. Generally the bandwidth increase is higher with sparse topologies, as the chances of finding a “good” path with sparse topologies are not as high as that with dense topologies. The average bandwidth increase is small when F is near unity, and it is larger when F is reduced to 0.5, where the increase is about 18% for sparse topologies and 10% for dense topologies.

Figure 4.7 Average bandwidth increase

For the sets of sparse and dense network topologies, the average overprovisioning factors of multi-path routing with different F values are presented in Figure 4.8. The trend is similar to the average bandwidth increase in Figure 4.7. When F decreases from 1 to 0.5, the overprovisioning factor correspondingly increases. The sparse topologies require larger overprovisioning than the dense topologies.

Figure 4.8 Overprovisioning factor vs F

4.4.3 Blocking Performance of Multi-Path Routing and Tree Routing

With full provisioning in the hose model, one reserves enough bandwidth based on the b^+ and b^- bounds so that all traffic not violating the b^+ and b^- bounds will be admitted. In this case, whether the hose model is implemented in tree routing or multi-path routing, the same b^+ and b^- bounds apply, consequently, yielding the same blocking performance. This is true for different F values in multi-path routing. Figure 4.9 shows a typical example for the blocking probabilities of tree routing and multi-path with $F=0.9$.

Figure 4.9 Blocking probabilities with full provisioning, $F=0.9$.

With sub-provisioning, the bandwidth capacity of tree routing and multi-path routing will be reduced to a certain percentage of full-provisioning. In this case, multi-path routing exhibits much better blocking performance than tree routing; and the smaller the sub-provisioning, the more significant the improvement will be. The results for 90% as well as 50% of full provisioning with $F=0.9$ for multi-path routing are shown in Figure 4.10 and Figure 4.11 respectively. When F takes a value less than 1, the total bandwidth requirement of multi-path routing will increase. In order to give a fair

Figure 4.10 Blocking probabilities with 90% provisioning

Figure 4.11 Blocking probabilities with 50% provisioning

comparison between the two routing schemes, the link capacity of tree routing (with sub-provisioning) is scaled up accordingly, so that tree routing and multi-path routing have the same amount of bandwidth capacity. For example, with $F=0.9$, the total bandwidth requirement of multi-path routing is increased by 2.2%; thus the bandwidth capacity of tree routing on each link is also increased by 2.2%. If the amount of

bandwidth capacity is not an integer, it is rounded up to the nearest whole number. Note that for full provisioning case, the blocking probabilities are dominated entirely by b^+ and b^- values, hence, scaling up capacity in tree routing is useless.

The results of Figure 4.10 and Figure 4.11 clearly show multi-path routing greatly improves the blocking performance over tree routing. The improvement is more phenomenal when the provisioning is relatively low, e.g., 50%. At low load region the blocking probabilities of multi-path routing are lower than tree routing by several orders of magnitude (for class-1 connections, multi-path routing gives 4.2×10^{-5} and tree routing gives 6.8×10^{-3} at 90 Erlangs). As the load increases, the network resources are depleted, thus the blocking performances of the two routing schemes draw closer. With sub-provisioning, the link bandwidth availability becomes the dominating factor in admitting a connection. Because of the one-endpoint-pair-one-path property, tree routing tends to concentrate traffic together and leads to possible congestion. This is why it has higher blocking probabilities. On the other hand, multi-path routing enables a connection to explore available bandwidth more thoroughly in the network, thus increases its chances of being admitted, so multi-path routing has much lower blocking probabilities.

More experiments are conducted to examine the provisioning percentage needed to support a fixed blocking probability. The results for class-1 connections with blocking probabilities of 0.01 and 0.001 respectively are shown in Figure 4.12. The F value is 0.9 for multi-path routing. Tree routing is made to have a matching bandwidth capacity with multi-path routing.

Figure 4.12 Load vs provisioning percentage for fixed blocking probability

The results in Figure 4.12 show that for a given provisioning percentage, multi-path routing is able to support more traffic than tree routing. In other words, given a fixed load, multi-path routing requires less provisioning than tree routing to support a certain blocking probability. For example, to support a blocking probability requirement of 0.01 with 140 Erlangs traffic, multi-path routing needs about 45% of full provisioning, whereas tree routing needs about 65%. The results for other classes are similar.

The effects of F on the blocking performance of multi-path routing are examined with F varying from 1 to 0.5 with a step length of 0.1. F values smaller than 0.5 are excluded. When F takes a small value less than 0.5, there will be too many paths, with some of them carrying a very small fraction of traffic (e.g., less than 1%). A path with a very small fraction can be interpreted as having a small chance to carry a request, but the path has to be maintained. In addition, with small F value, more traffic will have to take less optimal paths, so the total bandwidth requirement will increase. Tree

routing and multi-path routing have matching capacities; the F values for multi-path routing and the corresponding scaling factors for tree routing are shown in Table 4.3. The blocking probabilities for different F values at 150 Erlangs are presented in Figure 4.13.

Table 4.3. F for multi-path routing vs scaling factor for tree routing

F for multi-path routing	0.5	0.6	0.7	0.8	0.9	1
Scaling factor for tree routing	1.182	1.099	1.066	1.044	1.022	1

Figure 4.13 Blocking probabilities vs F

Figure 4.13 shows that multi-path routing gives a drastic decrease in blocking probabilities whereas the blocking probabilities of tree routing remain high despite the scaling up of the bandwidth. The drastic decrease of blocking probabilities in multi-path routing is attributed to two factors: the increase of bandwidth capacity and the increase of average number of paths (Figure 4.14) between a VPN endpoint pair. The increase of bandwidth capacity means more available bandwidth in the network. The

increase of average number of paths means more chances to find an admissible path. Thus, both contribute towards the lowering of the blocking probabilities.

The variation of F values with the maximum number of paths as well as the average number of paths for an endpoint pair is given in Figure 4.14. The large increase from $F=1$ to $F=0.9$ may seem surprising. The objective of the multi-path routing LP is to minimize the total bandwidth requirement, not the number of paths. Thus, as long as splitting traffic onto more paths can result in a better objective value, the LP will return the result. This is analogous to certain situations in tree routing: in order to minimize the total bandwidth requirement, some endpoint pairs may have to take longer paths rather than every endpoint pair takes the shortest path. The F value does not guarantee the fraction of traffic on a path must be F , nor does it guarantee the least number of paths will be used. For example, consider the number of paths between endpoint 1 and 6. With $F=0.9$, it does not mean there will be two paths, one carrying 0.9 traffic and the other 0.1. In fact there are four paths, with each of them carrying a

Figure 4.14 Number of paths between a VPN endpoint pair

fraction of 7.5%, 5%, 83.7%, and 3.8% respectively. The average number of paths depends on the topology, the hose parameters being used and the F value.

Finally the blocking probabilities of multi-path routing and tree routing using both analytical modeling and discrete-event simulations are compared. This is shown in Figure 4.15. The results are for 50% of full provisioning with $F=0.9$ for multi-path routing. Tree routing is made to have matching capacity with multi-path routing. By and large, the results from the two approaches agree well. For tree routing, the difference for all cases is less than 1%. As for multi-path routing, some discrepancies between analytical and simulation results are found. This is attributed to the complex overlapping patterns among the set of paths produced by the multi-path routing algorithm, so that it is difficult to accurately model the offered load on each alternative path and give very accurate approximation. However, for most of the cases, the difference from the two approaches lies within 2%.

Figure 4.15 Analytical vs simulation blocking probabilities

4.4.4 Constraining the Number of Paths

As mentioned in Section 4.4.2 and Section 4.4.3, the MFTP constraint is able to bring better blocking performance. However, it also raises the problem of potential too many paths between a VPN endpoint pair, as it only sets an upper limit on the fraction of traffic on a path, not mandating the least number of paths should be used. A large number (4 to 7) of paths between each VPN endpoint pair will cause scalability problem when the size of the network becomes large. Therefore, it is desirable to suppress the excessive paths.

For this purpose, we designed a path aggregation algorithm, which is based on the observation that the too many problem often arises from over-splitting, i.e., some paths carry very small fractions of traffic. Paths with very small fractions of traffic can also be interpreted as they have rare chances to carry some traffic, but they have to be maintained. On the other hand, they are good candidates to be aggregated. Aggregating a path p_1 to a path p_2 is to add the fraction on p_1 to p_2 and delete p_1 . Thereafter we will refer to path p_1 as candidate path and path p_2 as destination path.

The path aggregation will render the results from the multi-path routing LP no longer optimal, resulting in an increase in the total bandwidth capacity. Therefore the objective of the path aggregation algorithm is not to increase the total bandwidth capacity much while reducing the average number of paths between a node pair. To achieve this objective, we propose two criteria. First the candidate paths are those with small fractions; second, a candidate path is aggregated onto a destination path that has the most overlapping links with it; if such paths are more than one, the shortest one is chosen. In this way, the path aggregation algorithm will not disturb the

optimality of the multi-path routing LP result much, thus not to increase the total bandwidth capacity much.

The actual path aggregation algorithm is presented in Figure 4.16. Its basic working mechanism is to aggregate a candidate path with a fraction less than a threshold value (0.1 in this study) to a destination path that has the most overlapping links with it. The aggregation process will produce a new routing result in the path flow form, which is then converted back to the arc flow form, i.e., a new set of $f_{u,v}^e$'s. The new set of $f_{u,v}^e$'s are then fed to the same min-cost flow algorithm in Section 2.3.2 to calculate a new set of link bandwidth reservation x_e 's.

Regarding the effects of the path aggregation algorithm on blocking probability, on one hand, the path aggregation process reduces the number of paths, which tends to increase the blocking probability; on the other hand, it increases the total bandwidth capacity, which tends to reduce the blocking probability. Note that the bandwidth capacity increase depends on how much disturbance the path aggregation algorithm does to the routing result from multi-path routing LP; it is not necessarily the greater the F value, the less the bandwidth capacity increase. The maximum and average number of paths before and after aggregation for different F values are shown in Figure 4.17; the bandwidth capacity increase is shown in Table 4.4; and the blocking probabilities for 50% provisioning with $F=0.9$ before and after aggregation are shown in Figure 4.18. For this case, the increase in bandwidth clearly outweighs the reduction of average number of paths. In practice, parameters in the path aggregation algorithm (like the threshold value) and strategies in selecting destination paths can be varied to cater for different needs.

```

input: THRESHOLD
for each VPN endpoint pair {
step 1: sort the set of paths in descending order according to the fractions of
        traffic associated with them
step 2: partition the paths with fractions greater than the THRESHOLD value
        into setMajor and the remaining paths into setMinor
step 3: for each candidate path in setMinor {
    step 3.1: find a destination path in setMajor that has the most overlapping
              links with the current candidate path; if such paths are more than
              one, choose the shortest one. Aggregate the current candidate path
    step 3.2: if no suitable destination path is found in step 3.1, i.e., the current
              candidate path is link-disjoint to all the paths in setMajor, find a
              destination path in setMinor that has the most overlapping links
              with the current candidate path; if such paths are more than one,
              choose the shortest one. Aggregate the current candidate path. If
              the fraction of the result path is greater than the THRESHOLD,
              move the result path into setMajor; otherwise move the result path
              to the end of setMinor
    step 3.3: if no suitable destination path is found in step 3.2, i.e., the current
              candidate path also is link-disjoint to all the paths in setMinor, find
              a destination path in setMajor that has the smallest fraction of traffic.
              Aggregate current candidate path
        }
    }
step 4: convert the results from path flow form back to arc flow form, and
        compute a new set of link bandwidth reservation

```

Figure 4.16 Path aggregation algorithm to remove excessive paths

Figure 4.17 Maximum and average number of paths

Table 4.4 Bandwidth increase after path aggregation

MFTP	0.5	0.6	0.7	0.8	0.9
Bandwidth increase (%)	22	22	13	14	24

Figure 4.18 blocking probability before and after aggregation

4.5 Conclusions

This chapter studies the bandwidth efficiency and blocking performance of multi-path routing and tree routing for VPN bandwidth provisioning in the hose model. For the case without the MFTP constraint ($F=1$), multi-path routing often turns out to be single-path routing or even tree routing, thus the two routing schemes have almost the same bandwidth requirement. For multi-path routing, bandwidth savings occur only with rare combinations of network topologies and hose parameters, and the savings are usually small.

In order to alleviate the overprovisioning problem of the hose model, a new concept of sub-provisioning is proposed. Based on this concept, the study on blocking performance of multi-path routing and tree routing using static reduced provisioning is presented. The results show that with full provisioning, multi-path routing and tree routing have the same blocking performance. However, with sub-provisioning, multi-path routing is capable of delivering an excellent blocking performance, much better than tree routing. The improvement is even phenomenal when the sub-provisioning is relatively low, where the blocking probability of multi-path routing is reduced by a few orders of magnitude at low load region. With sub-provisioning, link bandwidth availability becomes the dominating factor in admitting a connection. The MFTP constraint enforces the employment of multiple paths when it takes a less-than-1 value. Consequently, it allows a connection request to explore available bandwidth more thoroughly, thus increasing its chances of being admitted.

Experiments with multiple sub-provisioning cases show that for a fixed load, multi-path routing requires less bandwidth provisioning than tree routing in order to support

a certain blocking probability. A comparison of the blocking probabilities of multi-path routing and tree routing from both analytical modeling and discrete-event simulation shows a close agreement between the two approaches. It indicates that the multi-rate reduced-load approximation technique is able to model multi-path routing as well as tree routing adequately.

In order to constrain the number of paths between a VPN endpoint pair, a new path aggregation algorithm is introduced, which is shown to be able to limit the number of paths between each VPN endpoint pair within a manageable range.

Multi-path routing has a number of attractive properties, such as polynomial optimal algorithms for general hose parameters with finite link capacities, fault tolerance, etc. In addition, this chapter has demonstrated its superior blocking performance to tree routing. Consequently, it should serve as an attractive scheme to be adopted for the provisioning of VPN service in the hose model.

Chapter 5 Providing Availability Guarantee in Hose Model VPNs

5.1 Introduction

VPN services are usually provisioned with certain quality of service (QoS) guarantees, among which two are of particular importance. The first is resource guarantee, mandating adequate resources (bandwidth) to meet the customers' traffic requirements; and the other is service reliability guarantee, mandating service availability even in the presence of network failures. Service reliability is usually measured in terms of connection availability, which is the probability that a connection is found in correct operating state at any point of time. In practice VPN customers are likely to have availability requirement as well as bandwidth requirement in the service level agreement (SLA) with service providers.

In literature, little research work has been done to address availability issues in hose model VPNs with multi-path routing. Erlebach *et al.* [10] studied hose model VPNs with multi-path routing but without availability consideration. The works in [86-88] studied either multi-path routing or availability guarantee, but none of them was done in the context of VPN. Also, the two heuristics proposed in [87] suffered from the problem that a connection might be split onto too many paths, and the algorithms in [86] and [87] often produced overlapping paths, which were undesirable in provisioning availability guaranteed service. Furthermore, the online primary-backup path based algorithm proposed by Huang *et al.* [89] tends to make the provisioning of a connection a lengthy process. Finally, the work in [90] assumed a set of static

connections that were known in priori, which made it resemble the pipe model and thus was not applicable to the hose model.

This chapter investigates the problem of provisioning availability guaranteed services in hose model VPNs. Two multi-path routing heuristics are proposed for this purpose: the availability-centric approach and the bandwidth-centric approach. The two heuristics both employ multiple link-disjoint paths for each VPN endpoint pair, so that a pair of primary and backup paths can be used to provide the required availability; they differ in the way to find the link-disjoint paths. Each of the two heuristic approaches has its merits and drawbacks, and can be used in different situations accordingly.

5.2 Preliminaries of Availability Guaranteed Service

5.2.1 Joint Availability Calculation

The availability of a resource (a link or a node) is defined as [108]:

$$A = MTTF / (MTTR + MTTF),$$

where *MTTF* is *mean time to failure* (average time between failures) and *MTTR* is *mean time to repair* (the average time taken to fix a problem that occurs on the network). The availability of an end-to-end path from a source node to a destination node is defined as the product of the availabilities of the constituent components along the path, assuming the failures of the components are independent. For simplicity, only links are considered. But this causes no loss of generality, as node cost can always be converted to link cost through standard network transformation [103] (i.e., using the availability as the cost metric). Formally let p be a path and link $l \in p$. Denote the availability of link l as A_l , thus the availability of a path p is given by

$$A(p) = \prod_{l \in p} A_l .$$

Given two link-disjoint paths, namely p_1 (primary path) and p_2 (backup path), if they together are needed to provide the required availability, then the joint availability of these two paths is given by

$$\begin{aligned} A(p_1 p_2) &= 1 - (1 - A(p_1)) * (1 - A(p_2)) \\ &= A(p_1) + (1 - A(p_1)) * A(p_2) \end{aligned} \quad (5.1)$$

This technique can be easily extended to calculate the joint availability of n ($n > 2$) link-disjoint paths,

$$A(p_1 p_2 \dots p_n) = 1 - \prod_{i=1}^n (1 - A(p_i)) .$$

The case for overlapping paths is more complex. Consider the example in Figure 5.1 where the primary path p_1 and the backup path p_2 are overlapped. p_1 consists of links 1-2-3-4-7 and p_2 consists of links 1-2-5-6-7; the non-overlapping section of p_1 is denoted by N_p and that of p_2 is N_b .

Figure 5.1 Overlapping path example

The joint availability= (the availability of overlapping section)×(the availability of primary path non-overlapping section and backup path non-overlapping section together). That is

$$\begin{aligned}
 A(p_1 p_2) &= \prod_{l \in p_1 - N_p} A_l * \left(\prod_{l \in N_p} A_l + \left(1 - \prod_{l \in N_p} A_l \right) * \prod_{l \in N_b} A_l \right) \\
 &= A(p_1) + \left(1 - \prod_{l \in N_p} A_l \right) * A(p_2)
 \end{aligned} \tag{5.2}$$

Suppose p_1 and p_2 in both the link-disjoint case and overlapping case have the same number of links, and all the corresponding links have the same availabilities. Compared with (5.1), as $\left(1 - \prod_{l \in N_p} A_l \right) < 1 - A(p_1)$, it shows that two overlapping paths can only provide smaller joint availability than their link-disjoint counterparts. Thus, overlapping paths are always less effective in raising the joint availability.

Since overlapping paths are less effective in raising the joint availability, more paths are likely needed to provide the required availability. Nonetheless, the overwhelmingly many overlapping patterns make the joint availability calculation for a set of overlapping paths a *NP*-hard problem [82]. Greenbeg *et al.* [82] proposed an approximation that can be used to obtain a fairly good result. Let $X(p_i)$, $i=1,2,\dots,n-1$, denote the event that path p_i is admissible (in normal operating state) given that path p_n is admissible. The *conditional path independence* approximation assumes that the set of events $X(p_i)$, $i=1,2,\dots,n-1$, are statistically independent. Under this approximation, the probability $P(n)$ that the n -th path is available while the preceding $n-1$ paths are out of service is given by

$$P(n) = \prod_{i=1}^{n-1} \left(1 - \prod_{k \in p_i - p_n} A_k \right).$$

With this approximation, the joint availability of a set of n overlapping paths is given by

$$A(p_1 p_2 \dots p_n) = \sum_{i=1}^n P(i) * A(p_i).$$

Each term in the summation can be viewed as the contribution of the corresponding path to the joint availability. The complex calculation is likely to make the connection provisioning a lengthy process, which is unacceptable for connections requiring fast set-up.

5.2.2 Preference of Link-Disjoint paths

Tree routing and single-path routing both have the one-endpoint-pair-one-path property, which makes them inadequate for availability guaranteed service provisioning in the hose model. A single path can only provide a fixed and limited availability, which is unable to meet the stringent availability requirements of some connections assuming critical tasks. On the other hand, hose model VPNs with multi-path routing are able to overcome the drawback with tree routing and single-path routing. When a primary path fails to meet the connection's availability requirement, a backup path can be provided to increase the joint availability. Thus multiple alternative paths for each VPN endpoint pair are essential for availability guaranteed services provisioning in the hose model.

As discussed in the Section 5.2.1, the multiple alternative paths should be preferably link-disjoint. Firstly, link-disjoint paths are more effective in increasing the joint availability compared with overlapping paths, thus only a handful of link-disjoint paths are required. A small number of paths are more manageable and require less memory and computing power to store and process them, hence incurs lower cost. In addition, a small number of link-disjoint paths also eliminate the too-many-path problem faced by other studies. Secondly, link-disjoint paths make the provisioning

process fast and straightforward. Overlapping paths will render the provisioning a lengthy and cumbersome process, as now the state of a path is dependent on that of others, which makes the joint availability calculation very complex. Thirdly overlapping paths run astray from employing multiple paths for higher availability, since a link failure in the overlapping section will still result in large scale service disruption.

5.3 Two Multi-Path Routing Heuristics

5.3.1 Problem Formulation

The network is modeled as a bi-directed graph $G=(V,E)$, where V is the set of network nodes and E the set of edges. In a bi-directional network, $(i, j) \in E$ implies $(j, i) \in E$ as well. The set of VPN endpoints is denoted by Q , $Q \subseteq V$. For each $v \in Q$, the aggregate ingress and egress parameters are given to specify v 's traffic demand. A traffic matrix is represented by $D = (d_{u,v})_{u,v \in Q}$ where $d_{u,v}$ is the traffic demand between u and v ; D is valid only when for each $v \in Q$, the sum of incoming (outgoing) traffic of v is no greater than v 's ingress (egress) parameter.

The problem of availability guaranteed service provisioning in the hose model can be stated as follows: Given a physical network G , a set of VPN endpoints Q and their respective hose parameters, provisioning the VPN such that it satisfies:

- The VPN has enough bandwidth to meet the hose requirements.
- The VPN is able to provide certain availability guarantee.

For the rest of this section, two heuristics are presented for provisioning availability-guaranteed services in hose model VPNs: the availability-centric approach and the

bandwidth-centric approach. The two heuristics both make use of multiple link-disjoint paths between a VPN endpoint pair, and each of these paths is associated with a fraction of traffic between this pair. If a path is assigned a fraction of 0.7, then 70% of all traffic between this endpoint pair will be routed on this path. The fractions are used to calculate the minimum bandwidth capacity needed to dimension the hose model; they are not used when selecting path(s) for a connection request. Different routing schemes have different total bandwidth capacity requirements. If the traffic between a VPN endpoint pair is routed according to the fractions of traffic assigned to the paths, the total bandwidth capacity calculated by these two heuristics will be the minimum provided the hose ingress and egress capacities are observed. This dimensioning is based on the worst-case traffic split. But quite often, the traffic split is not so extreme, so the bandwidth capacity for the VPN is overprovisioned to a certain extent. This overprovisioning could be used to provide availability guaranteed service in normal situations.

The two approaches differ in the way of locating link-disjoint paths. The availability-centric approach first finds alternative link-disjoint paths with higher path availability, and then calculates the bandwidth needed to satisfy the hose ingress and egress capacities. The second approach aims at minimizing the total bandwidth capacity needed in dimensioning the VPN. It begins with a multi-path routing LP with the MFTP constraint (to be explained later), which enables link-disjoint paths for each VPN endpoint pair to be found.

The two heuristics have the following advantages. Firstly, the heuristics only need to maintain a handful of link-disjoint paths, so they are free from the too-many-path

problem and the associated complexities encountered by other studies. Secondly, they make the provisioning of availability guaranteed connections fast and straightforward, as the joint availability calculation is faster for link-disjoint paths than overlapping paths.

We exclude the use of two link-disjoint trees for availability-guaranteed services in the hose model, for it will make the overprovisioning problem with the hose model even worse. Unlike the bandwidth-centric and the availability-centric approaches, link-disjoint trees do not share any links, so they cannot take any advantages of hose and VPN specific state to reduce the total bandwidth capacity. This means two link-disjoint trees will consume more bandwidth than even the availability-centric approach. This amount of overprovisioning is costly, albeit it is a trade-off for higher availability. In addition, it also deviates from our objective: providing availability-guaranteed services by wisely employing the intrinsic overprovisioning with the hose model. Furthermore, two link-disjoint trees could be seen as a special case of multi-path routing, where there are two link-disjoint paths between each VPN endpoint pair. So the results from the bandwidth-centric approach and the availability-centric approach will provide good insight in using the link-disjoint trees.

5.3.2 Availability-Centric Approach

The availability-centric approach is very simple and straightforward. The objective of this approach is to maximize the availabilities of the multiple link-disjoint paths for each VPN endpoint pair. It first finds several link-disjoint paths with highest availabilities for each VPN endpoint pair, and then calculates the bandwidth capacity required to satisfy the hose ingress and egress capacities. For example, if it is decided

that three paths are needed for a VPN endpoint pair, then the paths with the highest, second-highest, and third-highest availabilities are picked for this endpoint pair.

In this study this is evaluated by repeatedly running the Dijkstra shortest path algorithm. As the cost function of Dijkstra algorithm is additive whereas that of the path availability is multiplicative, the link cost uses the negative of the logarithmic value of the link availability. After finding the path with the highest availability, the availabilities of the links on this path are set to a very small value, so that the next run of the Dijkstra algorithm will result in a link-disjoint path. This process is repeated until the desired number of paths is found.

The complexity of the availability-centric approach is governed by that of the Dijkstra algorithm, which is $O(|V|^2)$; V is the set of network nodes. After finding the link-disjoint paths and assigning fraction of traffic to each path, the min-cost flow technique in [10] (and Section 2.3.2) is used to calculate the bandwidth reservation needed to satisfy the hose capacities.

5.3.3 Bandwidth-Centric Approach

This approach consists of several steps.

- a) Multi-path routing linear program with the MFTP constraint

The bandwidth-centric approach begins with the following multi-path routing LP.

$$\begin{aligned}
 \min \quad & \sum_{e \in E} c_e x_e \\
 \text{s.t.} \quad & \sum_{u,v \in Q} d_{u,v} \cdot f_{u,v}^e \leq x_e, \quad D \in \mathcal{D}, e \in E \\
 & \sum_{e \in \Gamma^+(u)} f_{u,v}^e - \sum_{e \in \Gamma^-(u)} f_{u,v}^e = 1, u, v \in Q, u \neq v \\
 & \sum_{e \in \Gamma^+(v)} f_{u,v}^e - \sum_{e \in \Gamma^-(v)} f_{u,v}^e = -1, u, v \in Q, u \neq v
 \end{aligned}$$

$$\begin{aligned}
& \sum_{e \in \Gamma^+(w)} f_{u,v}^e - \sum_{e \in \Gamma^-(w)} f_{u,v}^e = 0, \\
& u, v \in Q, u \neq v, w \in V \setminus \{u, v\} \\
& 0 \leq f_{u,v}^e \leq F, \quad u, v \in Q, u \neq v, e \in E \\
& x_e \geq 0 \quad e \in E
\end{aligned}$$

This LP is the same as the one in Section 4.3.2, presented together with its running time characteristics; it is shown here for self-containing and easy reference purposes. Two points need further attention. First, in order to take the link availability into consideration, c_e is the cost per unit of bandwidth (which is set to 1 in this study) divided by the link availability. In this way the VPN provisioning will prefer links with higher availabilities. Secondly, the MFTP constraint (F) sets an upper limit on the maximum fraction of traffic on a path by mandating that the fraction of traffic routed on each link on this path cannot exceed a certain value. The following section will show that the MFTP constraint can guarantee multiple alternative paths for each VPN endpoint pair. In addition, it can also guarantee some of the paths are link-disjoint. This property is crucial for provisioning connections with availability requirements.

Before finding the multiple link-disjoint paths for each VPN endpoint pair, the results from the multi-path routing LP are converted from arc flow form to path flow form with the flow decomposition algorithm [103], i.e., a set $\mathcal{P}_{u,v}$ for each (u,v) pair specifying the set of alternative paths together with the fractions of traffic on them. Note that the set of paths usually are overlapped on each other.

b) The MFTP constraint

Property 2: The MFTP constraint guarantees that there are multiple link-disjoint paths for each VPN endpoint pair.

This property can be established through an algorithmic process to find the multiple link-disjoint paths. First pick up a path from the set of paths for a VPN endpoint pair, e.g., the path that carries the maximum fraction of traffic $p_{\max}$. The MFTP constraint sets an upper limit on the maximum fraction of traffic on a path. It does this by restricting the fraction of traffic on each individual link on this path to be less than a constant F , i.e., $f_{u,v}^e \leq F, F < 1, \forall e \in p_{\max}$. This in effect has limited the sum of fractions associated with this path and all the paths that have overlapping links with it is less than 1. Thus there must exist at least one path that carries some fraction of traffic and has no overlapping links with the selected path; otherwise the flow conservation constraint will be violated. Hence we could prune all the paths that have overlapping links with the selected path, and each of the remaining paths will be link-disjoint to it.

The MFTP constraint is in sharp contrast to the link capacity constraint in [10], which cannot provide true multiple-path guarantee as discussed in Section 4.2.2. The actual number of link-disjoint paths between a VPN endpoint pair depends on the network topology, most notably the node degree of the VPN endpoints. For example, there cannot be more than two link-disjoint paths for a VPN endpoint pair if one of the endpoint has only a degree of 2.

c) Multiple link-disjoint paths for each endpoint pair

This section presents the methods to find the multiple link-disjoint paths. The set of paths for a VPN endpoint pair is first sorted in descending order according to the fractions of traffic on them, and the first path for each endpoint pair is always kept as one of the link-disjoint paths. The remaining paths are examined sequentially one-by-one; if the path is link-disjoint to all previously found paths, and then it is added to the

set of link-disjoint paths. A parameter NO_LINKDISJOINT_PATHS is used to constrain the number of paths between each VPN endpoint pair. If the number of link-disjoint paths for an endpoint pair is greater than the value of the parameter, then only the first NO_LINKDISJOINT_PATHS paths are selected. In order to maintain the hose requirements, the paths left out by the link-disjoint path selection process are aggregated to the set of link-disjoint paths.

As the path aggregation algorithm presented in Chapter 4, the objective of the link-disjoint path selection and path aggregation processes is not to increase the total bandwidth requirement too much. To deal with this concern, paths carrying relatively larger fractions of traffic are preferred as the final link-disjoint paths. In addition, in the path aggregation process, the current candidate path is always aggregated to the link-disjoint path that has the most overlapping links with it. This process could overcome the too-many-path problem that often results from over-splitting, i.e., many overlapping paths each carrying a small fraction of traffic.

The entire bandwidth-centric approach is summarized in Figure 5.2. In the study, we set the input argument NO_LINKDISJOINT_PATHS to 3, i.e., a VPN endpoint pair has two or three link-disjoint paths. The multi-path routing LP is known to be a polynomial algorithm [10]. The complexity of finding the link-disjoint paths is $O(|Q|^2)$ ($|Q|$ is the number of VPN endpoints). This is because this process runs for every VPN endpoint pair, and the number of paths between a VPN endpoint pair is usually bounded by a relatively small constant (e.g., 10), so the most significant factor in determining the complexity is the number of VPN endpoints.

```

input: NO_LINKDISJOINT_PATHS
for each of VPN endpoint pair {
step 1: sort the set of paths in descending order according to the fractions of
        traffic associated with them
step 2: find out the set of link-disjoint paths as destination paths and the other
        paths as candidate paths. If the no of link-disjoint paths is greater than
        NO_LINKDISJOINT_PATHS, the excessive paths are treated as
        candidate paths to be aggregated
step 3: for each of the candidate path {
        step 3.1: find a destination path that has the most overlapping links with it,
                and aggregate the current candidate path
        step 3.2: if no suitable destination path is found in step 3.1, i.e., the current
                candidate path is link-disjoint to all the paths in the destination set,
                aggregate the current candidate path to the destination path with
                the smallest fraction of traffic
        }
}
step 4: convert the results from path flow form back to arc flow form, and
        compute a new set of link bandwidth reservation

```

Figure 5.2 Complete bandwidth-centric approach

Compared with previous studies, the bandwidth-centric approach has a number of advantages. Firstly it provides true multiple-path guarantee. Secondly the works in [86, 87] can only find multiple paths for a single source-destination pair, whereas our algorithm guarantees multiple link-disjoint paths for each VPN endpoint pair. Thirdly the bandwidth-centric approach can be easily extended to incorporate path-count constraint and link capacity constraint. For example, if there are more link-disjoint paths than desired, the excessive paths can be aggregated. In addition, other desirable features, such as load balancing, can also be incorporated as will be shown in the next chapter.

5.4 Admission Control

A connection request with availability requirement is defined by the tuple $\langle s, d, b, a \rangle$, where s , d are the source and destination VPN endpoints, b the bandwidth requirement, and a the availability requirement. A connection is admitted into the network only when the following three conditions are satisfied:

1. Admitting this connection should not violate the egress and ingress capacities of the source and destination endpoints respectively.
2. The connection's bandwidth requirement is satisfied, i.e., one of the paths between s and d has at least a free bandwidth no less than b .
3. The connection's availability requirement is satisfied, either by a single path or by a primary-backup path pair. For the single path case, it means the path availability $A(p_1) \geq a$; for the primary-backup path pair case, it means the joint availability $A(p_1 p_2) \geq a$. For the latter case, it requires that the free bandwidth on both the primary path and the backup path must be greater than or equal to b .

For each VPN endpoint pair, the set of alternative link-disjoint paths is sorted in descending order according to their availabilities, and the admission control will examine the paths in that order. The admission control consists of three stages, hose capacity checking, bandwidth checking, and availability checking. If a connection request is blocked in an earlier stage, it is assumed lost and all later stage checking is skipped; connection blockings occurred at each of the three stages are called hose capacity blocking, bandwidth blocking, and availability blocking respectively. Three counters, BLK_{cap} , BLK_{bw} , and BLK_{av} , are used to track the number of blocked connections due to capacity blocking, bandwidth blocking, and availability blocking.

These counters are first initialized to 0; when a connection is blocked and classified into one category, the corresponding counter is increased by 1. Denote the number of total connection requests by N_{req} , the blocking probabilities of hose capacity blocking, bandwidth blocking, and availability blocking are given respectively by

$$\text{capacity blocking} = \frac{BLK_{cap}}{N_{req}} \times 100\% ,$$

$$\text{bandwidth blocking} = \frac{BLK_{bw}}{N_{req}} \times 100\% ,$$

$$\text{availability blocking} = \frac{BLK_{av}}{N_{req}} \times 100\% .$$

Distinguishing blockings in this manner will give better insight into the ability of various routing schemes in provisioning availability guaranteed services. The admission process is presented below in a step-wise manner followed by an example.

Step 1: Check whether the admission of the incoming connection request will violate the hose ingress and egress capacities. If yes, the admission process ends with the connection request blocked and classified into hose capacity blocking; otherwise continue to Step 2.

Step 2: Check whether a path between s and d with adequate free bandwidth ($\geq b$) can be located. If yes, denote the path as p_p and go to step 3; otherwise, the admission process ends with the request blocked and classified into bandwidth blocking.

Step 3: Check if the availability $A(p_p) \geq a$. If yes, the admission process ends with the connection request provisioned with this single path p_p ; otherwise, go to step 4.

Step 4: The path p_p found in Step 3 is regarded as primary path. Check if there is a backup path with adequate free bandwidth ($\geq b$). If yes, denote this path as p_b and go

to Step 5; otherwise, the admission process ends with the request blocked and classified into availability blocking.

Step 5: Calculate the joint availability of the primary path and backup path. If the joint availability $A(p_p p_b) \geq a$, the connection is admitted and provisioned with two paths. Otherwise, the admission process ends with the connection request blocked and classified into availability blocking.

Suppose there are three paths between endpoints s and d , and they have been sorted in descending order according to their availabilities. The three paths are denoted by $p_1(1, 0.99)$, $p_2(10, 0.95)$ and $p_3(8, 0.92)$, where the first term in parenthesis represents the path's free bandwidth and the second represents the availability. Also assume that the egress capacity and the ingress capacity of s and d respectively still have more than 5 units of free bandwidth. A connection request $r\langle s, d, 5, 0.99 \rangle$ arrives, requesting a bandwidth of 5 units and availability of 0.99.

At the first step of the admission process, as s and d have enough free hose capacities, the admission process will proceed to the second step, where the second path p_2 will be chosen, as the first path p_1 only has a free bandwidth capacity of 1, unable to meet the connection request's bandwidth requirement. Subsequently in step three, after observing that the availability of p_2 (0.95) is less than that being requested (0.99), the admission process enters the forth step to look for a backup path, where path p_3 is selected, for it also has a free capacity greater than 5 units. At step 5, the joint availability of p_2 and p_3 is calculated, yielding $A(p_2 p_3) = 0.996 > 0.99$. So finally this connection request is accepted and provisioned with two paths, p_2 and p_3 .

When a connection is admitted, b units of bandwidth are deducted from the hose egress and ingress capacities of s and d respectively, as well as from the path(s) for this connection. This includes the backup path where a primary-backup path pair is needed, as protection resource must be reserved in advance.

The admission process reveals that using multi-path routing for availability-guaranteed service implies some tradeoffs. First of all, providing availability-guaranteed service needs more provisioning, for availability-guarantee is achieved through overprovisioning, using either protection or restoration techniques. In this chapter, we alleviate the overprovisioning problem by taking advantage of the intrinsic overprovisioning in the hose model, but this gives rise to another problem. As backup paths for connection requests with higher availability requirements consume extra bandwidth, the VPN is no longer able to handle the worst case traffic split. In addition, the admission control becomes more complex. The sorting of paths according to their availabilities, the examination of the paths to find a primary-backup path pair, and the joint availability calculation require more overhead as compared to using single path to provide availability guarantee.

5.5 Simulations and Numerical Results

Experiments are conducted using the NSFNET topology with the same set of hose parameters as shown in Table 4.1 in Section 4.4. All network nodes are used as VPN endpoints. Link availabilities are chosen from the following three values $\{99.95\%, 99.96\%, 99.97\%\}$, roughly proportional to the inverse of their distances. Connection availability requirements are uniformly distributed over $\{99\%, 99.9\%, 99.99\%\}$.

The network is assumed to support two classes of connections, namely class-1 and class-2, with bandwidth requirements $b=1$ and $b=5$ respectively. Connections arrive at each VPN endpoint according to an independent Poisson process, and their holding time is exponentially distributed. The number of connections for each class is inversely proportional to their bandwidth requirement, i.e., the number of connections with $b=1$ is five times that of connections with $b=5$. Thus if the total network offered load is ρ , the network offered load of class-1 and class-2 connections are $\rho(1)= 5\rho/6$ and $\rho(2)= \rho/6$ respectively. The traffic injected by an endpoint into the network is proportional to the ratio of its egress parameter to the sum of egress parameters of all VPN endpoints; similarly the traffic injected by an endpoint and directed to a specific destination is proportional to the ratio of the destination endpoint's ingress parameter to the sum of ingress parameters of all VPN endpoints (excluding that of the source endpoint). In this way, endpoints with larger egress parameters are more likely to be the source of a connection; and endpoints with larger ingress parameters are more likely to be the destination of a connection. A detailed mathematical description on the traffic generation can be found in Section 3.3. The simulation duration is set to 50 hours.

The study includes tree routing, and the two multi-path routing heuristics. Best-cost (with minimum total bandwidth capacity) shortest-path tree is used for tree routing. First the shortest-path tree for each node in the network is calculated, then, the tree structure with the minimal cost is taken as the final routing result. The technique used to calculate the bandwidth reservation on a tree link follows that in [8]. Techniques presented in Section 5.3 are used to find a routing and to dimension the VPN for the two heuristic approaches. In order to make a fair comparison between the two multi-

path routing heuristics, the two approaches are made to have the same number of paths for each VPN endpoint pair. The paths in both approaches are sorted in descending order according to their availability; for the availability-centric approach, a path is assigned the same fraction of traffic as its counterpart in the bandwidth-centric approach. For example, if the path with maximum availability in the bandwidth-centric approach carries 0.7 of the total traffic, then the path with maximum availability in the availability-centric approach also is assigned 0.7 of the total traffic. The fraction of traffic on a path will affect the bandwidth demand of each approach. As the input argument NO_LINKDISJOINT_PATHS is set to 3, there are two to three paths between every VPN endpoint pair.

The results are presented in four parts. The first part shows that compared with tree routing, the proposed multi-path routing heuristics significantly increase the joint availability for each VPN endpoint pair. The second part shows that availability blocking is the one that usually dominates the overall blocking probabilities. The third part shows that as a result of the increased availability, the blocking probabilities from multi-path routing heuristics are drastically lower than tree routing in provisioning connections with availability requirements. Finally, in the fourth part, a comparison of the availability-centric approach and the bandwidth-centric approach for multi-path routing is made together with discussions of their relative merits.

5.5.1 Joint Availabilities of Tree Routing and Multi-Path Routing

The joint availabilities between each VPN endpoint pair for tree routing and multi-path routing heuristics are shown in Figure 5.3. With tree routing, there is only one path, so the joint availability is the availability of the single path. For the multi-path routing heuristics, the availability is calculated based on two paths: the primary and

the backup paths. The two paths used are the paths with the smallest and second smallest availability; and the technique presented in Section 5.2.1 is used for the calculation. The results give the minimum availability that the multi-path routing heuristics can provide should a primary-backup path pair is needed. Joint availabilities are good indicators of the abilities of tree routing and multi-path routing in provisioning availability guaranteed services.

Figure 5.3 Joint availabilities

With tree routing, the joint availability varies with different endpoint pair; but for all the endpoint pairs it is less than 99.99%. Thus no connections with an availability requirement $a=99.99\%$ or higher can be provisioned. In addition, the joint availabilities of 104 out of the 182 possible endpoint pairs fall under 99.9%, indicating a large amount of connections with availability requirement $a=99.9\%$ will also be blocked. On the other hand, the joint availabilities with the multi-path routing heuristics are largely uniform, with the minimum one to be 99.9996%, well above 99.99%. Thus tree routing is not suitable for provisioning availability guaranteed services. The results also show that the difference between the joint availabilities of the bandwidth-centric (Multi-BW) approach and the availability-centric (Multi-AV)

approach is rather small as indicated by the overlapped curves. In the study, the largest difference is 2×10^{-6} . This suggests that so far as the ability to provide joint availability is concerned, the two multi-path heuristics are about the same.

5.5.2 Effects of Three Types of Blocking

For tree routing and multi-path routing heuristics (with $F=0.9$ for the bandwidth-centric approach), the total blocking, hose capacity blocking, bandwidth blocking, and availability blocking are plotted together for connections with availability requirement $a=99.9\%$ in Figure 5.4, Figure 5.5 and Figure 5.6 respectively, where b stands for the bandwidth requirement of a connection's request.

For tree routing, the single path cannot provide adequate availabilities for connections with higher availability requirements, thus the total blocking probabilities are dominated by availability blocking. The high availability blocking brings in two other results. Firstly, hose capacity blocking is quite small, usually lower than availability blocking by several orders of magnitude. For example, for connections with

Figure 5.4 Various types of blockings for tree routing

bandwidth requirement $b=1$, at the load of 90 Erlangs, hose capacity blocking is lower than availability blocking by about four orders of magnitude. Secondly, bandwidth blocking is zero. This is because in the hose model, adequate bandwidth is reserved such that the traffic not violating the ingress/egress capacities will not be blocked. As most of the traffic is blocked due to availability constraint, much of the hose ingress/egress capacities are not used, resulting in low hose capacity blocking. Likewise, much of the link bandwidth is not used, resulting in zero bandwidth blocking. As much as 57% of the connections throughout the entire load region is blocked due to availability. Since much of the bandwidth is left unused, the network resource is not used efficiently in tree routing. The high availability blocking shows that tree routing is inadequate for provisioning availability guaranteed services. In addition, increasing bandwidth cannot help to improve the situation, as it does not increase the path availability.

For the bandwidth-centric approach, total blocking is also dominated by availability blocking; hose capacity blocking comes next, and bandwidth blocking is the smallest. To support high availability in multi-path routing, much bandwidth is reserved on primary as well as backup paths. Consequently, bandwidth blocking may occur. However, as shown in Figure 5.5, bandwidth blocking is still the smallest among the three constraints. In multi-path routing, a backup path can be provided to increase the joint availability. So, availability blocking is much lower than that of tree routing. At the load of 90 Erlangs, the availability blocking of bandwidth-centric approach is lower than that of tree routing by 1 order of magnitude for connections with $b=5$, 2 orders of magnitude for connections with $b=1$; at the load of 225 Erlangs, the

improvement is 2 times for connections with $b=5$ and 10 times for connections with $b=1$.

In the availability-centric approach, the total blocking is dominated by hose capacity blocking instead of availability blocking, and the bandwidth blocking is practically

Figure 5.5 Various types of blockings for bandwidth-centric approach

Figure 5.6 Various types of blockings for availability-centric approach

zero. This is because the availability-centric approach consumes much more bandwidth than the bandwidth-centric approach. For the set of hose parameters studied, the availability-centric approach consumes as much as 1.73 times bandwidth as the bandwidth-centric approach; the average value for 50 sets of randomly generated hose parameters is 1.77. More bandwidth consumption means more chances to find a primary-backup path pair for connections with high availability requirement; thus, the availability blocking drops. Lower availability blocking leads to more hose capacity to be occupied, thus the hose capacity blocking will correspondingly become higher. Consequently availability blocking falls below hose capacity blocking. With the provisioning of abundant bandwidth, the bandwidth requirement is no longer a restricting factor, so the bandwidth blocking becomes zero.

The results in Figure 5.5 and Figure 5.6 show that in multi-path routing, connections with smaller bandwidth requirement ($b=1$) has a lower blocking probability than connections with higher bandwidth requirement ($b=5$). This is because the chances are smaller for connections with $b=5$ to find adequate free hose capacity, a primary path and a backup path, so the blocking probability for $b=5$ is higher than that with $b=1$.

For connections with availability requirement $a=99.99\%$, the situation is similar to that with $a=99.9\%$. The difference is that as no path in tree routing has availability greater than or equal to 99.99% , so all connections with $a=99.99\%$ are blocked. For connections with $a=99\%$, as the smallest path availability in both tree routing and multi-path routing is greater than 99% , so the connections will be blocked due to insufficient hose capacity and bandwidth, giving a very low blocking probability for all three routing schemes.

5.5.3 Total Blocking

This section compares the blocking performance of tree routing, multi-path routing bandwidth-centric approach and multi-path routing availability-centric approach. In the experiment, an incoming connection request is equally likely to assume any one of the availability requirement of 99%, 99.9%, or 99.99%. Thus connections with different availability requirements coexist in the VPNs. The results are shown in Figure 5.7 and Figure 5.8 respectively. Note that the curves for the availability-centric approach are overlapped on each other, as in this routing approach, hose capacity blocking instead of availability blocking dominates.

The results show that multi-path routing heuristics deliver much better blocking performance than tree routing for connections with high availability requirement at the expense of connections with lower availability requirement. First look at connections with availability requirement $a=99.99\%$ and $a=99.9\%$. For connections with bandwidth requirement $b=1$, at low load region bandwidth-centric approach is better than tree routing by about two orders of magnitude; and the availability-centric

Figure 5.7 Total blocking of tree routing and two heuristics with $b=1$

Figure 5.8 Total blocking of tree routing and two heuristics with $b=5$

approach is in turn better than the bandwidth-centric approach by about 4 to 5 times. At high load region, the improvement of the bandwidth-centric approach over tree routing is around 1 order of magnitude, and the improvement of availability-centric approach over bandwidth-centric approach is around two times. For connections with $b=5$ at low load region, the bandwidth-centric approach is better than tree routing by about 1 order of magnitude; and the improvement of availability-centric approach over bandwidth-centric approach is about 4 to 5 times. At high load region, the bandwidth-centric approach is better than tree routing by 1.5 to 2 times; and the availability-centric approach is better than the bandwidth-centric approach by around 2 times.

For connections with $a=99\%$, tree routing is better than availability-centric approach by about 1 order of magnitude for connections with both $b=1$ and $b=5$, and the bandwidth-centric approach is slightly better than the availability-centric approach. In tree routing, very high blockings of connections with $a=99.99\%$ and $a=99.9\%$ leave much free hose capacity and much free bandwidth in the network. As the minimum

path availabilities in all three routing schemes are all above 99%, so connections with $a=99\%$ will be blocked solely due to hose capacity blocking or bandwidth blocking. Thus connections with $a=99\%$ in tree routing benefit from the unused capacity and enjoy a much lower blocking probability.

5.5.4 Comparison between the Two Heuristic Approaches

For multi-path routing heuristics, there is a trade-off between bandwidth usage and availability blocking, i.e., using more bandwidth to achieve lower availability blocking. More bandwidth provides higher chances to locate a primary-backup path pair to cater for connections with high availability requirements, so the blocking probability will be reduced.

To see the effect of increased bandwidth on the multi-path routing heuristics, simulations are carried out with the link bandwidth capacity in the bandwidth-centric approach scaled up by different amount. For example, for a scaling factor of 1.2, the link capacity in the bandwidth-centric approach is scaled up by 1.2 times. For different scaling factors, the blocking probabilities of connections with availability requirement $a=99.99\%$ are shown in Figure 5.9 and Figure 5.10.

The results show that different scaling factor results in different blocking probability. The performance is better when more bandwidth is provided. While it is desirable to support high availability, the need to provide more bandwidth may cause the concern of overprovisioning. Without considering availability, the hose model generally requires twice as much bandwidth as the conventional pipe model [11]. To support high availability, additional bandwidth is needed. With $F=0.9$, for 50 sets of randomly

Figure 5.9 Blocking probabilities versus scaling factors, $b=1$

Figure 5.10 Blocking probabilities versus scaling factors, $b=5$

generated hose parameters, the overprovisioning of bandwidth-centric approach and availability-centric approach to the original hose model is shown in Figure 5.11. The bandwidth-centric approach requires an average of 1.16 times bandwidth of the original hose model. In turn, the availability-centric approach requires 1.77 times bandwidth of the bandwidth-centric approach. Though it is difficult to accurately estimate the amount of bandwidth needed to provision the hose model, such that the

blocking is solely caused by the hose ingress and egress capacities. Nonetheless, a rough estimate will be more than 2 (1.16×1.77) times that of the original hose model. Thus, compared to the pipe model, this amounts to 4 times overprovisioning. This amount of overprovisioning is a costly tradeoff between bandwidth provisioning and blocking performance.

Figure 5.11 Overprovisioning of two multi-path approaches

The blocking probability depends on bandwidth provisioning as well as network load. In practice, a VPN may not always operate at high load. Consequently, the amount of bandwidth required may not be too high. For the availability-centric approach, the amount of bandwidth required is high. It is difficult to adjust the bandwidth requirement to achieve certain performance, while still satisfying the hose capacity constraint. With the bandwidth-centric approach, it is more flexible, the desired bandwidth-performance tradeoff can be adjusted by scaling the link bandwidth capacity according to practical needs.

In summary, the two multi-path routing heuristics have their merits and drawbacks. The availability-centric approach can run very fast, suitable for situations where quick VPN setup is needed and very high availability is desirable. The disadvantage is that it consumes much more bandwidth. For the bandwidth-centric approach, it allows one to adjust the bandwidth-performance tradeoff; it is easily extensible to incorporate the link capacity constraints as well as other features such as load balancing. The weakness is that it requires longer computation time due to the LP. For our study, the computation time is around 10-15 minutes on a Pentium IV PC running Windows XP with 256 MByte memory. If the VPN is designed to operate for a certain length of time after the initial setup, this amount of setup time is still acceptable.

5.6 Conclusion

This chapter studies the availability guaranteed service provisioning in hose model VPNs such that the VPN connections satisfy the hose capacity constraint, link bandwidth constraint and availability constraint. Two multi-path routing heuristics are proposed: the bandwidth-centric approach and the availability-centric approach. The two heuristics employ multiple link-disjoint paths, which make the provisioning process fast and straightforward. Subsequently the link-disjoint paths are used to provision connections with availability requirements. The two heuristics elegantly solve the too-many-path problem encountered in other studies in literature. In addition, they also eliminate the path-overlapping problem, which tends to make the provisioning of connections with availability requirement cumbersome and lengthy.

Simulation results show that availability blocking usually dominates the total blocking in provisioning availability guaranteed services. By employing a primary-backup path pair, the proposed heuristics are able to greatly increase the joint availability for each

VPN endpoint pair, thus significantly reduce the blocking probabilities in provisioning connections with availability requirements. Compared to tree routing, the improvement could be in a few orders of magnitude, which is highly significant. The two multi-path routing heuristics have their respective strengths and weaknesses, and can be deployed in different situations accordingly.

Chapter 6 Load Balancing in Hose Model VPNs with Multi-Path Routing

6.1 Introduction

Though a number of research works in literature have studied load balancing issues, none of them was applicable to hose model VPNs. The works in [91-95] all focused on a single pair of nodes, thus were not suitable for load balancing in network design and provisioning. Furthermore they assumed (with no guarantee) that multiple paths had been set up between the source-destination pair. The multiple-path assumption may be valid in the Internet, but it may not hold in VPNs. The works in [67, 86, 96, 97] could be used to provision a VPN, yet they all assumed that a static traffic matrix was known in priori, which made the VPN resemble the pipe model.

This chapter investigates load balancing in provisioning hose model VPNs and proposes a novel multi-objective multi-path (MOMP) routing linear program (LP) for the purpose. The objective of load balancing is considered at both the service provider level and the VPN customer level. The service provider level load balancing aims at minimizing the bandwidth reservation on the most loaded network link, thus avoiding potential congestion problems in service provider network. On the other hand, the VPN customer level load balancing aims at provisioning multiple alternative paths for each VPN endpoint pair, thus allowing load balancing within the VPN.

6.2 Load Balancing in Hose Model VPNs

Tree routing and single-path routing achieve greater flexibilities through multiplexing traffic together on a small number of network links. However, this action may raise congestion problems on heavily loaded links while other links are only lightly loaded.

In some cases, this concentration of traffic on a small number of links may cause blocking of future non-VPN traffic.

Figure 6.1 shows a simple directed network topology of a service provider. V_1, V_5, V_6 , and V_9 are VPN endpoints and are labeled with their (egress, ingress) parameters; all other nodes are non-VPN endpoints. Suppose the hose model VPN is dimensioned using single-path routing or tree routing with the usual objective of minimizing the total bandwidth requirement, thus the resulting VPN topology will include the set of links marked with an asterisk. Since all traffic from $\{V_1, V_6\}$ to $\{V_5, V_9\}$ traverses the link (7,8), 100 units of bandwidth will be reserved on it, leading to potential congestion problems on this link. If the free bandwidth on link (7,8) is depleted due to this VPN provisioning, it will leave node 10 disconnected from nodes 5, 8, 9, and 11 for non-VPN traffic. Thus all future non-VPN traffic originating from node 10 to nodes 5, 8, 9, and 11 will be blocked.

In addition, the one-endpoint-pair-one-path property of single-path routing and tree routing puts this VPN at risk of service degradation caused by uneven traffic distribution. As all traffic from $\{V_1, V_6\}$ to $\{V_5, V_9\}$ traverses the link (7,8), the load on (7,8) will increase quickly. Consequently the traffic may experience long delay and high losses, which tend to increase sharply as the traffic load increases.

Figure 6.1 an example for load distribution

On the other hand, multi-path routing has the potential to overcome the above drawbacks. With multi-path routing, traffic from a source endpoint can take several alternative paths to a destination endpoint, thus the maximum bandwidth reservation on the most loaded link could be reduced. In the above example, if multi-path routing is used for the provisioning, an additional path, 1-2-3-4-5, can be added for the VPN endpoint pair (V_1, V_5) . The newly added path may unload 50 units of traffic from link (7,8) and leave some free capacity there, thus the traffic from node 10 can reach nodes 5, 8, 9, and 11. As the traffic load on the link (7,8) decreases, the traffic traversing it will enjoy shorter delay and lower losses.

Furthermore, splitting of traffic among multiple paths is actually load balancing at VPN customer level, which can improve the hose model VPN's performance. As the traffic load on link (7,8) decreases, the traffic will experience shorter delay and lower losses. Thus overall quality of service (QoS) provided by the VPN is enhanced.

In addition, multi-path routing also improves the VPN's reliability. In tree routing and single-path routing, a single link failure will partition the VPN into two disconnected parts. But in multi-path routing, so long as the link failures do not cover the multiple alternative paths simultaneously, the communication can still continue; and since the disruption only affects parts of the traffic, it is also easier to restore them. In the above example, if the path 1-2-3-4-5 fails due to link failure, the communication between (V_1, V_5) can still take the path 1-7-8-5. In Chapter 5, we have shown multi-path routing is able to bring in significant improvement in provisioning availability guaranteed services when compared to tree routing and single-path routing.

However, load balancing does not come as a natural result of multi-path routing, because some network links may still be heavily loaded; in addition, the good performance achieved by distributing load evenly among multiple alternative paths, and the benefits of increased reliability can only become real if there are indeed multiple paths between each VPN endpoint pair. This is what has been left out by previous studies.

6.3 Load Balancing with Multi-Objective Multi-Path Routing Algorithm

This section first gives the problem formulation, and then presents the multi-objective multi-path routing linear program. Subsequently we show that the MFTP constraint guarantees there are multiple paths between each VPN endpoint pair.

6.3.1 Problem Formulation

The service provider network is modeled as a bi-directed graph $G=(V,E)$, where V is the set of network nodes, and E the set of links. In bi-directional networks, a link $(i,j) \in E$ implies $(j,i) \in E$ as well. The set of VPN endpoints is denoted by Q , $Q \subseteq V$. For each $v \in Q$, an ingress and egress parameters are given to specify v 's traffic demand. x_e is the amount of bandwidth reserved on link e , such that all valid traffic matrix $\mathcal{D}$ can be accommodated. A traffic matrix $D=(d_{u,v})_{u,v \in Q}$, $D \in \mathcal{D}$, is valid only when the sum of endpoint v 's outgoing traffic does not exceed its egress parameter, as well as the sum of v 's incoming traffic does not exceed its ingress parameter.

The problem is to provision the VPN, such that:

1. All valid traffic matrix $\mathcal{D}$ can be accommodated;

2. The maximum of x_e is minimized;
3. There are multiple alternative paths between each VPN endpoint pair to enable the implementation of load balancing within the VPN.

6.3.2 The Multi-Objective Multi-Path Routing Algorithm

In order to balance the load among the set of service provider network links involved in the hose model VPN provisioning, one straightforward way is to minimize the maximum bandwidth reservation among the set of network links. However such a min-max objective will spread the bandwidth reservation as widely (i.e., use many links) and as evenly as possible, which often results in large increase in the total bandwidth requirement and excessive computation time. Our solution is a multi-objective multi-path (MOMP) optimization approach: while keeping the min-max objective, we also try to minimize the total bandwidth requirement; and by assigning proper weights to the two objectives, we can achieve a balance between them.

In the remaining part of this chapter, we shall call the objective of minimizing total bandwidth requirement as the cost objective, because bandwidth is associated with monetary cost; and call the min-max objective as the load-balancing objective, because it is crucial for the load balancing purpose. Through a large number of experiments, we find that when the two objectives carry approximately equal weights, we can get the benefits from both sides: a smaller maximum load on service provider network links and also a relatively small total bandwidth requirement. A large weight for the cost objective often overrides the effects of introducing the load-balancing objective, and a large weight for the load-balancing objective will bring in undesirable results of excessive computation time and a large total bandwidth requirement. The weights for the cost objective and the load balancing objective are both set to 1.

In addition to load balancing at the service provider network level, multiple alternative paths are also desirable for implementing load balancing within the VPN. However the above multi-objective optimization approach cannot guarantee that there are multiple paths between each VPN endpoint pair. We overcome this drawback by introducing the maximum fraction of traffic on a path (MFTP) constraint, which will be shown shortly to be able to guarantee multiple paths for each VPN endpoint pair.

Now the MOMP routing linear program (LP) with the MFTP constraint can be summarized as follows:

$$\min \quad w_C \sum_{e \in E} c_e x_e + w_L \cdot x_{\max} \quad (6.1a)$$

s.t.

$$\sum_{u,v \in Q} d_{u,v} \cdot f_{u,v}^e \leq x_e, \quad D \in \mathcal{D}, e \in E \quad (6.1b)$$

$$x_e \leq x_{\max}, \quad e \in E \quad (6.1c)$$

$$\sum_{e \in \Gamma^+(u)} f_{u,v}^e - \sum_{e \in \Gamma^-(u)} f_{u,v}^e = 1, \quad u, v \in Q, u \neq v \quad (6.1d)$$

$$\sum_{e \in \Gamma^+(v)} f_{u,v}^e - \sum_{e \in \Gamma^-(v)} f_{u,v}^e = -1, \quad u, v \in Q, u \neq v \quad (6.1e)$$

$$\sum_{e \in \Gamma^+(w)} f_{u,v}^e - \sum_{e \in \Gamma^-(w)} f_{u,v}^e = 0,$$

$$u, v \in Q, u \neq v, w \in V \setminus \{u, v\} \quad (6.1f)$$

$$0 \leq f_{u,v}^e \leq F, \quad u, v \in Q, u \neq v, e \in E \quad (6.1g)$$

$$x_e \geq 0 \quad e \in E \quad (6.1h)$$

The objective function (6.1a) consists of two terms. The first one ($\sum_{e \in E} c_e x_e$) is the cost objective, which aims at minimizing the total cost of bandwidth requirement to dimension the hose model VPN; the second term, $x_{\max}$, is the load-balancing objective, which aims at minimizing the maximum bandwidth reservation among the set of network links. $x_{\max}$ is an upper bound on the set of x_e 's. w_C and w_L are the

respective weight of the cost objective and load balancing objective; they are both set to 1 in this study.

Constraint (6.1c) requires that all the link bandwidth reservations must be below a certain upper limit. It is the most important constraint for the load balancing objective at the service provider level. The MFTP constraint (6.1g) will be shown in following section that it guarantees there are multiple paths between each VPN endpoint pair, which are essential for load balancing at VPN customer level. All other constraints have the same meanings as that in Section 2.3.4.

The MOMP routing LP will give the result in arc flow form. A flow decomposition algorithm [103] can be used to convert the arc flows to path flows i.e., a set $\mathcal{P}_{u,v}$ for each (u,v) pair specifying the set of alternative paths. Label Distribution Protocol (LDP) [109] could be used to set up these paths, taking the advantage of explicit routing capability of MPLS.

6.3.3 Multiple Path Guarantee with the MFTP constraint

Property: The MFTP constraint guarantees that the multi-path routing LP will provide multiple alternative paths for each VPN endpoint pair.

The MFTP constraint sets an upper limit on the maximum fraction of traffic on a path by restricting the fraction of traffic on each link on this path to be no greater than a constant. This essentially constrains that the sum of fractions on a path and all the paths that have overlapping links with it is less than 1. Thus there exists at least one path that carries some fraction of traffic and has no overlapping links with the selected path; otherwise the flow conservation constraint will be violated. Section 5.3.3(b) has

given a stronger version of this property, plus an algorithmic process to find the set of alternative link-disjoint paths.

6.4 Other Hose Model VPN Provisioning Algorithms

For comparison purposes, this section presents two commonly used alternative algorithms for the hose model VPN provisioning that do not consider load-balancing features. These two algorithms are tree routing and single-objective multi-path (SOMP) routing.

6.4.1 Tree Routing Algorithm

It has been shown in [8] that the optimal tree can only be obtained polynomially when the hose parameters are symmetric, i.e., the ingress parameter is the same as the egress parameter. Best-cost shortest-path tree is used for tree routing. For this purpose, the shortest-path tree (with link weight set to 1) for every node in the network is calculated first, and then the one with the best cost is chosen as the final routing result. The method for calculating the bandwidth reservation on each tree link follows the approach in [8]. Detailed descriptions on how to obtain the link bandwidth reservation can be found in Section 2.3.3 and Section 4.2.1, thus no repetition is given here.

6.4.2 Single Objective Multi-Path (SOMP) Routing Algorithm

We adapt the linear program in [10] (without the link capacity constraint) and show it below. The sole purpose of the single-objective multi-path routing algorithm is to minimize the total bandwidth requirement.

$$\begin{aligned}
 & \min \quad \sum_{e \in E} c_e x_e \\
 & s.t. \quad \sum_{u,v \in Q} d_{u,v} \cdot f_{u,v}^e \leq x_e, \quad D \in \mathcal{D}, e \in E
 \end{aligned}$$

$$\begin{aligned}
 \sum_{e \in \Gamma^+(u)} f_{u,v}^e - \sum_{e \in \Gamma^-(u)} f_{u,v}^e &= 1, \quad u, v \in Q, u \neq v \\
 \sum_{e \in \Gamma^+(v)} f_{u,v}^e - \sum_{e \in \Gamma^-(v)} f_{u,v}^e &= -1, \quad u, v \in Q, u \neq v \\
 \sum_{e \in \Gamma^+(w)} f_{u,v}^e - \sum_{e \in \Gamma^-(w)} f_{u,v}^e &= 0, \\
 u, v \in Q, u \neq v, w \in V \setminus \{u, v\} \\
 0 \leq f_{u,v}^e &\leq F, \quad u, v \in Q, u \neq v, e \in E \\
 x_e &\geq 0 \quad e \in E
 \end{aligned}$$

Comparing SOMP with MOMP presented in Section 6.3, the SOMP does not carry the load balancing objective $x_{\max}$ and the load balancing constraint (6.1c). It may or may not include the MFTP constraint (6.1g).

Note that the work in [10] can neither do load balancing nor provide multiple-path guarantee for each VPN endpoint pair. The link capacity constraint states that the bandwidth reservation will not exceed the capacity. It does not guarantee the traffic will be distributed evenly, nor does it guarantee the availability of multiple paths for each VPN endpoint pair. Depending on the set of link capacities, there may still be only one path between some VPN endpoint pairs. In the example in Section 6.2, if link (7,8) has a relatively large residual capacity, then the algorithm in [10] will still produce a VPN topology comprising only links marked with asterisks.

6.5 Numerical Results and Discussions

We carry out simulation experiments with 50 sets of hose ingress and egress parameters on the NSFNET topology. The hose parameters are randomly generated and are uniformly distributed between 20 and 60. The parameters can take any appropriate units. For example, in IP networks, the unit could be in x Kbps; and in optical networks, the units could be in y Mbps or number of wavelengths.

The performance metric for service provider level load balancing is the amount of bandwidth reserved on the most loaded link, or the maximum link bandwidth reservation. A smaller maximum link bandwidth reservation indicates that the traffic is more evenly distributed in the network, so congestion problems are unlikely to occur. In order to show the effectiveness of the proposed MOMP routing on load balancing, we will compare this metric from the proposed MOMP routing to that from SOMP routing and tree routing.

For the VPN customer level load balancing, we show that the MFTP constraint indeed guarantees there are multiple paths for each VPN endpoint pair. This is in sharp contrast to tree routing and single-path routing, where the total traffic is routed on a single path for each endpoint pair.

6.5.1 Typical Traffic Distribution

For a typical set of hose parameters presented in Table 4.1, the traffic distribution of tree routing, SOMP routing and MOMP routing without the MFTP constraint (i.e., setting $F=1$) is shown in Figure 6.2, Figure 6.3, and Figure 6.4 respectively.

As shown in Figure 6.2 and Figure 6.3, tree routing and SOMP routing algorithms aggregate traffic on a small number of network links, thus the bandwidth reservation on those links are very high. For tree routing the maximum value is 182, and for SOMP, it is 212. The SOMP routing without the MFTP constraint often turns out to be single-path routing and even tree routing. Although SOMP routing and tree routing often consume the same amount of total bandwidth, their resulting VPN topology and the bandwidth distribution on links generally are not the same, thus they usually have different maximum link bandwidth reservations.

Figure 6.2 Traffic distribution with tree routing

Figure 6.3 Traffic distribution with SOMP routing

Figure 6.4 Traffic distribution with MOMP routing

Figure 6.4 shows the proposed MOMP routing algorithm distributes the traffic more evenly among the set of network links, and thus effectively reduces the maximum bandwidth reservation to 112. Therefore the proposed MOMP routing algorithm can significantly reduce the maximum link bandwidth reservation as compared with tree routing and SOMP routing, even without the MFTP constraint.

The overall bandwidth capacities for tree routing and SOMP routing are 1976 units of bandwidth, whereas that for the MOMP routing algorithm is 2026.09 units, which amounts to a 2.5% increase. For tree routing and SOMP routing, there is only one path between every VPN endpoint pair; for MOMP routing, the number of paths between a VPN endpoint pair ranges from 1 to 3.

6.5.2 Service Provider Level Load Balancing without MFTP

Constraint

The results of maximum bandwidth reservations for tree routing, SOMP routing, and MOMP routing algorithms for 50 set of hose parameters are shown in Figure 6.5. The MFTP constraint is set to $F=1$.

The maximum link bandwidth reservation varies with different hose parameters for the three routing algorithms. But for all the parameters studied, SOMP routing has the largest maximum link bandwidth reservation, tree routing the second and MOMP routing the least. Let M_S , M_T , and M_M be the maximum link bandwidth reservation

Figure 6.5 Maximum link bandwidth reservation

for SOMP routing, tree routing and MOMP routing respectively. We define the bandwidth reduction of MOMP routing to SOMP routing as well as MOMP routing to tree routing as follows:

$$\Delta M_i (\%) = \frac{M_i - M_M}{M_i}, \quad i = \begin{cases} S & \text{for SOMP routing} \\ T & \text{for tree routing} \end{cases}$$

We calculate the bandwidth reductions for 50 sets of hose parameters, and show the maximum and the average values in Table 6.1.

Table 6.1 Bandwidth Reduction to SOMP Routing and Tree Routing

	Maximum	Average
ΔM_S (%)	68.3	41.7
ΔM_T (%)	61	34

These results show a remarkable reduction of the maximum link bandwidth reservation. The maximum reduction is over 60% and the average reduction is over 34%. This is highly significant. The reduction is attributed to the newly added load-balancing objective, which aims at minimizing the maximum bandwidth reservation among the entire set of network links.

6.5.3 Service Provider Level Load Balancing with MFTP Constraint

The proposed MOMP routing with the MFTP constraint can bring in more aggressive load balancing at the service provider level. For each different F value from 0.5 to 0.9 with a step length of 0.1, we calculate the maximum and average bandwidth reduction of the proposed MOMP routing to tree routing and show the results in Figure 6.6.

Figure 6.6 Bandwidth reduction of MOMP routing to tree routing versus F

Compared to the MOMP routing without the MFTP constraint where the average bandwidth reduction is about 30%, this version with the MFTP constraint delivers more significant load balancing results: the average bandwidth reduction in all but one case are well above 50%, the only exception case ($F=0.9$) is over 40%. For the maximum value, the results are about the same for all different F 's, slightly over 60%. For the average value, generally the smaller the F value the larger the reduction. Nevertheless, as F approaches 0.5, the reduction rate slows down and eventually drops slightly, which is explained in the following paragraph.

Without the MFTP constraint, SOMP routing, and possibly MOMP routing are trying to route all the traffic through the optimal path just like tree routing, thus the total bandwidth capacity is relatively smaller. Introducing the MFTP constraint forces some fraction of traffic to take a less optimal path. The effect is of two folds: on one hand it brings in the desired load balancing result; on the other hand it causes an increase in the total bandwidth capacity (please refer to Table 6.2). Higher total bandwidth capacity tends to draw up the average link bandwidth reservation; and the smaller the F value, the more evident the draw-up effect. The increased average link bandwidth reservation causes a slight drop in the average bandwidth reduction with $F=0.5$.

The MFTP constraint works equally well to reduce the maximum link bandwidth reservation for SOMP routing. However, the effect on the MOMP routing is greater. The results for the maximum and average bandwidth reduction of MOMP routing to SOMP routing are shown in Figure 6.7.

Figure 6.7 Bandwidth reduction of MOMP routing to SOMP routing versus F

Figure 6.7 reveals that when both routing algorithms are using the MFTP constraint, the proposed MOMP routing can still achieve a significant bandwidth reduction on the most loaded link. The maximum reduction ranges from 30% ($F=0.6$) to nearly 60% ($F=0.9$) and the average reduction ranges from 17% ($F=0.6$) to 36% ($F=0.9$). With the MFTP constraint, the traffic between a VPN endpoint pair is split among multiple alternative paths. This provides a potential for more aggressive load balancing, as the traffic can still aggregate on some bottleneck links. The load-balancing objective turns this potential into reality.

6.5.4 Bandwidth Overprovisioning for Load Balancing

The benefit of load balancing using MOMP routing is associated with a cost of increased bandwidth requirement as compared to tree routing and SOMP routing. Let BW_T , BM_S , and BM_M be the total bandwidth requirements of tree routing, SOMP routing and MOMP routing respectively. The bandwidth overprovisioning of the MOMP routing to tree routing as well as MOMP routing to SOMP routing is defined as follows:

$$\Delta BW_{Mi}(\%) = \frac{BW_M - BW_i}{BW_i}, i = \begin{cases} S & \text{for SOMP routing} \\ T & \text{for tree routing} \end{cases}.$$

For various F values from 0.5 to 1, the bandwidth overprovisioning of MOMP routing to tree routing and SOMP routing are summarized in Table 6.2. The bandwidth overprovisioning to SOMP routing is calculated when both algorithms are with the MFTP constraint. MOMP routing will cause an overprovisioning to tree routing from as low as 0.9% to as high as 20.5%. The maximum overprovisioning to SOMP routing is only 0.9%.

Table 6.2 Overprovisioning to Tree Routing and SOMP routing

F	0.5	0.6	0.7	0.8	0.9	1
$\Delta BM_{MT}(\%)$	20.5	11.9	7.8	5.3	3.1	0.9
$\Delta BM_{MS}(\%)$	0.2	0.3	0.4	0.5	0.7	0.9

As load balancing tends to distribute the traffic more evenly in the network, some fraction of traffic has to take less optimal paths than that in tree routing and SOMP routing, thus the total bandwidth requirement of MOMP routing will increase. Incorporating the MFTP constraint will make this effect more evident. The smaller the F value, the larger the fraction of traffic has to take less optimal paths and the larger the total bandwidth requirement. As the total bandwidth requirement of tree routing remains constant, ΔBM_{MT} increases while the F value decreases.

For SOMP routing, ΔBM_{MS} decreases correspondingly with the F value. On one hand, the MFTP constraint increases the total bandwidth requirement of SOMP routing. On the other hand, the load balancing objective and the MFTP constraint both tend to split the traffic among multiple paths. This overlapping effect slows down the bandwidth increase in MOMP routing. Consequently, the total bandwidth requirement

increases at a lower rate with MOMP routing as compared to SOMP routing. So, when the F value decreases, ΔBM_{MS} decreases accordingly. Thus we conclude that the benefit of the load balancing outweighs the disadvantage of the bandwidth increase.

Moreover, the overprovisioning is well justified because multi-path routing delivers many advantages over tree routing, including optimal polynomial algorithm for general hose parameters [10], enhanced ability to provide availability guaranteed services (Chapter 5), and the ability to balance load as shown in this chapter. In fact, the overprovisioning bandwidth can be controlled within 10% by selecting a relatively large F value, e.g., $F=0.7, 0.8$, or 0.9 . In this way, one can enjoy the benefits of multi-path routing without paying too high a price.

6.5.5 Multiple Paths for VPN Customer Level Load Balancing

The hose model explicitly allows a VPN endpoint to arbitrarily distribute its traffic to the other endpoints, and also allows an endpoint to dynamically change its traffic pattern. This flexibility makes the traffic pattern inside the VPN unpredictable, and thus causes potential congestion problems, as multiple VPN endpoint pairs may take the same links for their communications. Therefore it is highly desirable for every VPN endpoint pair to have multiple alternative paths, thus allowing some load balancing mechanism to distribute the traffic more evenly onto the multiple paths. For example, the routing of a new traffic demand can be based on some indicators of path congestion.

By definition, all VPN endpoint pairs in tree routing and single-path routing have only one path; the maximum and average number of paths for SOMP routing has been studied in Chapter 4. Regarding MOMP routing, for the typical set of hose parameters

in Table 4.1, 24 endpoint pairs without the MFTP constraint have only one path, other endpoint pairs have 2 to 3 paths. On the other hand, the MFTP constraint provides a guarantee of multiple paths for each VPN endpoint pair. If the MOMP routing algorithm generates too many paths, the path aggregation algorithm presented in Chapter 4 could be used to suppress the excessive paths. Figure 6.8 presents the maximum number of paths before and after aggregation for a typical set of hose parameters, and the average number of paths for 50 set of hose parameters before and after aggregation. The threshold value for aggregation is 0.1.

Figure 6.8 Maximum and average number of paths

In addition, compared to MOMP routing without aggregation, the path aggregation algorithm will cause bandwidth capacity and possible maximum bandwidth reservation increase. The results are summarized in Table 6.3.

When the MFTP constraint takes the value of 0.7, for the typical set of hose parameter studied, the aggregation algorithm reduces the maximum number of paths between a

Table 6.3 Effects of aggregation on bandwidth capacities

MFTP	0.5	0.6	0.7	0.8	0.9	1
Bandwidth capacity increase (%)	25.9	22.1	13.5	9	6.6	0.3
Maximum bandwidth reservation increase (%)	28.4	26.6	17.1	10	7.7	1.3

VPN endpoint pair from 7 to 5; for the 50 set of hose parameters, the average number of paths is reduced from more than 4 to 3. Furthermore, the aggregation algorithm causes 13.5% and 17.1% total bandwidth capacity and maximum bandwidth reservation increase respectively. The results for $F=0.8$, 0.9 and 1 continuously improve over that for $F=0.7$. For example, the maximum bandwidth reservation increase for $F=0.8$ is only 10%. Therefore, even with the path aggregation algorithm, larger values of F , like 0.8, 0.9, and 1, already deliver very good load-balancing effect, only cause small total bandwidth capacity and maximum bandwidth reservation increase, and also limit the average number of paths between a VPN endpoint pair to a manageable range, it seems little incentive to go for smaller value of F .

6.6 Summary

In this Chapter, we propose a multi-objective multi-path (MOMP) routing algorithm for load balancing in provisioning hose model VPN services. Compared to tree routing, the MOMP routing algorithm is able to reduce the bandwidth reservation on the most loaded link by as much as 50%. By incorporating the MFTP constraint, the MOMP routing algorithm guarantees the availability of multiple paths between each VPN endpoint pair, thus allowing load balancing at the VPN customer level. With the MFTP constraint, further reduction on the maximum link bandwidth reservation can be achieved by varying the F value.

Load balancing enables service providers to improve network utilization. This can be achieved with the MOMP routing algorithm. The tradeoff is a marginal increase in the total bandwidth requirement. Thus the new algorithm would be useful for the implementation of VPN services in the hose model.

Chapter 7 Conclusions and Recommendations

7.1 Conclusions

This thesis investigates provisioning VPN services in the hose model, which combines the cost-effectiveness of VPN and the flexibility of the hose model. Previous studies on this topic are confined with providing bandwidth guarantees; this thesis focuses on some important issues vital for the successful deployment of hose model VPNs.

After reviewing related literature in Chapter 2, Chapter 3 studies and compares the blocking performance of the pipe model and the hose model for bandwidth-guaranteed connections. The study employs both analytical modeling and discrete-event simulations. The results show that the hose model has a blocking performance usually better than the pipe model by several orders of magnitude under normal network operating loads; and the analytical results are in close agreement with the simulation results. This part of study consolidates the superior features of the hose model, providing strong support for the hose model to be the preferred resource management model for VPN service provisioning.

Chapter 4 investigates the bandwidth efficiency and blocking performance of tree routing and multi-path routing for the hose model. The first part of Chapter 4 studies bandwidth efficiency of the hose model with tree routing and multi-path routing on a wide range of randomly generated network topologies and hose parameters. The network topologies are of reasonable sizes, having 15 to 50 nodes. This bandwidth efficiency study has complemented previous similar investigations dealing with only a

small number of nodes. A newly introduced maximum fraction of traffic on a path (MFTP) constraint effectively guarantees the availability of multiple paths for every VPN endpoint pair. Without the MFTP constraint, multi-path routing most often turns out to be single-path routing or tree routing; it only delivers small bandwidth savings at rare combinations of network topologies and hose parameters.

The blocking performance study is under a new concept of sub-provisioning to address overprovisioning problem of the hose model. The sub-provisioning is defined as providing bandwidth capacity that is of a certain percentage of the full provisioning based on the b^+ and b^- bounds; it is more reliable and easier to implement than the previously proposed resizing technique. The results show that, given a certain percentage of sub-provisioning, multi-path routing with fixed-alternate routing can significantly improves the blocking performance over tree routing; and for a fixed blocking probability, multi-path routing needs less bandwidth provisioning. A new path aggregation algorithm is also proposed to constrain the number of paths between a VPN endpoint pair within a manageable range. This part of work establishes multi-path routing as the preferred routing scheme for the hose model.

Chapter 5 proposes two multi-path routing heuristics for providing availability guaranteed services in hose model VPNs, the availability-centric approach and the bandwidth-centric approach. The two heuristics both employ multiple link-disjoint paths for each VPN endpoint pair, so that a backup path can be provided to increase the joint availability. They differ in the way to find those paths. Simulation results show that the proposed heuristics greatly increases the joint availabilities between each VPN endpoint pair. As a result, the blocking probabilities for provisioning

availability guaranteed connections are lowered by several orders of magnitude when compared to tree routing and single path routing. The two heuristic algorithms have their respective advantages, and can be used to suite different needs accordingly. This part of the study has enabled hose model VPNs to provide availability guaranteed service in addition to bandwidth guaranteed service.

Chapter 6 investigates a special case of traffic engineering, load balancing, in hose model VPNs. A novel multi-objective multi-path (MOMP) routing linear program with the MFTP constraint is proposed. The MOMP LP aims at minimizing the bandwidth reservation on the most loaded link in the service provider network to avoid congestion problems; the MFTP constraint enforces the employment of multiple alternative paths between each VPN endpoint pair, allowing the possibility to do load balancing within the VPN. The MFTP constraint also helps to enforce the load balancing effect in the service provider network. The proposed MOMP LP is shown to be able to reduce the bandwidth reservation on the most loaded link by as much as 50% compared to tree routing or single path routing. This part of the study has incorporated traffic engineering features into hose model VPN service provisioning.

In summary, this thesis has presented a comprehensive study on VPN service provisioning in the hose model. Firstly, it shows the hose model to be the preferred resource management model for VPN service provisioning. Secondly, the thesis provides strong support for multi-path routing to be the favorable routing scheme for the hose model. It complements previous works by presenting a thorough bandwidth efficiency study between multi-path routing and other routing schemes for the hose model. In addition, it has shown multi-path routing could be used to deal with the

overprovisioning problem associated with the hose model. Thirdly, the thesis has enhanced hose model VPNs to provide availability guaranteed service with multi-path routing, where tree routing and single path routing are unable to perform. Lastly, it has incorporated load balancing features into multi-path routing. This thesis has extended and contributed to the knowledge of the hose model with both sound theoretical analysis and extensive simulations. It helps to make the study more complete. The results presented in this thesis will help to boost service providers' interest in provisioning VPNs with the hose model, as well as to raise the customers' confidence in taking such services.

7.2 Recommendation for Future Work

Three possible directions are identified for potential future work:

- New link-disjoint path selection and path aggregation strategy. In the availability study part, the link-disjoint path selection and path aggregation are done in an endpoint pair scope. Aggregating paths will disturb the optimality of the multi-path routing LP, which will cause an increase in the total bandwidth capacity. Though the current mechanism works, it may not be the best one. Different strategies considering global VPN state information (e.g., considering all the VPN endpoint pairs altogether) in selecting the link-disjoint paths and aggregating other paths are likely to do less disturbance to the optimality of the multi-path routing LP, and thus cause less bandwidth capacity increase. In addition, the possible new mechanisms will still preserve other advantages, such as a small number of paths between each VPN endpoint pair.
- How to embed multiple VPNs in a single public network infrastructure. This topic makes practical significance. For example, optical VPNs may be used to support multiple higher layer transport technologies, such as ATM and IP, so

there is the drive to efficiently embed multiple VPNs into a single optical backbone network. At the first stage, individual VPNs specified in the pipe model will be considered for its simplicity. After gaining some insights into this problem, at the second stage, individual VPNs specified in the hose model will be attempted. For optical networks, the resources, such as wavelengths, transmitters and receivers, are in discrete form. This will render the problem a large-scale combinatorial optimization problem. In case Integer Linear Program (ILP) fails to work, Genetic Algorithm (GA) [110] and Simulated Annealing (SA) [111] may appear to be the viable techniques.

- How to embed hose model VPNs with multi-path routing (the virtual topology) into an underlying physical optical network using survivable routing, i.e., failure of any physical link would not disconnect the logical topology [63]. Because there is only one path between every endpoint pair with tree routing and single-path routing, so that a single link failure will inevitably render the VPN disconnected. Even multi-path routing does not provide survivability guarantee automatically. Consider the case for Optical VPNs. The VPN actually is a virtual topology comprising lightpaths, which are routed through the underlying optical network. Suppose there are two lightpaths emanating from an endpoint. If the two lightpaths are routed on the same fiber link, then the endpoint will still be disconnected from the VPN when the fiber link fails. So survivability necessitates multi-path routing and survivable routing of lightpaths emanating from an endpoint, so that the failure of any physical link leaves the (logical) network connected [66].

Author's Publications

Published Journal and Conference Papers:

1. H. Wang and G.-S. Poo, "Blocking performance of the hose model and the pipe model for VPN service provisioning over WDM optical networks," *Journal of Optical Networking*, vol. 3, pp.623-635, Jul. 2004.
2. H. Wang and G.-S. Poo, "Availability guaranteed service provisioning in hose model VPNs with multi-path routing," in proceedings of *IEEE International Conference on Communications (ICC)*, vol. 3. pp.1014-1019, Jun. 2006, Istanbul, Turkey.
3. G.-S. Poo and H. Wang, "Multi-path routing versus tree routing for VPN bandwidth provisioning in the hose model," *Computer Networks*, vol. 51, pp. 1725-1743, 2007.
4. H. Wang and G.-S. Poo, "Load balancing in the provisioning of hose model VPNs with multi-path routing," Accepted for publishing by *IEE Proc. Communications*.

Submitted Journal and Conference papers:

1. H. Wang and G.-S. Poo, "Reliability guaranteed service provisioning in hose model VPNs with multi-path routing," under preparation for submission.

Bibliography

- [1] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan and J. E. van der Merwe, "A flexible model for resource management in virtual private networks," in proceedings of *ACM SIGCOMM*, 1998.
- [2] B. Davie and Y. Rekhter, *MPLS Technology and Applications*. San Francisco, CA: Morgan Kaufmann Publisher, 2000.
- [3] D. Fowler, *Virtual Private Networks*. San Francisco, CA: Morgan Kaufmann Publishers, 1999.
- [4] E. Rosen, A. Viswanathan and R. Callon "Multiprotocol Label Switching Architecture," IETF RFC 3031, 2001
- [5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss "An architecture for differentiated services," IETF RFC 2475, 1998
- [6] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin "Resource reSerVation Protocol (RSVP)," IETF RFC 2205, 1997
- [7] S. Kent and R. Atkinson "Security architecture for the Internet protocol," IETF RFC 2401, 1998
- [8] A. Kumar, R. Rastogi, A. Silberschatz and B. Yener, "Algorithms for provisioning virtual private networks in the hose model," *IEEE/ACM Transactions on Networking*, vol. 10, pp. 565-578, 2002.
- [9] A. Gupta, A. Kumar and T. Roughgarden, "Simpler and Better approximation algorithms for network design," in proceedings of *the 35th Annual ACM Symposium on Theory of Computing (STOC)*, 2003.
- [10] T. Erlebach and M. Ruegg, "Optimal bandwidth reservation in hose-model VPNs with multi-path routing," in proceedings of *INFOCOM*, 2004.
- [11] A. Jüttner, I. Szabó and Á. Szentesi, "On bandwidth efficiency of the hose resource management model in virtual private networks," in proceedings of *IEEE INFOCOM*, 2003.
- [12] CPLEX, ILOG CPLEX Division, <http://www.cplex.com/>
- [13] OPNET, OPNET Technologies, <http://www.opnet.com/>
- [14] Y. Maeda, "Standards for virtual private networks," *IEEE Communications Magazine*, vol. 42, pp. 114-115, 2004.
- [15] P. Knight and C. Lewis, "Layer 2 and 3 virtual private networks: taxonomy, technology, and standardization efforts," *IEEE Communications Magazine*, vol. 42, pp. 124-131, 2004.
- [16] M. Carugi and J. De Clercq, "Virtual private network services: scenarios, requirements and architectural constructs from a standardization perspective," *IEEE Communications Magazine*, vol. 42, pp. 116-122, 2004.
- [17] E. Rosen and Y. Rekhter "BGP/MPLS IP VPNs," draft-ietf-l3vpn-rfc2547bis-03.txt, IETF Internet draft, 2004
- [18] P. Knight, H. Ould-Brahim and B. Gleeson "Network based IP VPN architecture using virtual routers," draft-ietf-l3vpn-vpn-vr-02.txt, IETF Internet draft, work in progress, 2004
- [19] C. Metz, "The latest in virtual private networks: part I," *IEEE Internet Computing*, vol. 7, pp. 87-91, 2003.
- [20] C. Metz, "The latest in VPNs: part II," *IEEE Internet Computing*, vol. 8, pp. 60-65, 2004.

- [21] G. Rosenbaum, W. Lau and S. Jha, "Recent directions in virtual private network solutions," in proceedings of *the 11th IEEE International Conference on Networks (ICON)*. 2003.
- [22] Y. Rekhter and T. Li "A Border Gateway Protocol 4 (BGP-4)," IETF RFC 1771, 1995
- [23] T. Bates, Y. Rekhter, R. Chandra and D. Katz "Multiprotocol Extensions for BGP-4," IETF RFC 2858, 2000
- [24] G. Malkin "RIP version 2," IETF RFC 2453, 1998
- [25] J. Moy "OSPF Version 2," IETF RFC 2328, 1998
- [26] B. Mukherjee, *Optical Communication Networks*: McGraw-Hill, 1997.
- [27] K. M. Sivalingam and S. Subramaiaam, *Optical WDM Networks: Principles and Practice*. Boston: Kluwer, 2000.
- [28] E. Mannie "Generalized Multi-protocol Label Switching (GMPLS) Architecture," IETF RFC 3945, 2004
- [29] R. Dutta and G. N. Rouskas, "Traffic grooming in WDM networks: past and future," *IEEE Network*, vol. 16, pp. 46-56, 2002.
- [30] E. Modiano, "Traffic grooming in WDM networks," *IEEE Communications Magazine*, vol. 39, pp. 124-129, 2001.
- [31] H. Zhu, H. Zang, K. Zhu and B. Mukherjee, "A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 285-299, 2003.
- [32] K. Zhu and B. Mukherjee, "Traffic grooming in an optical WDM mesh network," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 122-133, 2002.
- [33] ITU-T Recommendation Y.1312, "Layer 1 virtual private network generic requirements and architecture elements," 2003
- [34] ITU-T Recommendation Y.1313, "Layer 1 virtual private network service and network architectures," 2004
- [35] T. Takeda "Framework and requirements for layer 1 virtual private networks," draft-takeda-l1vpn-framework-04.txt, IETF Internet draft, work in progress, 2005
- [36] T. Takeda, I. Inoue, R. Aubin and M. Carugi, "Layer 1 virtual private networks: service concepts, architecture requirements, and related advances in standardization," *IEEE Communications Magazine*, vol. 42, pp. 132-138, 2004.
- [37] S. French and D. Pendarakis, "Optical Virtual Private Networks: Applications, Functionality and Implementation," *Photonic Network Communications*, vol. 7, pp. 227-238, 2004.
- [38] Z. Zhang, Y.-Q. Zhang, X. Chu and B. Li, "An Overview of Virtual Private Network (VPN): IP VPN and Optical VPN," *Photonic Network Communications*, vol. 7, pp. 213-225, 2004.
- [39] T. Takeda, D. Brungard, D. Papadimitriou and H. Ould-Brahim, "Layer 1 virtual private networks: driving forces and realization by GMPLS," *IEEE Communications Magazine*, vol. 43, pp. 60-67, 2005.
- [40] L. Berger "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions," IETF RFC 3473, 2003
- [41] R. Cohen and G. Kaempfer, "On the cost of virtual private networks," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 775-784, 2000.

- [42] H. Liang, O. Kabranov, D. Makrakis and L. Orozco-Barbosa, "Minimal cost design of virtual private networks," in proceedings of *Canadian Conference on Electrical and Computer Engineering(CCECE)*, 2002.
- [43] S. Raghunath, S. Kalyanaraman and K. K. Ramakrishnan, "Trade-offs in resource management for virtual private networks," in proceedings of *IEEE INFOCOM*, 2005.
- [44] J. Zeng and N. Ansari, "Toward IP virtual private network quality of service: a service provider perspective," *IEEE Communications Magazine*, vol. 41, pp. 113-119, 2003.
- [45] T. Braun, M. Guenter and I. Khalil, "Management of quality of service enabled VPNs," *IEEE Communications Magazine*, vol. 39, pp. 90-98, 2001.
- [46] L. Zhang and L. Zheng, "IPv6 traffic with multi-class QoS in VPN," *Journal of Computer Network*, vol. 37, pp. 263-275, 2001.
- [47] R. Garg and H. Saran, "Fair bandwidth sharing among virtual networks: a capacity resizing approach," in proceedings of *IEEE INFOCOM*, 2000.
- [48] R. Issacs and I. Lesile, "Support for resource-assured and dynamic virtual private networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 460-472, 2001.
- [49] I. Khalil and T. Braun, "Edge provisioning and fairness in VPN-DiffServ networks," *Journal of Network and Systems Management*, vol. 10, pp. 11-37, 2002.
- [50] P. Hegyi, M. Maliosz, A. Ladanyi and T. Cinkler, "Shared protection of virtual private networks with traffic concentration," in proceedings of *4th International Workshop on Design of Reliable Communication Networks (DRCN)*, 2003.
- [51] J. Han, G. R. Malan and F. Jahanian, "Fault-tolerant virtual private networks within an autonomous system," in proceedings of *21st IEEE Symposium on Reliable Distributed Systems*, 2002.
- [52] A. Zhao, Y. Yuan, Y. Ji and G. Gu, "Research on tunneling techniques in virtual private networks," in proceedings of *International Conference on Communication Technology*, 2000.
- [53] Y. Yang, C. U. Martel and S. F. Wu, "On building the minimum number of tunnels: an ordered-split approach to manage IPSec/VPN policies," in proceedings of *IEEE/IFIP Network Operations and Management Symposium(NOMS)*, 2004.
- [54] I.-W. Chen, Y.-D. Lin and Y.-N. Lin, "Tunnel minimization and relay for managing virtual private networks," in proceedings of *IEEE Global Telecommunications Conference(GLOBECOM)*, 2004.
- [55] S. Al-Khayatt, S. A. Shaikh, B. Akhgar and J. Siddiqi, "A study of encrypted, tunneling models in virtual private networks," in proceedings of *International Conference on Information Technology: Coding and Computing*, 2002.
- [56] T. Takeda, H. Kojima and I. Inoue, "Optical VPN architecture and mechanisms," in proceedings of *the 9th Asia-Pacific Conference on Communications (APCC)*, 2003.
- [57] T. Takeda, H. Kojima, N. Matsuura and I. Inoue, "Resource allocation method for optical VPN," in proceedings of *Optical Fiber Communications (OFC)*, 2003.
- [58] J. Zheng, B. Zhou and H. T. Mouftah, "Virtual topology design and reconfiguration of virtual private networks (VPNs) over all-optical WDM Networks," *Photonic Network Communications*, vol. 7, pp. 255-266, 2004.

- [59] C.-J. Chae and R. S. Tucker, "Implementation of multiple optical virtual private networks over WDM passive optical network," *Electronics Letters*, vol. 40, pp. 331-333, 2004.
- [60] Y. Qin, K. Sivalingam and B. Li, "Architecture and analysis for providing virtual private networks (VPNs) with QoS over optical WDM network," *Optical Networks Magazine*, vol. 2, pp. 57-65, 2001.
- [61] Y. Qin, B. Li, W. L. Lim and S. H. Yeo, "A design for on-line virtual private networks (VPN) over optical WDM networks," in proceedings of *Global Telecommunications Conference (GLOBECOM)*, 2002.
- [62] Y. Qin and C. K. Siew, "Framework for dynamic optical virtual private networks (VPNs): architecture and analysis," *IEE Proceedings-Communications*, vol. 151, pp. 71-76, 2004.
- [63] D. Zhou and S. Subramaniam, "Survivability in optical networks," *IEEE Network*, vol. 14, pp. 16-23, 2000.
- [64] C. A. Al Sayeed and O. W. W. Yang, "Survivability of optical virtual private network using the load distributive shortest pair algorithm," in proceedings of *IEEE Pacific Rim Conference on Communications, Computers and signal Processing (PACRIM)*, 2005.
- [65] A. Haque and P.-H. Ho, "Survivable routing and wavelength assignment (RWA) in optical virtual private networks (O-VPNs)," in proceedings of *Workshop on High Performance Switching and Routing(HPSR)*, 2005.
- [66] E. Modiano and A. Narula-Tam, "Survivable lightpath routing: a new approach to the design of WDM-based networks," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 800-809, 2002.
- [67] C. Prommak and D. Tipper, "Load distribution-survivable lightpath routing for the optical virtual private network," in proceedings of *IEEE Workshop on High Performance Switching and Routing (HPSR)*, 2002.
- [68] C. Vijaya Saradhi and C. S. Ram Murthy, "A Framework for Differentiated Survivable Optical Virtual Private Networks," *Photonic Network Communications*, vol. 4, pp. 457-487, 2002.
- [69] C. Antal, J. Harmatos, A. Jütner and G. Toth, "A cluster-based resource provisioning model in virtual private networks," in proceedings of *the 5th International Workshop on Design of Reliable Communication Networks(DRCN)*, 2005.
- [70] T. W. Chim, K.-S. Lui, K. L. Yeung and C. P. Wong, "Routing algorithm for provisioning symmetric virtual private networks in the hose model," in proceedings of *IEEE Global Telecommunications Conference(GLOBECOM)*, 2005.
- [71] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi and B. Yener, "Provisioning a virtual private network: A network design problem for multi-commodity flow," in proceedings of *the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, 2001.
- [72] G. Italiano, S. Leonardi and G. Oriolo, "Design of networks in the hose model," in proceedings of *the 3rd Workshop on Approximation and Randomization Algorithms in Communication Networks (ARACNE)*, 2002.
- [73] J. Chu and C.-T. Lea, "Routing and restoration in networks with hose-model traffic patterns," in proceedings of *Workshop on High Performance Switching and Routing (HPSR)*, 2005.

- [74] Y.-L. Liu, Y. S. Sun and M. C. Chen, "Traffic engineering for hose-model VPN provisioning," in proceedings of *IEEE Global Telecommunications Conference(GLOBECOM)*, 2005.
- [75] Y.-L. Liu, Y. S. Sun and M. C. Chen, "On the On-Line Hose-Model VPN Provisioning," in proceedings of *the Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/ELearning on Telecommunications Workshop*, 2005.
- [76] G. Italiano, R. Rastogi and B. Yener, "Restoration algorithms for virtual private networks in the hose model," in proceedings of *IEEE INFOCOM*, 2002.
- [77] A. Balasubramanian and G. Sasaki, "Bandwidth requirements for protected vpns in the hose model," in proceedings of *IEEE International Symposium on Information Theory*, 2003.
- [78] L. Zhang, J. Muppala and S. Chanson, "Provisioning virtual private networks in the hose model with delay requirements," in proceedings of *International Conference on Parallel Processing*, 2005.
- [79] D. Wei and N. Ansari, "Implementing fair bandwidth allocation schemes in hose-modelled VPN," *IEE Proceedings-Communications*, vol. 151, pp. 521-528, 2004.
- [80] F. P. Kelly, "Routing and capacity allocation in networks with trunk reservation," *Mathematics of Operations Research*, vol. 15, pp. 771-792, 1990.
- [81] S.-P. Chung and K. W. Ross, "Reduced load approximations for multirate loss networks," *IEEE Transactions on Communications*, vol. 41, pp. 1222-1231, 1993.
- [82] A. G. Greenberg and R. Srikant, "Computational techniques for accurate performance evaluation of multirate, multihop communication networks," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 266-290, 1997.
- [83] S. Ramamurthy and B. Mukherjee, "Survivable WDM mesh networks. Part I- Protection," in proceedings of *IEEE INFOCOM '99*, 1999.
- [84] J. Manchester, P. Bonenfant and C. Newton, "The evolution of transport network survivability," *IEEE Communications Magazine*, vol. 37, pp. 44-51, 1999.
- [85] D. Johnson, N. Hayman and P. Veitch, "The evolution of a reliable transport network," *IEEE Communications Magazine*, vol. 37, pp. 52-57, 1999.
- [86] Y. Seok, Y. Lee, Y. Choi and C. Kim, "Dynamic constrained multipath routing for MPLS networks," in proceedings of *10th International Conference on Computer Communications and Networks(ICCNC)*, 2001.
- [87] S. Rai, O. Deshpande, C. Ou and B. Mukherjee, "Reliable multi-path provisioning for high-capacity optical backbone mesh networks," in proceedings of *IEEE International Conference on Communications(ICC)*, 2005.
- [88] A. Chakrabarti and G. Manimaran, "Reliability constrained routing in QoS networks," *IEEE/ACM Transactions on Networking*, vol. 13, pp. 662-675, 2005.
- [89] Y. Huang, J. P. Heritage, B. Mukherjee and W. Wen, "Availability-guaranteed service provisioning with shared-path protection in optical WDM networks," in proceedings of *Optical Fiber Communication Conference (OFC)*, 2004.

- [90] M. Tornatore, G. Maier and A. Pattavina, "Availability Design of Optical Transport Networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 1520-1532, 2005.
- [91] E. Dinan, D. O. Awduche and B. Jabbari, "Analytical framework for dynamic traffic partitioning in MPLS networks," in proceedings of *IEEE International Conference on Communications(ICC)*, 2000.
- [92] A. Elwalid, C. Jin, S. Low and I. Widjaja, "MATE: MPLS adaptive traffic engineering," in proceedings of *IEEE INFOCOM*, 2001.
- [93] Z. Cao, Z. Wang and E. Zegura, "Performance of hashing-based schemes for Internet load balancing," in proceedings of *IEEE INFOCOM*, 2000.
- [94] J. Song, S. Kim, M. Lee, H. Lee and T. Suda, "Adaptive load distribution over multipath in MPLS networks," in proceedings of *IEEE International Conference on Communications(ICC)*, 2003.
- [95] Z. Zhao, Y. Shu, L. Zhang, H. Wang and O. W. W. Yang, "Flow-level multipath load balancing in MPLS network," in proceedings of *IEEE International Conference on Communications (ICC)*, 2004.
- [96] Y. Wang and Z. Wang, "Explicit routing algorithms for Internet traffic engineering," in proceedings of the *Eight International Conference on Computer Communications and Networks (ICCCN)*, 1999.
- [97] Y. Lee, Y. Seok, Y. Choi and C. Kim, "A constrained multipath traffic engineering scheme for MPLS networks," in proceedings of *IEEE International Conference on Communications(ICC)*, 2002.
- [98] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization : algorithms and complexity*. Englewood Cliffs, N.J.: Prentice Hall, 1982.
- [99] K. Kar, M. Kodialam and T. V. Lakshman, "Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 2566-2579, 2000.
- [100] C. Xin and C. Qiao, "Performance analysis of multihop traffic grooming in mesh WDM optical networks," in proceedings of *The 12th International Conference on Computer Communications and Networks (ICCCN)*, 2003.
- [101] L. Brouwer, "Über Abbildung von Mannigfaltigkeiten," *Mathematische Annalen*, vol. 71, pp. 97-115, 1911.
- [102] F. P. Kelly, "Loss networks," *Annals of Applied Probability*, vol. 1, pp. 319-378, 1991.
- [103] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, *Network flows : theory, algorithms, and applications*. Englewood Cliffs, N.J.: Prentice Hall, 1993.
- [104] T. G. Robertazzi, *Computer networks and systems : queueing theory and performance evaluation*, 3rd ed. New York: Springer, 2000.
- [105] J. Kaufman, "Blocking in a shared resource environment," *IEEE Transactions on Communications*, vol. 29, pp. 1474-1481, 1981.
- [106] J. W. Roberts, "A service system with heterogeneous user requirements," in proceedings of *Performance of Data Communication System and Their Applications*, 1981.
- [107] B. M. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1617-1622, 1988.
- [108] K. S. Trivedi, *Probability and statistics with reliability, queueing, and computer science applications*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
- [109] L. Andersson, P. Doolan, N. Feldman, A. Fredette and B. Thomas "LDP specification," IETF RFC 3036, 2001

- [110] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Reading, Mass.: Addison-Wesley Pub. Co., 1989.
- [111] P. J. M. v. Laarhoven and E. H. L. Aarts, *Simulated annealing : theory and applications*. Dordrecht, the Netherlands: D. Reidel Publishing Company, 1987.