Helsinki University of Technology Networking Laboratory
Teknillinen korkeakoulu Tietoverkkolaboratorio
Espoo 2004 Report 5/2004

RESOURCE ALLOCATION AND PERFORMANCE ANALYSIS PROBLEMS IN OPTICAL NETWORKS

Esa Hyytiä

Dissertation for degree of Doctor of Science in Technology with due permission for public examination and debate in Auditorium S4 at Helsinki University of Technology (Espoo, Finland) on the 10th of December, 2004, at 12 o'clock noon.

Helsinki University of Technology
Department of Electrical and Communications Engineering
Networking Laboratory

Teknillinen korkeakoulu
Sähkö- ja tietoliikennetekniikan osasto
Tietoverkkolaboratorio

Distribution:
Helsinki University of Technology
Networking Laboratory
P.O.Box 3000
FIN-02015 TKK
Tel. +358-9-451 2461
Fax. +358-9-451 2474

ISBN 951-22-7398-5
ISSN 1458-0322

Picaset Oy
Espoo 2004

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY

HELSINKI UNIVERSITY OF TECHNOLOGY
P.O.BOX 1000, FIN-02015 TKK
http://www.hut.fi/

ABSTRACT OF DOCTORAL DISSERTATION

Author: Esa Hyytiä

Name of the dissertation
Resource Allocation and Performance Analysis Problems in Optical Networks

Date of manuscript	10th of November	Date of the dissertation	10th December 2004
☐ Monograph		☒ Article dissertation (summary + original articles)	

Department	Department of Electrical and Communications Engineering
Laboratory	Networking Laboratory
Field of Research	teletraffic theory, performance evaluation of optical communications networks
Opponent	Professor Ezhan Karasan (Bilkent University, Ankara, Turkey)
Supervisor	Professor Jorma Virtamo (Helsinki University of Technology)

Abstract

Optical networks pose a rich variety of new design and performance analysis problems. Typically, the static design problems belong to the field of combinatorial optimisation, whereas decision-making and performance analysis problems are best treated using appropriate stochastic models. This dissertation focuses on certain issues in resource allocation and performance evaluation of backbone wavelength-routed (WR) networks and metropolitan area optical burst switching (OBS) networks.

The first two parts of the thesis consider heuristic algorithms for the static routing and wavelength assignment (RWA) and logical topology design (LTD) problems that arise in the context of WR networks. In a static RWA problem, one is asked to establish a given set of lightpaths (or light trees) in an optical WR network with given constraints, where the objective often is to minimise the number of wavelength channels required. In LTD problem, the number of wavelength channels is given and one is asked to decide on the set of lightpaths so that, for instance, the mean sojourn time of packets travelling at the logical layer is minimised. In the thesis, several heuristic algorithms for both the RWA and LTD problems are described and numerical results are presented.

The third part of the thesis studies the dynamic control problem where connection requests, i.e. lightpath requests, arrive according to a certain traffic pattern and the task is to establish one lightpath at a time in the WR optical network so that the expected revenue is maximised or the expected cost is minimised. Typically, the goal of optimisation is to minimise some infinite time horizon cost function, such as the blocking probability. In this thesis, the dynamic RWA problem is studied in the framework of Markov decision processes (MDP). An algorithmic approach is proposed by which any given heuristic algorithm can be improved by applying the so-called first policy iteration (FPI) step of the MDP theory. Relative costs of states needed in FPI are estimated by on-the-fly simulations. The computational burden of the approach is alleviated by introducing the importance sampling (IS) technique with FPI, for which an adaptive algorithm is proposed for adjusting the optimal IS parameters at the same time as data are collected for the decision-making analysis.

The last part of the thesis considers OBS networks, which represent an intermediate step towards full optical packet switching networks. In OBS networks, the data are transferred using optical bursts consisting of several IP packets going to the same destination. On the route of the burst, temporary reservations are made only for the time during which the burst is transmitted. This thesis focuses on fairness issues in OBS networks. It is demonstrated that fairness can be improved by using fibre delay lines together with Just-Enough-Time protocol (JET). Furthermore, by choosing the routes in an appropriate way one can also reach a satisfactory level of fairness and, at the same time, lower the overall blocking probability. Possible scheduling policies for metropolitan area OBS ring networks are also studied.

Keywords	optical networks, WDM, routing and wavelength assignment, Markov decision process, policy iteration, optical burst switching		
UDC	621.39:621.372.8:004.7:518.857	Number of pages	106 pp. + app. 103 pp.
ISBN (printed)	951-22-7398-5	ISBN (pdf)	951-22-7405-1
ISBN (others)		ISSN	1458-0322
Publisher	Networking Laboratory / Helsinki University of Technology		
Print distribution			

☒ The dissertation can be read at http://lib.hut.fi/Diss/2004/isbn9512274051/

PREFACE

In January 1996, as an undergraduate student, I was walking in the corridors of the main building of the Helsinki University of Technology when I noticed an announcement on an noticeboard about a queueing theory course. I had no prior knowledge about the topic but as it seemed interesting and I had some free time in my schedule I decided to take the course. The next year I found myself working as an assistant of the same course. The course was led by Professor Jorma Virtamo, who since then has become my supervisor.

This dissertation is a result of work that started in 1997, when I started to work at the Laboratory of Telecommunications as a research assistant in the COST 257 project, funded by Nokia, Sonera, and Tekes. The laboratory was later renamed as the Networking Laboratory in order to reflect the change in the focus of the laboratory's research. Soon after I completed my Master's thesis, the Optical Access Networking project, funded by Nokia, Elisa, and Tekes was started, and this provided me with a chance to continue with my studies. For the last year I have been working in the FIT project, funded by the Academy of Finland. Additionally, I have received financial support in the form of scholarships from the Nokia Foundation, TAES Foundation, and Heikki and Hilma Honkanen Foundation.

Throughout my studies I have had the privilege of working with my supervisor Professor Jorma Virtamo. His valuable insight into different topics and numerous discussions with him have made this thesis possible and I indeed owe him gratitude. In addition, I would like to thank M.Sc. Laura Nieminen for her efforts as a co-author of publications. Additionally, the whole staff of the Networking Laboratory deserve special thanks for providing a good atmosphere and facilities for the research.

Espoo, November, 2004

Esa Hyytiä

CONTENTS

LIST OF PUBLICATIONS

[1] Esa Hyytiä. Heuristic algorithms for the generalized routing and wavelength assignment problem. In *Seventeenth Nordic Teletraffic Seminar, NTS-17*, Fornebu, Norway, August 2004.

[2] Esa Hyytiä. Maximization of single hop traffic with greedy heuristics. In *the 3rd International Conference on Wireless and Optical Communications, WOC*, pages 289–294, Banff, Alberta, Canada, July 2003.

[3] Esa Hyytiä and Jorma Virtamo. Dynamic Routing and Wavelength Assignment Using First Policy Iteration. In *the Fifth IEEE Symposium on Computers and Communications, ISCC'2000*, pages 146–151, Antibes, Juan les Pins, France, July 2000.

[4] Esa Hyytiä and Jorma Virtamo. Dynamic Routing and Wavelength Assignment Using First Policy Iteration, Inhomogeneous Traffic Case. In *the International Conference on Performance and QoS of Next Generation Networking, P&QNet2000*, pages 301–316, Nagoya, Japan, November 2000.

[5] Esa Hyytiä and Jorma Virtamo. Optimal importance sampling in Markov process simulation. In *International Conference on Applied Simulation and Modelling, ASM 2001*, pages 67–72, Marbella, Spain, September 2001.

[6] Esa Hyytiä and Jorma Virtamo. Adaptive importance sampling in routing and wavelength assignment. *European Transactions on Telecommunications (ETT)*, 13(4):331–339, July/August 2002. Special Issue on Rare Event Simulation.

[7] Laura Nieminen and Esa Hyytiä. Performance of the JET protocol with optical delay lines. In *Optical Network Design and Modelling, ONDM*, pages 769–784, Budapest, Hungary, February 2003.

[8] Esa Hyytiä and Laura Nieminen. Linear program formulation for routing problem in OBS networks. In *the Ninth IEEE Symposium on Computers and Communications, ISCC'2004*, pages 252–257, Alexandria, Egypt, June 2004.

[9] Esa Hyytiä and Laura Nieminen. Analytical loss models for MAC protocols in optical ring network operating under a static traffic load. In *2004 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04)*, pages 135–141, San Jose, California, July 2004.

ABBREVIATIONS

AIS	Adaptive Importance Sampling
ATM	Asynchronous Transfer Mode
CAT	constant assembly time
CoS	Class of Service
CWDM	Coarse Wavelength Division Multiplexing
DCA	Distinct Channel Assignment (constraint)
DQBR	Distributed Queue Bidirectional Ring
DWDM	Dense Wavelength Division Multiplexing
EDFA	Erbium Doped Fibre Amplifier
FCFS	First Come First Served
FDL	Fibre Delay Line
FPI	First Policy Iteration
FRP	Fast Reservation Protocol
GA	Genetic Algorithm
GMPLS	Generalised Multi Protocol Label Switching
GRWA	Generalised Routing and Wavelength Assignment
IETF	Internet Engineering Task Force
ILP	Integer Linear Programming
IP	Internet Protocol
IS	Importance Sampling
ITU	International Telecommunication Union
LP	Linear Programming (problem)
LRN	Logically-Routed Network
LSN	Logical Switching Node
LT	Logical Topology
LTD	Logical Topology Design
MAC	Media Access Control
MAN	Metropolitan Area Network
MDP	Markov Decision Process
MILP	Mixed Integer Linear Programming
MPLS	Multi Protocol Label Switching
OBS	Optical Burst Switching
OBSN	Optical Burst Switched Network
ODD	Only Destination Delay (protocol)
OEO	Optical-Electronic-Optical
ONN	Optical Network Node
OPS	Optical Packet Switching
OPSN	Optical Packet Switched Network
RWA	Routing and Wavelength Assignment
SA	Simulated Annealing
SDH	Synchronous Digital Hierarchy
SMDP	Semi-Markov Decision Process

SONET	Synchronous Optical Network
TD	Topology Definition
TR	Traffic Routing (in logical topology)
TS	Tabu Search
VAT	variable assembly time
WA	Wavelength Assignment
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing
WIXC	Wavelength Interchanging Cross-Connect
WR	Wavelength-Routed (network)
WRN	Wavelength-Routed Network
WSXC	Wavelength Selective Cross-Connect

1 INTRODUCTION

1.1 Wavelength Division Multiplexing, WDM

The rapid growth of Internet traffic has been the driving force for faster and more reliable data communication networks. The volume of packet traffic has been increasing at a much higher pace than traditional voice traffic and hence it has been predicted that in the future almost all traffic will be IP traffic. Wavelength division multiplexing (WDM) is a very promising technology to meet these ever-increasing demands. In a WDM network, several optical signals are transmitted on the same fibre using different wavelength channels. Traditionally only a small fraction of the fibre capacity has been in use, but by using WDM it is possible to exploit this huge capacity more efficiently. In fact, the capacity of WDM link can be as large as terabits per second in a single fibre. Furthermore, the possibility of using the existing fibres more efficiently makes WDM a commercially very attractive alternative, as it is often very expensive to install new fibres in the ground. This is especially the case in densely-populated areas such as cities, where fibres must be installed under streets etc. WDM technology has been recognised as one of the key components of the future networks and the commercialisation of WDM technology is progressing rapidly. There is also considerable interest in optical networking in the academic community, as it offers a rich research field for scientists, from the component level up to network protocols.

Optical Spectrum
WDM systems can be classified further on the basis of the channels used. The first WDM systems were so-called broadband WDM systems, using two widely-separated signals (typically at 1310 nm and 1550 nm). On the other hand, the term dense wavelength division multiplexing (DWDM) refers to a technology used in backbone networks, where up to 40 or 80 signals are combined in the same fibre. Furthermore, there is also so-called coarse wavelength division multiplexing (CWDM), where the channel spacing is 20nm in the range of 1270nm to 1610nm, giving up to 18 channels in total. Unlike the other two, CWDM is targeted at metropolitan area networks.

The International Telecommunication Union (ITU) has standardised the use of wavelength channels. Standard G.692 (see [ITU98]) defines the channel spacing for DWDM systems as 50 GHz or 100 GHz around the reference frequency of 193.10 THz, as depicted in Fig. 1.1. The reference frequency 193.10 THz corresponds to about 1550 nm, and hence the proposal is meant for the 1540 nm - 1560 nm pass band of the optical fibre.

All-Optical Networks
Initially, the WDM technique was used to increase the capacity of point-to-point optical links. At the end of each link, the signal is converted back to the electrical domain and the gain is simply a larger link capacity.

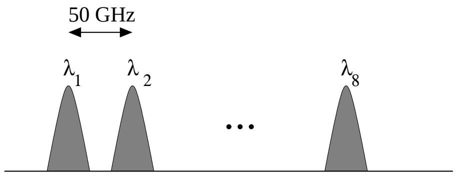

Figure 1.1: The optical spectrum and 8 wavelength channels.

However, the trend has been towards transparent all-optical networks, where the signal is routed through the network in the optical domain. Something that was especially important for the development of all-optical WDM networks was the invention of the optical fibre amplifier (Erbium doped fibre amplifier, EDFA) in 1987. The optical fibre amplifier is a component capable of amplifying several optical signals at the same time without converting them first to the electrical domain (opto-electronic amplification). It is also worth noting that EDFA can be used to amplify signals of different bit rates and modulations. Other important WDM components include, among others, lasers, receivers, wavelength division multiplexers, wavelength converters, optical splitters and tunable filters. Together these components enable us to build transparent all-optical networks to meet the ever increasing capacity demands that the future will bring.

1.2 IP over Optical

The telecommunication field has a variety of standards defining different layers for the whole infrastructure. In the past, the end users were people making phone calls or using fax machines etc. However, according to the current understanding, it seems that in the future almost all the traffic will be IP-based. The evolution will tend towards IP-over-WDM networks, for which several alternative approaches have been proposed in the literature [BRM00, GDW00, Dix03]. In Fig. 1.2, some of the possible layering alternatives are depicted. Consider first the IP over ATM over SONET/SDH over optical solution. Roughly speaking the role of each layer from bottom to top is as follows:

- Physical layer provides the optical fibres between the network nodes including possible optical fibre amplifiers.
- Optical layer provides transparent all-optical lightpaths between node pairs for a higher layer. Each physical link (or fibre) is capable of carrying several lightpaths using WDM technology, and each lightpath corresponds to an optical link for the SONET/SDH node.
- SONET/SDH layer provides constant bitrate transmission pipes from point A to point B over the SONET/SDH network. Furthermore, SONET/SDH network's protection and restoration capabilities can be used to ensure effectively uninterrupted bit flows.
- ATM layer can be used to provide virtual connections (VC) of arbitrary bitrate from point A to point B with different QoS parameters.

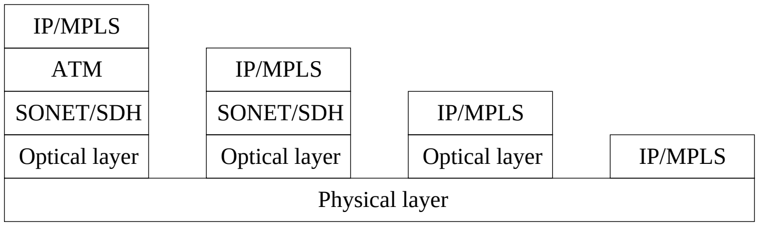

Figure 1.2: Possible alternatives for IP-over-WDM solutions.

ATM can be used for traffic engineering purposes, but it has become somewhat redundant as IP/MPLS routers tend to provide similar features.

- IP/MPLS layer only expects transmission links for IP packets.

Each additional layer naturally brings some extra overhead to the transmission. Hence, the typical IP over ATM over SONET/SDH over WDM mapping can be considered to be an inefficient solution.

Eventually the trend is towards IP-over-Optical solutions, where IP packets are transferred directly on the optical layer without any intermediate layer, i.e. IP/MPLS over WDM solution. One such proposal is a so-called λ-labelling presented in [Gha00]. Also in the IETF work is going on for standardising the so-called generalised multi protocol label switching (GMPLS), which is supposed to unify the management of the optical networks and allow interoperability between different manufacturers.

For completeness, Fig. 1.3 tries to illustrate the currently used solutions to carry IP traffic using an underlying WDM network.

1.3 Wavelength-Routed Networks

A wavelength-routed network (WRN) is an all-optical network, where the routing at the network nodes is based on the wavelength of the incoming signal [Muk97, RS98, Wil97, SB99, Dix03]. The optical connection between a node pair is referred to as a lightpath and the optimisation of a WRN consists of choosing a feasible route and wavelength for each light-

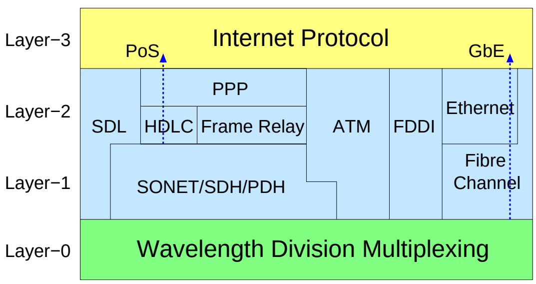

Figure 1.3: Different approaches for IP-over-Optical (WDM).

path. WRNs are very scalable[1] and can achieve a high degree of utilisation in an arbitrary mesh network. In other words, one can exploit the vast bandwidth of the optical fibre throughout the network more efficiently. A WRN also makes possible fast restoration schemes and modifications of the underlying network without the need for any reconfiguration of the upper layers. Hence the optical layer is used to build a so-called virtual/logical topology over the physical network for the logical layer (e.g. ATM or IP). The virtual topology can remain the same even if the physical network changes for some reason, such as a failure in some part of the network.

1.4 Logically-Routed Networks

The layered approach leads to the concept of logically-routed networks [SB99]. In a logically-routed network (LRN), a given logical topology is realised over a physical optical topology (see Fig. 1.4). Thus, transparent all-optical connections, i.e. lightpaths, are established in the physical network and logical switching nodes (LSN) see the logical topology instead of the physical one. In other words, the physical layer provides optical channels from point A to point B, and, from the logical layer point of view, it does not matter how the lightpath is routed on the physical layer. This mapping allows the physical network to be changed, e.g. in the event of some network failures, without changing the logical topology seen by the upper layers. Hence, it simplifies the description of the network to the upper layers. A typical LSN could be an ATM or IP switch.

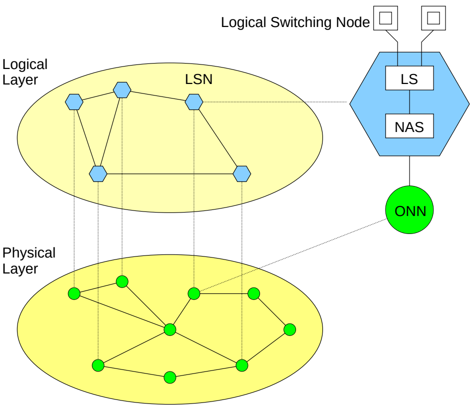

Figure 1.4: A logically-routed network (LRN) where a logical topology is built over a physical topology. The logical switching nodes (LSN) operate on a logical topology, e.g. ATM or SONET switches (adapted from [SB99]).

In LRN, the connections end users request can be created between any (end user) node pair, i.e. there is full connectivity. In the logical layer, the (ATM/IP) switch finds a feasible route via zero or more intermediate LSNs between the LSNs the end users are connected to. Similarly, the optical

[1] In comparison, broadcast and select optical networks do not scale well with the number of end systems.

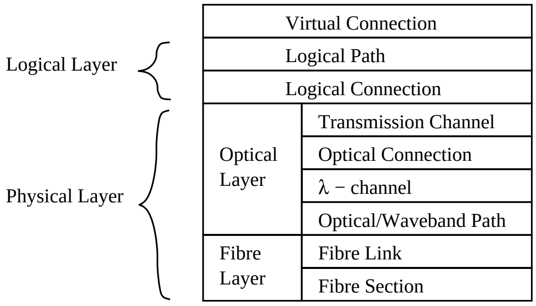

Figure 1.5: Layered architecture of the optical networks (adapted from [SB99]).

layer supports the optical connections for the logical layer. The layered architecture of the optical networks is depicted in Fig. 1.5 [SB99]. From bottom to up the layers are roughly as follows:

- Fibre layer corresponds to physical fibres between two ONNs. Each fibre link consists of one or more fibre sections.

- Optical connection, i.e. a lightpath consists of 1) a λ-channel (i.e. a wavelength channel) and 2) optical path. The λ-channel is a basic "transmission capacity unit" the optical layer can provide (typically 2.5-10 Gbit/s). Transmission channel sublayer in figure includes a conversion from logical signal (e.g. ATM cells, IP packets etc.) to transmission signal [SB99].

- Logical connection corresponds to a unidirectional connection between two external ports of network access stations (e.g. IP router).

- Logical path, on the other hand, consists of one or more consecutive logical connections, i.e. for example a route for an IP packet.

- Virtual connection corresponds to, e.g. a label switched path (LSP) in MPLS network, or a virtual path/connection in an ATM network.

For more details, see [SB99].

In practice, the bandwidths the end users require are far below that of the optical channel. Depending on the logical topology chosen, the end user's requirements can either be fulfilled or not. The limitations on the logical topology are set by the underlying physical network. Hence, on a generic level one can consider the problem where there are (time dependent) traffic flows between optical network nodes (ONN). These flows can be of any size, from a fraction of a single optical channel up to several channels. Once the logical topology is fixed, the logical layer is supposed to route the data flows using the established lightpaths. This means that data flows are aggregated at the optical layer, where they travel to the next LSN, which then demultiplexes them to different data streams for the next hop etc.

Generally, the capacity requirements for data flows (packet flows) are not integer multiples of the capacity of a single wavelength channel, but arbitrary multiples or fractions of that capacity. Furthermore, these flows can be aggregated at any node to a single flow and later split again at some intermediate node and then forwarded to other directions.

By a multihop network we mean a network where each data flow uses possibly more than one optical hop. This causes an extra processing load for the intermediate nodes and increases the delays packets experience, but makes possible more efficient use of the optical resources. The aggregation process corresponds to routing at the logical layer. Note that it is not usually practical to configure the network so that the logical and physical layers are topologically equivalent, because then the conversion between layers causes unnecessary delays to the traffic [RS96, MBRM96, ZA95].

The problem of deciding on both the lightpath establishment and the routing at the logical layer is often referred to as the logical topology design problem (LTD), while in [SB99] it is called the Multihop Network Configuration Problem. This kind of problem can be solved in a hierarchical way, as presented in Fig. 1.6. At the first step, current (average) data streams between the nodes are mapped to lightpath requests, i.e. the requirements for the logical topology are set (topology definition, TD). If there are enough resources available in the network, each lightpath request can be fulfilled and a feasible solution has been found.

A common approach is to first fix the logical topology and packet level routing, i.e. the TD and TR problems in Fig. 1.6. This step essentially defines the lightpath(s) each packet uses to travel through the network towards the destination node. Once this decision has been made, the problem is reduced to the establishment of lightpaths in the network (the third box from the top in Fig. 1.6).

In summary, the TD step defines a set of lightpath requests, i.e. by using the mean traffic flows between the node pairs (and possibly the knowledge about physical network) as an input it determines the number of lightpaths to be established between each node pair, which allows, in some sense, the most efficient transmission of data packets. The lightpath establishment step gets the lightpath requests (number of lightpaths to be established between each node pair) as an input and determines a feasible route and wavelength for each request.

The establishment of lightpaths in the network, i.e. the routing and wavelength assignment problem (RWA), is traditionally solved in one or two phases. By one-phase solution we mean an algorithm where both the route (for the lightpath) and its wavelength(s) are determined simultaneously. Alternatively the RWA problem can be further decomposed into lightpath routing (LR) and wavelength assignment (WA) steps and, consequently, by two-phase solution we mean an algorithm where the path is first fixed for each lightpath and then a feasible wavelength is assigned to each lightpath (see Section 2.5). Generally, shorter paths are usually good candidates. The optimisation of the physical network layer, i.e. the RWA problem, is discussed in detail in Chapter 2 and the LTD problem in Chapter 3.

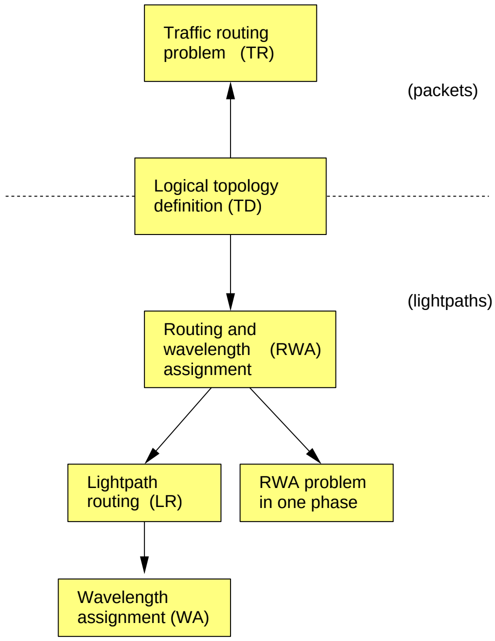

Figure 1.6: Hierarchical model of WDM network configuration.

1.6 Optical Packet Switching

In contrast to circuit-switched WDM networks, optical packet switching offers even more flexibility. A frequently-proposed idea is to build local area networks using optical packet switching (OPS). Photonic packet switching, however, involves many open questions [YMD00, PMMB00, BGD01]. The proposals for OPS can be divided into two categories, namely slotted and un-slotted. In the slotted solution, each packet has a constant length and the operation is synchronous, while in the un-slotted case packet lengths can vary and the operation is asynchronous. Generally, controlling the delay in photonic packet switched networks is an important issue.

In the electronic domain, the contention occurring in packet switching is resolved by a store-and-forward technique [YMD00]. Packets are stored in a queue from which they are forwarded later. In the case of optical transmission, the buffering is a complex task as there is no optical random access memory (RAM) available. The lack of optical RAM can be compensated for by using a set of fibre delay lines (FDL) to resolve contention. Other possible schemes to deal with contention include deflection routing, where otherwise lost packets are sent in some other direction in the network, from where they will be forwarded once again towards their original destination.

The header format must be chosen carefully as the capacity of optical networks is huge and the processing of the headers must be accomplished in a shorter time interval than is the case in electronic networks. It is likely that header processing must first be done electronically, which means a conversion from the optical to the electronic domain for the making of a routing decision, and then later back to the optical domain.

1.7 Optical Burst Switching

Optical burst switching (OBS) can be seen as an intermediate step from circuit-switched WDM networks towards optical packet switching. In an optical burst switched network (OBSN), the data are transferred using optical bursts consisting of several (IP) packets going to the same destination. Thus, OBSNs can be seen as a halfway technology towards all-optical packet switched networks (OPSN). Fig. 1.7 illustrates the evolution of optical networks. The OBS scheme has been proposed for metropolitan area ring networks (MAN) as well as to backbone networks (wide area networks, WAN).

Before sending the actual burst, the source node sends a control packet which will reserve the necessary resources. The reservation schemes can be divided into two classes [Dix03]. The first class consists of schemes where the reservation is made for a certain time period. Thus, the control packet includes information on both the arrival time of the burst and its length. The second class consists of schemes where the reservations are explicitly torn down by another control packet at some later point.

One popular proposal is the so-called Just-Enough-Time (JET) protocol originally proposed by Qiao and Yoo in [QY99]. In the JET protocol, the source node first sends a control packet, whose task is to reserve the necessary resources along the route for a certain time period. Then, without waiting for a positive acknowledgement, the source node sends the actual burst along the same route [QY99, DG01, Bat02].

The control packet is processed at every intermediate node to see if there are enough free resources available at the next link. If there is a free wavelength channel available the control packet is sent further. The total time it takes for the control packet to reach the destination node is simply the sum of the propagation times over the links and the processing times at the intermediate nodes.

As the actual burst is not processed at the intermediate nodes, the burst cuts through the network and and reaches the destination node faster than the control packet. Thus, the offset time between sending the control packet and the burst must be long enough to ensure that the destination node has processed the control packet before the actual burst arrives. In the JET protocol, the source node knows the route and hence the number of intermediate nodes in it. Thus, the source node can determine the minimum possible offset time before sending each burst. The OBS protocol is discussed in more detail in Chapter 5.

1.8 Contents of the Thesis

This thesis focuses on certain issues in resource allocation and the performance evaluation of backbone wavelength-routed (WR) networks and the metropolitan area optical burst switched networks (OBSN). In particular, first we consider heuristic algorithms for the static routing and wavelength assignment (RWA) and logical topology design (LTD) problems arising in the context of WR networks. Then, in the next part we apply the theory of Markov decision processes (MDP) to the dynamic RWA problem in WR

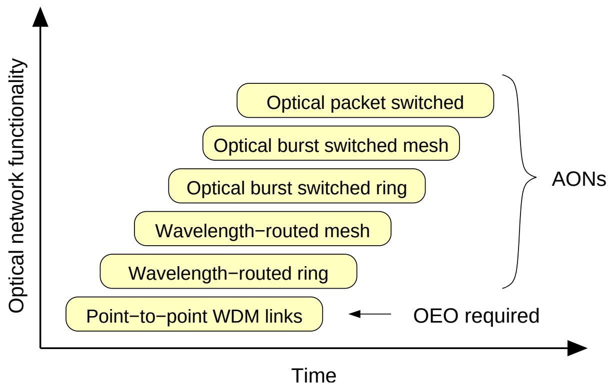

Figure 1.7: The optical network evolution (adopted from [BP03, DGSB01]).

networks including a novel decision analysis using importance sampling techniques. The topic of the final part is OBS networks where the focus is on fairness issues.

Static RWA and LTD Problems

The network topologies considered are generally arbitrary mesh networks, but regular structures, such as ring networks are also discussed. In backbone networks, both static and dynamic cases are considered. In a static RWA problem, one is asked to set up a given set of lightpaths (or light trees) in an optical WR network with given constraints, where often the objective is to minimise the number of wavelength channels required. In Publication 1 (and [HV98, HV99]), our contribution is the introduction of heuristic algorithms for static RWA problems. In particular, in [HV98, HV99] the present author has studied the applicability of several well-known combinatorial algorithms (e.g. simulated annealing and genetic algorithms) to the wavelength assignment (sub)problem in single and multifibre networks. In Publication 1, we formulate a generalisation of the standard static RWA problem (GRWA), where the request set may consist of anycast and multicast requests as well. Furthermore, several heuristic algorithms are generalised to the GRWA problem and compared by means of numerical examples.

Finally, in Publication 2 a slightly more generic problem called the logical topology design (LTD) problem is considered. In the LTD problem, one is asked to decide on the lightpaths to be established as well as the routing at the packet layer (e.g. IP). The complexity of the LTD problem is very high and in practice heuristic solutions are used. In Publication 2, two improved heuristic algorithms for LTD are proposed. The proposed changes allow us to use alternative routes, which is not a trivial task as the longer routes consume more resources and should generally be avoided. To this end we propose a heuristic order in which the different route candidates are tried in a greedy manner (first-fit).

Dynamic RWA Problem

In the dynamic control problem, connection requests, i.e. lightpath requests, arrive according to a certain traffic pattern and the task is to establish one lightpath at a time in the WR optical network so that the expected revenue rate is maximised or the expected cost rate is minimised. Typically the optimisation goal is to minimise some infinite time horizon cost function, such as the blocking probability.

In this thesis, the analysis of dynamic RWA policies is performed within the framework of Markov Decision Processes (MDP). In Publications 3 and 4, we study several heuristic algorithms and propose a very generic framework for improving any given RWA algorithm by using a so-called first policy iteration (FPI). In FPI, one has few good candidates to choose from and proceeds by evaluating the expected change in future costs resulting from each decision by short simulations. In simulations, consecutive decisions are made using a so-called standard policy.

The usability of the FPI approach is limited by the computational effort involved in the estimation of the changes in future costs. In order to alleviate this, in Publications 5 and 6 we combine the importance sampling technique (IS) with FPI and propose an adaptive algorithm which estimates the optimal IS parameters at the same time as it collects data for the decision-making analysis.

OBS Networks

The last part of the thesis considers OBS networks, which are an intermediate step towards full optical packet switching networks. In OBS networks, the data are transferred using optical bursts consisting of several (IP) packets going to the same destination. On the route of the burst temporary reservations are made only for the time during which the burst is transmitted. While several OBS protocols have been proposed in the literature, this thesis mainly deals with the Just-Enough-Time protocol (JET), in which the link reservations are made by a control packet and the actual burst is sent after a certain offset time without waiting for any acknowledgement of the reservations. Our contributions to the OBS paradigm mainly deal with fairness issues, which tend to be a weak point of OBS protocols. In Publication 7, we study how fibre delays lines (FDL) affect the JET protocol and demonstrate that they not only lower the blocking probability but also improve the fairness between long and short connections. This is a side effect of the increasing offset time as the bursts get delayed.

In Publication 8, we propose an OBS-aware routing scheme, which tries to minimise the link loads at the same time as ensuring fairness. The routing problem is formulated as a mixed integer linear programming (MILP) problem, where we explicitly deny new connections on the same route and wavelength channel after a certain number of hops. Thus, each burst competes for resources at most on the first m hops and consequently different traffic flows are treated more equally.

OBS has also been proposed for metropolitan area ring networks, which typically operate either on fixed transmission channels, or, alternatively, using fixed receivers. The edge router connected to the ring collects arriving packets and assigns them to different queues on the basis of their destination address. A scheduling algorithm then decides which queues are to be served and when. In Publication 9, we derive analytical models for the blocking probability of some MAC protocols using either random order or a round-robin order scheduling policy.

2 STATIC ROUTING AND WAVELENGTH ASSIGNMENT PROBLEM

2.1 Introduction

In this chapter, we consider the problem where a given set of lightpaths is to be established in a given physical network. This is called the static routing and wavelength assignment (RWA) problem. The RWA problem in wavelength-routed WDM networks (WRN) consists of choosing a route and a wavelength (RW-pair) for each connection so that no two connections using the same wavelength share the same fibre [BM96, Bar98]. The requirement that connections sharing the same fibre must use different wavelength channels is referred to as *distinct channel assignment* requirement, or shortly DCA (see, e.g. [SB99, RS98, Muk97]):

Definition 2.1 (Distinct Channel Assignment [DCA])
Connections sharing a common fibre must use distinct wavelengths.

A violation of the DCA constraint is often referred to as a *wavelength conflict*. Furthermore, if wavelength conversion is not possible at the network nodes, then an additional constraint, called *wavelength continuity*, must be satisfied, i.e. each connection must use the same wavelength on every link. This constraint together with DCA gives the RWA problem in all-optical network its special characteristics.

In this chapter, it is assumed that traffic is static, i.e. the problem is to establish a given static set of connections between the given nodes in the network. The network itself can be a single or multifibre network. This kind of approach is relevant in the backbone networks, where it may be reasonable to assume that the traffic is static.

2.2 Problem Formulation

The static case of the routing and wavelength assignment problem has been widely studied in the literature. For the reference see, e.g. [TP95, BB97, GSM97, Bar98].[1] Formally the static RWA problem can be stated as follows:

Problem: Static Routing and Wavelength Assignment[S-RWA]

For a given

- *physical network $\mathcal{G} = (V, E)$, where V is the set of network nodes and E the set of links, and each link $e \in E$ has a certain number of bidirectional (or unidirectional) fibres, and*
- *set of bidirectional (or unidirectional) lightpath requests $\mathcal{L}$,*

determine a feasible RWA with a minimal number of wavelength channels, $W_{\min}$.

[1] Another kind of problem formulation arises in the context of Linear Lightwave Networks (LLN) where wavelength selective routing at the nodes is replaced by Linear Divider Combiners (LDC) [BSSLB95, SB99]. This case is not considered in this thesis.

By a bidirectional lightpath request we mean a request for a lightpath in both directions along the same route. The problem can be further classified based on the type of optical crossconnects (OXC),

WSXC: the wavelength conversion is not possible, or

WIXC: the wavelength conversion is possible.

In general, the nodes may consist of a mixture of these two types as well as nodes having a partial wavelength conversion. With a wavelength conversion capable nodes the maximum number of lightpaths sharing a link (fibre) defines the number of wavelength channels required and the WA subproblem also becomes trivial.

The absence of wavelength conversion means that the so-called wavelength continuity constraint must be satisfied in addition to the DCA constraint. The used technology sets a bound to the maximum number of wavelengths available, and if the solution requires more than that it cannot be realised in practice. Hence, by solving the S-RWA problem one knows whether a given set of lightpaths can be established into the current physical network, or whether new links (or fibres) have to be added to the network.

Typically several lightpath establishments using only $W_{\min}$ wavelength channels exist. In [PKG02b] Puech et al. propose a procedure where one first determines the minimum number of required wavelength channels, $W_{\min}$, and then solves a secondary optimisation problem within the set of solutions yielding the optimal solution in the sense of wavelength channels required. In particular, the authors propose minimising the mean number of physical links a lightpath traverses. This can be motivated by the fact that such a solution leaves more resources available for the future lightpath demands and thus possibly allows one to establish more lightpaths before a reconfiguration of the whole network required.

Furthermore, the problem definition can be extended to include installation of new fibres with a certain cost and then the aim of the optimisation becomes the minimisation of the costs. This problem is often called as a physical topology design problem and is out of the scope of this thesis.

2.3 Integer Linear Programming Formulation

The static RWA problem can be formulated as an integer linear programming problem (ILP). For simplicity let us consider the case where the lightpath requests are unidirectional and the OXC's are not capable of performing wavelength conversions. Let b_{ij} denote the number of lightpaths to be established from node i to node j. Furthermore, in order to prune the search space we introduce an additional constraint, viz. we require that the paths used to establish a lightpath from i to j may consist at most h_{ij} optical hops, where the h_{ij} are some appropriately chosen constants. The rest of the symbols are described briefly in Table 2.1. With these notations the ILP formulation is straightforward:

Objective: minimise the number of used wavelength channels W, i.e.,

$$\min\ W, \tag{2.1}$$

constant	explanation
b_{ij}	**number of lightpaths** $i \to j$.
p_{ij}	**number of physical fibres** $i \to j$, 0 if none.
h_{ij}	**physical hops constraint**, the maximum number of links a lightpath $i \to j$ can use.
variable	**explanation**
$c_{ij}^{(k)}$	**number of lightpaths** $i \to j$ **using the wavelength channel** k.
$c_{ij}^{(k)}(l,m)$	**number of lightpaths** $i \to j$ **using the wavelength channel** k **on link** $l \to m$.

Table 2.1: Notation.

Subject to:

$$\text{(channel assignment)} \qquad \sum_{k=1}^{W} c_{ij}^{(k)} = b_{ij}, \ \forall\,(i,j) \tag{2.2a}$$

$$\text{(consistency)} \qquad c_{ij}^{(k)}(l,m) \le c_{ij}^{(k)}, \ \forall\,(i,j),\ (l,m),\ k \tag{2.2b}$$

$$\text{(distinct channel)} \qquad \sum_{(i,j)} c_{ij}^{(k)}(l,m) \le p_{lm}, \ \forall\,(l,m),\ k \tag{2.2c}$$

(lightpath continuity)

$$\sum_{k=1}^{W}\sum_{m}\left(c_{ij}^{(k)}(l,m) - c_{ij}^{(k)}(m,l)\right) = \begin{cases} b_{ij}, \text{if } l = i, \\ -b_{ij}, \text{if } l = j, \\ 0, \text{otherwise.} \end{cases} \quad \forall\,(i,j)\,l \tag{2.2d}$$

$$\text{(hop constraint)} \qquad \sum_{(l,m)} c_{ij}^{(k)}(l,m) \le h_{ij}, \ \forall\,(i,j),\ k \tag{2.2e}$$

$$\text{(value range)} \qquad c_{ij}^{(k)} \in \{0,1\}, \ \forall\,(i,j),\ k \tag{2.2f}$$

$$c_{ij}^{(k)}(l,m) \in \{0,1\}, \ \forall\,(i,j),(l,m),k \tag{2.2g}$$

Example
In Fig. 2.1, an example of a wavelength-routed network is depicted. The network consists of $N = 9$ nodes and $L = 13$ links, where each link is assumed to consist of a single bidirectional fibre, $F = 1$.

In the example network, a lightpath is established between each node pair, i.e. the resulting logical topology is a fully connected mesh. It is assumed that the OXCs are not capable of performing wavelength translation, i.e. they are of the type WSXC. The illustrated lightpath establishment uses 5 wavelengths, which also turns out to be the optimal solution in this case [Hyy01].

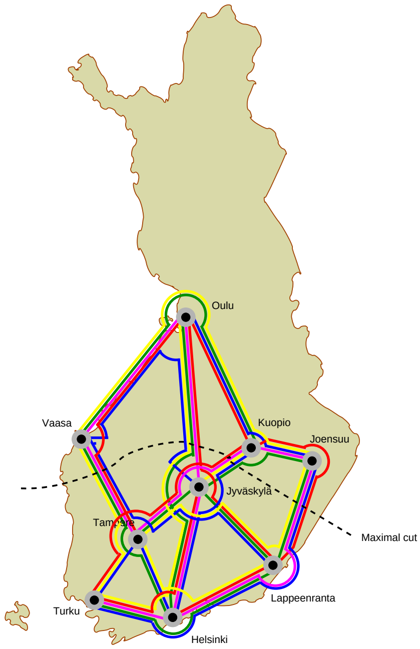

Figure 2.1: A hypothetical WDM network in Finland with a connection between every node pair.

2.4 Bounds for the Number of Wavelengths Required

Partition Bound

A lower bound for the number of wavelength channels required can be found by cutting the network into two parts (see e.g. [Bar98, SB99] and [HV98]). Let c denote an arbitrary cut of the physical topology into two sets. For any cut c a certain number of connections, denoted by Z_c, cross it. By dividing the number of connections crossing the cut by the number of physical fibres going through the cut we get the average number of connections per fibre, which is clearly a lower bound for the number of wavelength channels required in order to avoid wavelength conflicts in those fibres, i.e. we get the so-called partition (or maximal cut) bound,

$$W_{\min} \geq \max_c \left\lceil \frac{Z_c}{F_c} \right\rceil, \tag{2.3}$$

where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x, Z_c the number of connections through cut c and F_c the number of fibres crossing the cut.

In the example network depicted in Fig. 2.1, the optimal cut divides the network horizontally just above the city of Jyväskylä. There are 4 links crossing the cut and $4 \cdot 5$ connections using those links, i.e. at least 5 wavelength channels are required to establish a fully connected logical topology.

In the case of a fully connected network, the number of connections crossing the cut is generally $N_A \cdot (N - N_A)$ where N is the total number of nodes and N_A is the number of nodes in part A. Hence, assuming N is even, the maximum number of connections crossing a cut is obtained by dividing the nodes into two equally large groups, giving a total number of $N^2/4$ connections. However, the number of fibres crossing such a cut can be great and generally the tightest lower bound is found by exhaustively considering all possible cuts. This is clearly infeasible in practice as the number of possible cuts can be enormously large. However, a heuristic algorithm can be used to find a reasonably good cut within a practical amount of time [SB99].

Distance Bound

Another lower bound for $W_{\min}$ can be obtained by considering the minimum number of link channels needed for establishing the set of lightpath requests. Assuming unidirectional lightpath requests $\{s, d\}$ let d_{sd} denote the length of the shortest path, in number of optical hops, from s to d. Then, the total number of link channels needed to support the given set of lightpaths must be at least $\sum_{(s,d)} d_{sd}$, which yields the so-called distance bound, [Bar98, SB99]

$$W_{\min} \geq \left\lceil \frac{\sum_{(s,d)} d_{sd}}{2 \cdot LF} \right\rceil, \tag{2.4}$$

where L denotes the total number of links and F the number of (bidirectional) fibres on each link.

It is easy to see that in the example network of Fig. 2.1 at least 128 link channels are required to establish full logical connectivity. The network has $L = 13$ links and thus the distance bound is 5, i.e. $W_{\min} \geq 5$.

Bounds for Regular Ring Topologies

In addition to general mesh topologies, also regular topologies such as rings have been widely studied. A ring network is particularly appealing because it can provide full protection against any cable cut and at the same time is easy to configure. For these reasons it has been used in, e.g., self-healing SDH/SONET rings (SHR) [SB99].

For bidirectional WDM rings with arbitrary number of nodes the optimal lightpath establishment providing a full connectivity by a single lightpath between each node pair is known exactly [LS00b, LH03]. Namely, the minimum number of wavelength channels required to establish a full connectivity in a single fibre optical ring network with N nodes is

$$W_{\min} = \begin{cases} \lfloor \frac{N+2}{4} \rfloor + \lfloor \frac{N(N-2)}{8} \rfloor & \text{for even number of nodes } N, \\ \frac{N(N-1)}{2} & \text{for odd number of nodes } N, \end{cases} \tag{2.5}$$

where $\lfloor x \rfloor$ denotes the highest integer lower than or equal to x. In [LH03] the authors also propose an elegant algorithm for determining the optimal lightpath establishment without using any complex calculations or data structures.

indexRWA!decomposition

The static RWA problem can be solved either in one phase, where both a route and a wavelength are determined at the same time, or alternatively in two phases, where first the routes are fixed and then a feasible wavelength assignment (WA) is determined for the given routing. One can find algorithms based on both approaches in the literature. In this section we consider the two phase approach, i.e. decompose the static RWA into two subproblems. Later in Section 2.7 we present a so-called layered algorithms which represents the other approach, i.e. where both the route and wavelength channel are determined at the same time.

Routing

The traditional strategy to solve the static RWA problem is to first determine a route for each lightpath and then assign feasible wavelengths to them. Even though the problems are not independent, this is likely to yield moderately good solution to static RWA problem. The usual way to decide on the routes is to choose (one of) the shortest path(s) for each connection. This is justified by the fact that the longer routes use more network resources and are thus more likely to lead to a less efficient overall solution (c.f. the distance bound in Section 2.4). If there are several equally long shortest paths, then typically one of them is chosen randomly [HV98]. The optimal solution is often obtained by using short routes, but not necessary by the shortest path for every lightpath request (in order to avoid unnecessary congestion on some links). As the number of lightpaths sharing an optical link sets an lower bound for the number of wavelength channels needed, it can be used as a temporary objective for routing step (cf. MNH algorithm [BB97, Bar98]). However, the final quality of the chosen routing is only known after the wavelength channel assignment step has been accomplished.

Wavelength Assignment

The unique feature of WDM networks is that wavelength conflicts are not allowed (DCA constraint, Def. 2.1), i.e. no two connections using the same wavelength may share a common link (or fibre to be exact). Once routing is fixed the problem is to assign a feasible wavelength for each route using a minimum number of wavelength channels. This is called the wavelength assignment (WA) problem. In the general case, there are several fibres between some of the links. A straightforward approach is to assign the lowest possible[2] wavelength to one connection at a time in some order (first-fit algorithm). The order in which the wavelengths are assigned can be a critical factor for greedy algorithms. A good rule of thumb is to assign a wavelength first to those connections which have the most dependencies, i.e. share most links with other connections. This is presented formally in Algorithm 1.

[2] a wavelength that does not cause a wavelength conflict

Algorithm 1 Static wavelength assignment (WA) algorithm

```
let $\mathcal{X}$ be a predetermined set of routes for lightpath requests
let $d_j = \sum_{i \in \mathcal{X}} I(i \text{ and } j \text{ share a common link}) \; \forall j \in \mathcal{X}$
sort routes in the decreasing order of $d_j$
start from an empty network
for each $j \in \mathcal{X}$ in sorted order do
    set up a lightpath using route $j$ on the lowest feasible wavelength(s)
end for
```

Iterative Improvements

A straightforward way to improve the current solution is to change the set of routes a little and then assign the wavelengths again[3]. If the wavelength assignment is not computationally a too expensive operation, then well-known local search techniques such as simulated annealing, genetic algorithms, or tabu search can be applied to obtain the optimal set of routes (for a given wavelength assignment policy). Thus, the result from wavelength assignment is fed back to the upper level algorithm as the value of the cost function in the current point (=routing). However, as stated before, this requires a moderately fast wavelength assignment algorithm.

2.6 Single Fibre Networks

Graph Node Colouring Problem

The wavelength assignment in a single fibre network[4] is equivalent to the node colouring problem, which is a well-known graph theoretic NP-hard problem (see e.g. [SK77, AH77, BM76]). In the node colouring problem, the task is to assign a colour to each node of the given graph with minimal number of colours so that no neighbour nodes have the same colour. Graph node colouring problems arise in many different context and in practice one often must rely on heuristic algorithms. Greedy node colouring algorithms are typical lightweight approaches where the algorithms assigns the lowest possible colour to one node at a time at a certain order. For details see, e.g. [Mit76, Bré79]. Also several novel heuristic algorithms have been applied to solve the node colouring problem including simulated annealing, tabu search and genetic algorithms [HdW87, Ree95, RSORS96].

The relation between the wavelength assignment and graph node colouring problem is the following. When the set of routes is fixed the task is to assign a feasible wavelength to each connection, i.e. no two connection using the same link may use the same wavelength. Let each connection represent a node in a so-called (WA) conflict graph $\mathcal{G}$ and set such connections that share at least one link (fibre) as neighbours in the graph. By finding the optimal colouring for this graph, we have also found the optimal wavelength assignment for the given routing.

In [HV98], the present author has studied the applicability of the pro-

[3] The opposite, changing wavelengths little and then finding a feasible routing, is usually harder to solve.

[4] Or in multifibre network if the routing step has fixed also the used fibre on every link.

posed heuristic graph node colouring algorithms for the wavelength assignment problem. In particular, we have considered simulated annealing (SA), genetic algorithms (GA) and tabu search (TS) heuristic algorithm and compared their performance to several well-known greedy algorithms. With random network topologies the Tabu search version turned out to give the best results in terms of the number of wavelength channels required.

Graph theory also provides some bounds for the number of wavelengths required. For example, an upper bound for the number of wavelengths required is obtained from

$$W_{\min} \leq \Delta + 1,$$

where Δ is the maximum degree (the number of neighbours a connection has) of the conflict graph $\mathcal{G}$. A more strict lower bounds can be obtained by considering cliques in the conflict graph $\mathcal{G}$. Graph theoretic bounds, however, are usually not very strict and thus not very useful in this context.

Greedy RWA Algorithm

Next we briefly present a simple greedy algorithm, Algorithm 2, which can be used to obtain a reasonable good solution to the static RWA problem. The algorithm solves the static RWA problem in two parts. First it uses a shortest path algorithm to obtain a single path for each lightpath request, and then assigns the wavelength channels using an arbitrary node colouring algorithm. A reasonably good greedy node colouring algorithm can be found, e.g. from [Bré79]. If the problem size permits it may be possible to use more sophisticated node colouring algorithms, such as tabu search or simulated annealing, or even to do an exhaustive search [HV98]. Note that Algorithm 2 uses only (one of) the shortest path(s) for each lightpath.

Algorithm 2 Greedy RWA Algorithm for Single Fibre Networks

1: find (one) shortest path for each lightpath request $(s_i, d_i) \in \mathcal{L}$
2: form WA graph $\mathcal{G}$ where each node represents one lightpath, and lightpaths
3: form conflict graph $\mathcal{G}$ for WA where each node represents one lightpath, and lightpaths sharing a link are set as neighbours
4: use greedy node colouring algorithm (e.g. [Bré79]) to colour the nodes of $\mathcal{G}$
5: return paths and chosen wavelengths

2.7 Layered Approach for Static RWA

For the rest of this chapter we do not limit ourselves to the single fibre case. The algorithm described next tries to solve the static RWA problem so that the chosen routing takes into account the restrictions from the wavelength assignment, while still being very fast. The layered RWA algorithm proposed by the present author in Publications 1 and 2 resembles closely the CP1 algorithm (Algorithm 5) originally proposed in [ZA95]. The CP1 algorithm is a logical topology design algorithm which has to decide on the logical topology as well. To this end, CP1 algorithm first fixes one route for each node pair. Then each node pair (s, d) is given a rank based on the

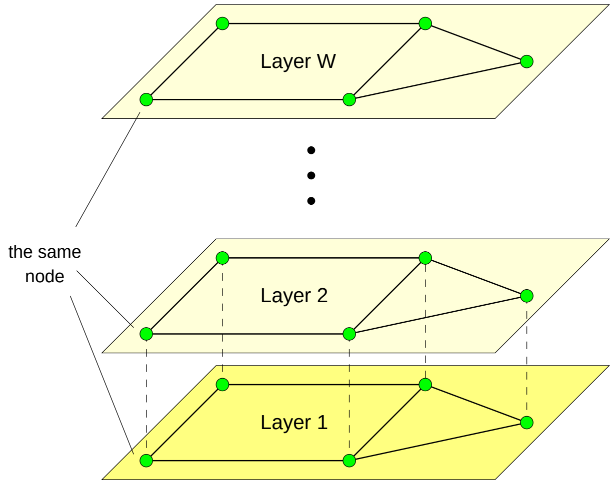

Figure 2.2: Layered view: WRN consists of W identical layers each having the same set of links.

amount of single hop traffic, i.e. the volume of traffic from s to d. At each step, CP1 sets up a lightpath for a node pair having the highest rank with a free lightpath using the previously determined route.

Thus, CP1 algorithm tries to set up as many lightpaths as possible in the order defined by how much single hop traffic a new lightpath between a given node pair would carry. The layered RWA algorithm, on the other hand, tries to establish a certain set of lightpaths using any feasible routes.

Wavelength Layers

In the absence of the wavelength conversion, a WRN with W wavelength channels consists of W identical layers as illustrated in Fig. 2.2. In other words, each layer represents available links at the respective wavelength and, without wavelength conversion, a lightpath must remain at the same wavelength layer throughout the path from the source to the destination.

Lexicographic Order for Paths

First-fit algorithms with alternate routes require a well-defined order between the alternative paths. A convenient way to express this is to define a unique number sequence for each path and then use the so-called lexicographic order:

Definition 2.2 *Let A and B be arbitrary sequences of real numbers, $a = (a_1, \ a_2, \ \ldots)$ and $b = (b_1, \ b_2, \ \ldots)$. Then, $a < b$ in lexicographic order iff*

$$\exists i \ : \ a_i < b_i \text{ and } a_j = b_j \ \forall j < i.$$

In other words, the order of two sequences is defined by the first different number in the sequence (smaller first).

Path Candidates

The RWA algorithm presented in Publication 2 utilises a so-called alternate routing strategy. That is, for each lightpath request we determine one or more possible path candidates in advance. These can be obtained by using a k-shortest path algorithm first to obtain a path candidate set. The set is then ordered using a heuristic rule, which tends to lead close to optimal solutions. In particular, we define a unique number sequence $S(p)$ for each path $p = (a_0, a_1, \ldots, a_{\ell(p)})$,

$$S(p) = (\Delta(p), -\ell(p), -b(p), a_0, a_1, \ldots, a_{\ell(p)}, 0, \ldots) \qquad (2.6)$$

where $b(p)$ is 2 if path p is to be reserved in both directions (i.e. a bidirectional request) and 1 otherwise, $\ell(p)$ is the length of path p (in hops), and $\Delta(p)$ is the number of additional hops the path p uses when compared to the shortest path in the sense of number of hops,

$$\Delta(p) = \ell(p) - \ell_{\min}(p),$$

where $\ell_{\min}(p)$ is the length of the shortest path between the same node pairs.

The path candidates are then sorted using $S(p)$ and the lexicographic order given by Def. 2.2. Note that the number sequence in (2.6) sorts the paths in a certain, usually favourable, order. The primary key is the number of additional hops, i.e. the paths using additional hops are given a lower priority (c.f. the distance bound in Section 2.4). The second criterion is the length of the path, i.e. longer paths come before shorter ones. This is motivated by the fact that it tends to be harder to find a free wavelength for a longer connection than for a shorter connection (c.f. greedy node colouring algorithm in [Bré79]). If these two criteria are still equal, then possible bidirectional requests are given a higher priority over unidirectional requests. If all three criteria are equal, then the order is defined by the first different node along the paths; smaller node number comes first.[5] The last criterion is only needed to ensure an unambiguous ordering, which makes it possible to reproduce the simulation results.

Hence, the first set consists of the shortest paths, the second set of paths with one additional hop and so on. Within each set the order is such that the longer paths come first. The order of equally long paths is defined by the nodes' numbers.

Determination of the RW-pairs

Similarly as in [ZA95], after the route candidates have been determined the problem is split into several subproblems in which the objective is to use the resources of one wavelength layer maximally. First we take the wavelength layer 1 and establish as many lightpaths there as possible. After that layer 1 is considered "frozen" and we are left with a new smaller problem where the established lightpaths are removed from the demand list. By doing this we hope to find a good lightpath establishment for one wavelength layer at a time leading to a satisfactory overall solution.

[5] When searching the k shortest paths we use the same logic in case of equally long paths.

Algorithm 3 Layered RWA, L-RWA

let $\mathcal{L} = \{(s_i, d_i)\}$ be a set of node pairs corresponding to the lightpath requests
find the k shortest paths for each $(s_i, d_i) \in \mathcal{L}$, and store the triples (s_i, d_i, p_i), where p_i is the corresponding path, to a list $\mathcal{X}$
sort the list $\mathcal{X} = \{s_i, d_i, p_i\}$ in lexicographic order defined by $S(p_i)$
set $W \leftarrow 0$
while $\mathcal{X} \neq \emptyset$ **do**
    set $W \leftarrow W + 1$
    for each $(s, d, p) \in \mathcal{X}$ **do**
        if p fits in layer W **then**
            assign path p at layer W for lightpath (s, d)
            remove all paths $s \rightarrow d$ from $\mathcal{X}$
        end if
    end for
end while
return W

The layered RWA algorithm proposed in Publication 2 is described formally in Algorithm 3. The algorithm also closely resembles the greedy node colouring algorithm described earlier. The difference is that here we fix both the route (node) and the colour, and only one node corresponding to each $s - d$ pair is given a colour [LS00a].

The algorithm can easily be extended to the case where there is more than one lightpath request for some $s - d$ pairs. Namely, instead of immediately removing all the paths $s \rightarrow d$ after assigning one path, one decreases a "multiplicity" counter for $s \rightarrow d$. Then the paths $s \rightarrow d$ are only removed from the working set $\mathcal{X}$ when the corresponding counter reaches zero.

The performance of the presented algorithm is limited by the set of paths defined by the parameter k and, especially, by their ordering. Indeed, there always exists a constant $k < N$ and some order for the candidates routes, which leads to the global minimum. Thus, we have the following proposition.

Proposition 1
An optimal solution in the sense of the required number of channels can always be reached by the layered RWA algorithm for some order of paths.

Proof:
Let Z_i be the set of paths at layer i in the optimal solution. Clearly any ordered set beginning with $\{Z_0 \; Z_1 \; \ldots \; Z_W\}$ leads to the optimal solution.

2.8 Results with RWA Heuristics

In order to validate the performance of the presented heuristic RWA algorithms, we use the network illustrated in Fig. 2.3. The logical topology to be established consists of a lightpath between every node pair. Table 2.2 contains the numerical results obtained with different heuristic algorithms. The bidirectional case corresponds to the situation where each lightpath is used in both directions. Similarly, unidirectional case corresponds to the situation where the lightpath $a \rightarrow b$ can traverse a different route than the

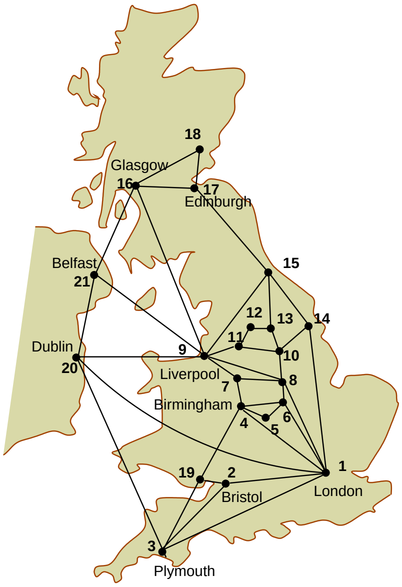

Figure 2.3: UKNet, a telephone network located in UK consisting of 21 nodes and 39 links.

lightpath $b \to a$. Clearly the bidirectionality is an additional constraint and the optimal solution to unidirectional problem will use at most the same number of wavelength channels as any feasible solution to the bidirectional version of the same problem.

It can be noted from the results that the simple greedy heuristics with the shortest path routing does not perform very well. On the other hand, the layered RWA algorithm gives reasonably good results. The results of [Bar98] presented here for comparison were obtained with a moderately complex heuristic algorithm (MNH, see [BB97, Bar98] or Publication 1 for details). As a conclusion we can expect that the layered RWA algorithm, while being very fast, also provides near optimal solutions.

Algorithm	unidirectional	bidirectional
Greedy	32	31
Layered	22	23
Baroni [Bar98]	-	20

Table 2.2: Results of fully-connected UKNet (Publication 2).

2.9 Generalised Routing and Wavelength Assignment Problem

In Publication 1, the present author generalises the static RWA problem to include both multicast and anycast requests in addition to standard unicast requests (lightpaths between given node pair). The resulting problem is called the generalised routing and wavelength assignment (GRWA) problem.

$$
\begin{aligned}
t(a) &= \text{type of request (unicast, anycast or multicast)},\\
s(a) &= \text{source node of request } a,\\
D(a) &= \text{set of destination nodes of request } a,\\
m(a) &= \text{multiplicity, number of wavelength channels},\\
b(a) &= \text{bidirectionality: 2 (bidirectional) or 1 (unidirectional)}.
\end{aligned}
$$

Table 2.3: The notation in GRWA problem for a request a.

Note that an optical multicast request corresponds to a light-tree from a root node to a given set of destination nodes. Such a tree can be used to deliver the same data to several destinations simultaneously. For example, one could consider using an optical multicast tree to broadcast multiple video streams to several locations at the same time, or to distribute huge amounts of scientific or backup data to several locations. Alternatively a light-tree can be used as a shared medium with an appropriate MAC protocol (cf. passive optical networks).

Similarly, an anycast request corresponds to a lightpath request from a given source node to any destination node. Anycast request can be used, e.g. as an optimisation tool, in order to allocate remaining resources in a reasonable way. On the other hand, an operator-to-operator interface may consist of several alternative edge nodes, which clearly corresponds to an anycast request. Other possible applications could be distributed data storage and backup services.

In summary, for a given optical network $\mathcal{G}$ one is suppose to establish a given set of connections consisting of three different types:

1. **unicast** lightpath requests (point-to-point),
2. **anycast** requests, i.e. the destination node can be any of the given set (one-to-any),
3. **multicast** requests, i.e. the optical signal is routed to several destinations (one-to-many).

In each case the request can be for one or more channels, where each channel can be routed independently of others using a different route and wavelength. Furthermore, requests may be bidirectional, where the same route and wavelength is used in both directions, or unidirectional. Denote the set of connection requests with $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$. The multiplicity of request a is denoted with $m(a)$ and it defines the number of requested wavelength channels. For example in case of normal bidirectional lightpath request the multiplicity defines the number of requested lightpaths between the node pair. The notation is presented in Table 2.3.

Formally the generalised (static) RWA problem can be stated as follows:

Problem: Generalised Routing and Wavelength Assignment[**GRWA**]

For a given

- *physical network $\mathcal{G} = (V, E)$, where V is the set of network nodes and E the set of links, and each link $e \in E$ has a certain* *number of bidirectional (or unidirectional) fibres, and*

– *set of optical connection requests $\mathcal{A}$ consisting of both unidirectional and bidirectional unicast, anycast, and multicast requests*

determine a feasible RWA with a minimal number of wavelength channels, $W_{\min}$.

2.10 Heuristic Algorithms for GRWA Problem

In Publication 1 the present author also proposes two efficient heuristic layered algorithms (L-GRWA and DL-GRWA) for the GRWA problem. Both algorithms use the same layered approach as described earlier, i.e. they establish one connection (unicast, anycast or multicast) at a time in the current wavelength layer until the layer is fully utilised and then move to the next layer. In addition to layered algorithms several other heuristic RWA algorithms proposed in the literature are extended to GRWA problem (e.g. a so-called minimum hop heuristic (MNH) by Baroni et al. in [BB97] and anycast algorithms by Tang et al. in [TJWW03]).

Layered GRWA Algorithm
The layered GRWA algorithm (L-GRWA) is a straightforward extension of L-RWA Algorithm 3. In particular, each request can fulfilled by allocating a set of links and at the start the L-GRWA algorithm determines a set of possible routes or trees for each request $a \in \mathcal{A}$. Then these route/tree candidates are ordered similarly as in L-GRWA based on the number of additional links reserved, the number of links reserved, etc. After that the algorithm proceeds similarly as the L-RWA algorithm. For details see Publication 1.

Dynamic Layered GRWA Algorithm
For simplicity let us first assume that the request set only consists of normal lightpath requests. The dynamic layered GRWA (DL-GRWA) algorithm proceeds as follows. At each step DL-GRWA first picks one connection request and establishes a lightpath for it using the shortest free path at the current wavelength layer. Once no new connection can be established at the current layer the algorithm moves to the next layer and repeats the same procedure. This is repeated until all the requests have been set up. In order to reach the optimal configuration the algorithm should always pick "the right connection". In Publication 1 the present author proposes a heuristic order similar to the one used with L-GRWA, i.e. at each step the longest connection using the least amount of "extra links" is set up. This can be achieved as follows. Let $\ell(s,d)$ denote the length of the shortest path (in hops) from s to d in an empty network and, similarly, let $\ell'(s,d)$ denote the respective length at the current state. At each step the request $a = (s,d)$ to be established next is the one which minimises the quantity,

$$\begin{aligned} c(s,d) &= \left(\tfrac{N-1}{N}\right) \cdot \ell'(s,d) - \ell(s,d) \\ &= (\ell'(s,d) - \ell(s,d)) - \tfrac{1}{N} \cdot \ell'(s,d). \end{aligned} \tag{2.7}$$

Clearly the first term, $\ell'(s,d) - \ell(s,d)$, corresponds to the number of extra hops and the second term, $\frac{1}{N} \cdot \ell'(s,d)$, is a normalised path length less than

1 (note that for all reachable (s,d)-pairs $\ell'(s,d) < N$). Thus, the number of extra hops, $\ell'(s,d) - \ell(s,d)$, serves as the primary key (less extra hops first) and the ties are broken by the length of the path (longer paths first).

The DL-GRWA algorithm is formally described in Algorithm 4 for a general case where $\mathcal{A}$ consists of both unidirectional and bidirectional requests. If there are both unidirectional and bidirectional requests one needs to determine the shortest paths (and the respective criteria) for both cases separately. To this end, let $H_1(\mathbf{P}) = \mathbf{H}_1$ denote the unidirectional "hop matrix", and $H_2(\mathbf{P}) = \mathbf{H}_2$, respectively, the bidirectional "hop matrix",

$$h_1(i,j) = \begin{cases} 1, & \text{if } p_{ij} > 0 \\ \infty, & \text{otherwise.} \end{cases}$$

$$h_2(i,j) = \begin{cases} 1, & \text{if } p_{ij} > 0 \text{ and } p_{ji} > 0 \\ \infty, & \text{otherwise.} \end{cases}$$

By using an all-pairs shortest path algorithm for $\mathbf{H}_1$ and $\mathbf{H}_2$ one obtains the lengths of the shortest unidirectional and bidirectional lightpaths, i.e. let matrix $\mathbf{D}_1$ contain the number of optical hops along the shortest unidirectional lightpath from i to j, and matrix $\mathbf{D}_2$ contain the shortest distances for the bidirectional lightpaths, respectively. At each round the algorithm determines a path selection criterion $c(a)$ for each request $a \in \mathcal{A}$ based on the appropriate hop metric ($\mathbf{D}_1$ or $\mathbf{D}_2$) in an empty network and at the current state.

Furthermore, an optional constraint on the maximum number of extra hops allowed, Δ_ℓ, is introduced. Note that setting $\Delta_\ell = \infty$ means that all feasible paths are accepted. However, setting a finite limit on the number of extra hops might turn out to be useful when more connections are expected later.

Note that there are no additional parameters, like for pruning the routes, or successive iterations involved, which makes this approach particularly attractive.

It is straightforward to extend the DL-GRWA algorithm to handle anycast requests. For anycast request $a = \{s, D\}$ one must evaluate each possible destination and the path selection criterion must be adjusted slightly to take into account the alternative destinations. In particular, for anycast request a and $(s,d) \in a$ we suggest using the shortest path in an empty network to the nearest node from D as the reference length,

$$c(a,s,d) = \left(\frac{N-1}{N}\right) \cdot \ell'(s,d) - \min_{i \in D(a)} \ell(s,i). \tag{2.8}$$

Multicast connections could be handled by the dynamic routing algorithm as well, but that would increase the complexity considerably. Thus, in Publication 1 we propose using a fixed set of routes for multicast connections along the lines of L-GRWA algorithm and set up them first before continuing with DL-GRWA. It is assumed that this approach does not hinder the final solution much as long as the proportion of multicast requests is reasonably small.

Complexity of DL-GRWA Algorithm

For now assume that there are no multicast requests. At each step of Algo-

Algorithm 4 Dynamic Layered GRWA, DL-GRWA.

find all-pairs shortest paths for $H_1(\mathbf{P})$ and $H_2(\mathbf{P}) \Rightarrow$ distances $\{\mathbf{D}_1, \mathbf{D}_2\}$
$W \leftarrow 1$
$\mathbf{P}' \leftarrow \mathbf{P}$
while $\mathcal{A} \neq \emptyset$ **do**
  find all-pairs shortest paths for $H_1(\mathbf{P}')$ and $H_2(\mathbf{P}') \Rightarrow$ distances $\{\mathbf{D}_1', \mathbf{D}_2'\}$
  $\mathbf{C}_i \leftarrow \left(\frac{N-1}{N}\right) \cdot \mathbf{D}_i' - \mathbf{D}_i,\ i = 1, 2$
  set $c(a) = \left(\mathbf{C}_{b(a)}\right)_{s(a),d(a)} \quad \forall a \in \mathcal{A}$ {path selection criteria}
  $\mathcal{A}' \leftarrow \{a \in \mathcal{A} : c(a) < \Delta_\ell - 1\}$
  if $\mathcal{A}' = \emptyset$ **then**
    $W \leftarrow W + 1$
    $\mathbf{P}' \leftarrow \mathbf{P}$
  else
    $a \leftarrow \underset{a \in \mathcal{A}'}{\arg\min}\ c(a)$
    set up request a using the respective shortest path p to layer W
    reduce the number of free fibres in $\mathbf{P}'$ along path p
    decrement multiplicity $m(a)$ by one
    if $m(a) = 0$ **then**
      remove request a from $\mathcal{A}$
    end if
  end if
end while

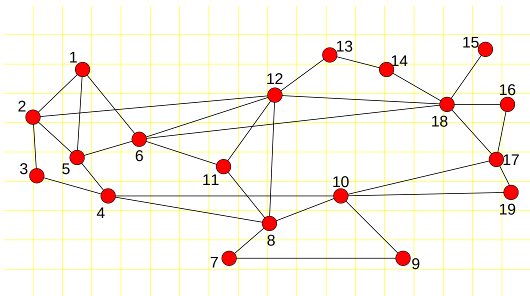

Figure 2.4: Example network: MCI backbone network in US [TJWW03].

rithm 4 the all-pairs shortest path algorithm is executed (twice in the case of mixed directions) having a complexity of the order of $|V|^3$. Assuming that the final configuration requires W wavelengths the main loop is repeated $|\mathcal{A}| + W - 1$ times. However, when the algorithm moves to the next wavelength layer the shortest paths are already known. Thus, the complexity of the algorithm is $|V|^3 \cdot |\mathcal{A}|$.

2.11 Results with GRWA Heuristics

The numerical results presented next are obtained using the MCI backbone network depicted in Fig. 2.4 consisting of 19 nodes and 32 links. All links are assumed to have a single fibre in both directions and no wavelength conversion is available.

The traffic scenario is generated as follows. At start the network is empty

request set	no. of requests	SP	BWC	MNH	MNH+	L-GRWA	DL-GRWA
	20	8.64	8.16	8.10	7.82	7.94	7.55
uni/any/	40	16.19	15.43	15.16	15.03	15.06	14.55
multi-	60	23.71	22.56	22.19	22.19	22.13	21.55
cast	80	31.24	29.63	29.22	29.35	29.19	28.57
	100	38.75	36.67	36.24	36.49	36.24	35.60

Table 2.4: Numerical results with the MCI backbone network.

and a set of n requests is generated. Then the chosen algorithm sets up the requests and the number of wavelength channels used is recorded. The average number of used wavelength channels is estimated by running 10000 independent request realisations.

In this scenario nodes 5, 7, 12, 18 and 19 serve as a "destination set" $D \subset V$ and the set of requests consists of unicast, anycast and multicast requests. For each request first the source node $s \in V$ is picked randomly. If $s \in D$ then the request is interpreted as a bidirectional multicast request from s to $V \setminus D$. Otherwise we pick a random destination node $d \in V \setminus \{s\}$ and if $d \in D$ then the request is interpreted as a bidirectional anycast request from s to D, and otherwise as a normal lightpath request from node s to node d.

From the numerical results, presented in Table 2.4, it can be seen that the layered algorithms L-GRWA and DL-GRWA perform well. See Publication 1 for details on other algorithms and for numerical results with other traffic scenarios.

2.12 Summaries

Summary of Publication 1

All the previous work considers either normal lightpath connections between given node pairs, or anycast connections from certain nodes to any destination node, or multicast connections from a given node to several destination nodes. In Publication 1, the present author formulates a generalised RWA problem (GRWA), where the connection requests are a mixture of normal unicast requests, anycast requests and multicast requests. In addition, we extend several previously proposed algorithms to the GRWA problem and compare their performance by means of numerical simulations. From the tested algorithms the so-called layered GRWA algorithm seems to yield the best results in terms of the number of wavelength channels used.

In contrast to the other Publications in this thesis, Publication 1 has not been subject to a peer review.

Author's Contribution to Publication 1

Publication 1 is the sole work of the present author.

3 LOGICAL TOPOLOGY DESIGN

3.1 Introduction

In wavelength-routed optical networks (WRN), each lightpath constitutes a so-called logical link to the logical topology (LT). In other words, the optical layer provides a logical topology for a higher layer protocol, e.g. ATM or IP. The topic of this chapter is logical topology design (LTD), where the aim is to determine the optimal set of lightpaths. In general, the LTD problem in optical networks is a multilayer network optimisation problem, where one has to take into account the constraints set by the packet, optical and physical layers.

Logical Topology Design Problem

Generally, in the logical topology design (LTD) problem one must decide on logical links, i.e. the virtual links visible to a higher level protocol like IP. The set of logical links constitute the logical topology (LT) and fixing the LT is referred to as the topology definition (TD).

Typically one optimises some quantity such as congestion in the network, the average packet delay or the total number of electronical interfaces. Usually the problem is formulated as a mixed integer linear programming (MILP) problem. The proposed formulations tend to lead to intractable problems. Hence, one often decomposes the problem into several subproblems, each of which may be expressed as an MILP problem. Then each subproblem is solved either exactly or approximately, e.g. by using some heuristic algorithms.

Shared-Channel Networks

Initially the LTD problem was studied in context of optical network without wavelength routing [BFG90, LA91]. In a given physical topology, e.g. a tree or a bus, the optical signals are guided to every node regardless of destination of individual wavelengths and thus there is no spatial wavelength re-use. The problem itself consists of choosing a wavelength channel to each transmitter and receiver which constitutes the logical topology, and then determining the optimal routing at the packet level. Note that the number of transmitters and receivers to be used in every node is assumed to be given and, assuming each wavelength is assigned to exactly one transmitter and one receiver, the number of wavelength channels thus gets defined.

Wavelength-Routed Networks

A WRN is an attempt to use the best of both the optical and electronical world. In WRN, we have a wavelength re-use, i.e. several connections may use the same wavelength as long as they do not use a common link (or fibre to be exact). Thus, in WRN the LTD problem involves also the determination of routes and wavelength channels for lightpaths and the overall problem becomes even more challenging.

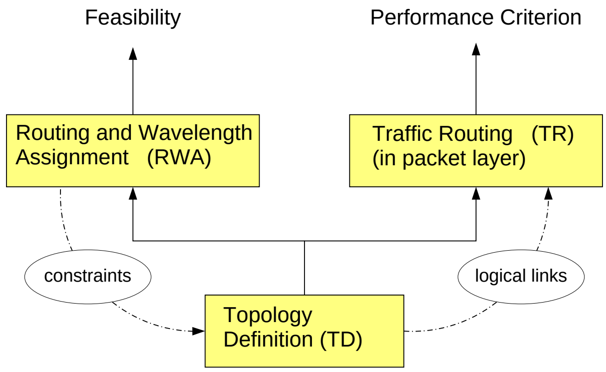

Figure 3.1: One view on relationships between TD, RWA and TR in the joint LTD problem.

Typical decomposition of the joint problem is that one first decides on the set of lightpaths (TD) which constitute the links for the logical layer, and then tries to establish the set of lightpaths in the network (RWA problem) and determine a (sub)optimal routing at the logical layer which, e.g., minimises the maximum traffic load on the logical links (TR problem). Note that the joint LTD problem involves two routing problems (see Fig. 3.1): routing of lightpaths as a part of the RWA problem and routing of (packet) traffic (TR problem).

The optical layer provides enormous capacity, while the electronical layer allows much finer granularity. In the LTD problem, the objective is to find such set of lightpaths that can be realised at the optical layer and for which a routing at the logical layer minimises, e.g., the mean delay in the network. During the network optimisation a tradeoff must be made between the huge capacity in the fibre with the electro-optic conversion and electronical processing time at the logical layer.

3.2 Problem Formulation

One can find several formulations of the LTD problem in the literature and different papers also tend to use slightly different terminologies. In this work we consider the following LTD problem:

Problem: Logical Topology Design [LTD]

For a given,

- *physical network* $\mathcal{G} = (V, E)$
- *traffic matrix* T

find

- *a set of lightpaths (TD)*
- *traffic routing at the logical topology (TR)*
- *routing and wavelength assignment of the lightpaths (RWA)*

that minimises, e.g. the mean packet delay in the network.

In [LMM00, MNG+01, MGL+02] a slightly different approach is considered, where the authors assume that the routing policy for the logical layer is already given (e.g. OSPF,BGP) and thus the TR step becomes trivial:

Problem: Logical Topology Design [LTD2]

For a given,

- *physical network* $\mathcal{G} = (V, E)$
- *traffic matrix* T
- *multi-hop routing strategy*

find

- *a set of lightpaths (TD)*
- *routing and wavelength assignment of the lightpaths (RWA)*

that minimises, e.g. the mean packet delay in the network.

The authors also argue that WA step can be neglected as the new transmission equipment is capable of handling a very high numbers of wavelength channels and if necessary the wavelength continuity constraint can be relaxed by using wavelength converters in some nodes. Furthermore, in [MNG+01] the multicast traffic flows are also considered.

3.3 MILP Formulation

In this section we give an MILP formulation for the joint LTD problem where the objective is to minimise the average number of optical hops a packet traverses in a WR optical network without wavelength conversion. The main difference to the most formulations is that in [KS01] and Publication 2 any number of physical fibres is allowed between each node pair, as well as, any number of lightpaths between each node pair. Furthermore, the formulation includes all three subproblems: TD, RWA and TR. The used notation and variables, following [RS96, DR00, LMM00, BM00b, KS01, MGL+02, PKG02a, PKG02b], is presented in Table 3.1. With these definitions the LTD problem can be formulated as follows:

Objective: minimise the average number of hops, i.e.,

$$\min \frac{1}{\sum_{(s,d)} \lambda^{(sd)}} \sum_{(i,j)} \lambda_{ij}, \tag{3.1}$$

Subject to:

Topology Definition: (logical layer)

(links in) $$\sum_j b_{ji} \leq \Delta^{(\mathrm{in})}_{\max}, \ \forall\, i \tag{3.2a}$$

(links out) $$\sum_j b_{ij} \leq \Delta^{(\mathrm{out})}_{\max}, \ \forall\, i \tag{3.2b}$$

(value range) $$b_{ij} \in \{0, 1, 2, \ldots\}, \ \forall\, (i, j) \tag{3.2c}$$

Traffic Routing: (logical layer)

$$\text{(congestion)} \qquad \lambda_{ij} \leq \lambda_{\max} \cdot b_{ij},\ \forall\,(i,j) \tag{3.2d}$$

$$\text{(total flow)} \qquad \lambda_{ij} = \sum_{(s,d)} \lambda_{ij}^{(sd)},\ \forall\,(i,j) \tag{3.2e}$$

$$\text{(existance)} \qquad \lambda_{ij}^{(sd)} \leq \lambda^{(sd)} b_{ij},\ \forall\,(i,j),\,(s,d) \tag{3.2f}$$

(flow conservation)

$$\sum_j \lambda_{ij}^{(sd)} - \lambda_{ji}^{(sd)} = \begin{cases} \lambda^{(sd)}, \text{if } i = s, \\ -\lambda^{(sd)}, \text{if } i = d, \\ 0, \text{otherwise.} \end{cases} \ \forall\,(s,d),\,i \tag{3.2g}$$

$$\text{(value range)} \qquad \lambda_{ij}^{(sd)} \geq 0,\ \forall\,(i,j),\,(s,d) \tag{3.2h}$$

Routing and Wavelength Assignment: (optical layer)

$$\text{(channel assignment)} \qquad \sum_k c_{ij}^{(k)} = b_{ij},\ \forall\,(i,j) \tag{3.2i}$$

$$\text{(consistency)} \qquad c_{ij}^{(k)}(l,m) \leq c_{ij}^{(k)},\ \forall\,(i,j),\,(l,m),\,k \tag{3.2j}$$

$$\text{(distinct channel)} \qquad \sum_{ij} c_{ij}^{(k)}(l,m) \leq p_{lm},\ \forall\,(l,m),\,k \tag{3.2k}$$

(lightpath continuity)

$$\sum_{k,m} c_{ij}^{(k)}(l,m) - c_{ij}^{(k)}(m,l) = \begin{cases} b_{ij}, \text{if } l = i, \\ -b_{ij}, \text{if } l = j, \\ 0, \text{otherwise.} \end{cases} \ \forall\,(i,j),\,l \tag{3.2l}$$

$$\text{(hop constraint)} \qquad \sum_{lm} c_{ij}^{(k)}(l,m) \leq h_{ij},\ \forall\,(i,j),\,k \tag{3.2m}$$

$$\text{(value range)} \qquad c_{ij}^{(k)} \in \{0,1\},\ \forall\,(i,j),\,k \tag{3.2n}$$

$$c_{ij}^{(k)}(l,m) \in \{0,1\},\ \forall\,(i,j),(l,m),k \tag{3.2o}$$

Note that the formulation allows dividing the logical layer traffic between a pair of nodes to any number of flows. However, in practice it is very unlikely that the network operators will allow this but instead use one (or few) route(s) for all the traffic between each $s-d$ pair. Not allowing the division, however, makes the traffic routing subproblem even harder to solve. Therefore, in the LTD problem formulations it is typically allowed to split each traffic flow to any number of routes.

Alternative Objective Functions

The presented MILP formulation minimises the average number of hops a packet takes [BM00a], i.e. the average number of times a packet needs to be processed at the electronical layer before it reaches its destination. Parameter $\lambda_{\max}$ defines the maximum allowed level of link load. Note that objective (3.1) is a linear function, as the factor in front of the summation is a constant for a given traffic matrix.

Alternatively one can minimise the maximum congestion $\lambda_{\max}$ as in [RS96, LMM00, MGL$^+$02], where the authors have limited themselves to

constant	explanation
p_{ij}	**number of physical fibres** $i \rightarrow j$, 0 if none.
$\lambda^{(sd)}$	**average traffic load** in logical layer from s to d, e.g. pkt/s.
h_{ij}	**physical hops constraint**, the number of links a lightpath $i \rightarrow j$ can traverse.
$\Delta_{\max}^{(\text{in})}$	**max. logical in degree**, i.e. number of optical receivers.
$\Delta_{\max}^{(\text{out})}$	**max. logical out degree**, i.e. number of optical transmitters.
$\lambda_{\max}$	**max. congestion** in the logical layer, $\lambda_{\max} \leq \max\limits_{i,j} \lambda_{ij}$.
variable	**explanation**
b_{ij}	**number of lightpaths** $i \rightarrow j$.
$c_{ij}^{(k)}$	**number of lightpaths** $i \rightarrow j$ **using the wavelength channel** k.
$c_{ij}^{(k)}(l, m)$	**number of lightpaths** $i \rightarrow j$ **using the wavelength channel** k **on link** $l \rightarrow m$.
$\lambda_{ij}^{(sd)}$	**proportion of traffic** from s to d routed through lightpath $i \rightarrow j$.
λ_{ij}	**virtual traffic load** in lightpath $i \rightarrow j$, consists of fractions $\lambda_{ij}^{(sd)}$.

Table 3.1: Notation.

the case of a single fibre per link and at most one lightpath between any node pair. With these restrictions the (almost) identical formulation turns out to be an MILP problem. These restrictions are dropped in formulation presented in [KS01].

Also in [PKG02a, PKG02b, KPG02] the authors have proposed to primarily minimise the maximum congestion $\lambda_{\max}$. As this objective function typically leaves a plenty of freedom on how to deal with non-congested links/lightpaths the authors propose to minimise the average packet hop distance as a secondary objective once the minimal $\lambda_{\max}$ has been obtained. The problem that the presented MILP formulation allows the traffic to bifurcate onto numerous lightpaths in arbitrary proportions is also addressed in [PKG02a].

Furthermore, in the designing phase one could ask how many electronical interfaces are needed to support a given traffic matrix, i.e. the number of electronical interfaces could be the objective function to be minimised. Taking one step further would allow one to install new fibres between the nodes leading to physical topology design problem.

Other Versions of the LTD Problem

In [LA91, RS96] the authors consider a version of the problem where the number of transmitters and receivers that must be used is fixed. Consequently, the total number of lightpaths to be established is fixed as well. That is, Eqs. (3.2a) and (3.2b) have equality,

(links in) $$\sum_j b_{ji} = \Delta_{\max}^{(\text{in})}, \quad \forall\, i \tag{3.3}$$

(links out) $$\sum_j b_{ij} = \Delta_{\max}^{(\text{out})}, \quad \forall\, i \tag{3.4}$$

In a general case, the number of transmitters and receivers can vary between the nodes. This corresponds to the case when the number of transmitters and receivers limit the optimal solution.

3.4 Approximate Solutions for the LTD Problem

Decomposition of LTD

The MILP formulation presented is generally intractable and one must try other approaches to tackle the problem. A straightforward approach is to solve the joint LTD problem in parts. A commonly used approach is to decompose the problem into the following subproblems (see Fig. 3.1):

i) Topology Definition (TD), i.e. Eqs. (3.2a)-(3.2c).

ii) Traffic routing (TR), i.e. Eqs. (3.2d)-(3.2h).

iii) Lightpath RWA, i.e. Eqs. (3.2i)-(3.2o).

Fixing the logical topology (LT), i.e. fixing the lightpaths, is in a way the hardest step to take as one does not know the final value of the objective function until the other subproblems, traffic routing and lightpath establishment, are solved. Nonetheless, one is supposed to decide on the logical topology before these later steps are taken based on the available information. For the logical topology we have two requirements. Firstly, the logical topology must be realisable, that is one must be able to solve the resulting RWA problem without violating the physical constraints (e.g. not exceeding the number of wavelengths available). Secondly, the resulting logical topology should allow an efficient routing at logical layer (e.g. IP layer).

Basically, one can first fix some LT and then try to find a feasible RWA and TR for it. This can be repeated iteratively; the LT can be modified based on the current set of lightpaths and then steps ii) and iii) can be repeated again (see, e.g. [PKG02b]).

An often used approach is to assume that the maximum logical in/out degrees limit the number of lightpaths (Eqs. (3.2a) and (3.2b)) and, consequently, the resulting RWA problem can be easily solved. Thus, one computes a solution to the TD and TR problems and neglects the constraints set by the RWA problem. For example, in [RS96, PKG02a, PKG02b, KPG02] the authors propose decomposition of the joint LTD problem into the three subproblems which can solved separately in sequence and present an MILP formulation for the joint TD and TR problem.[1]

On the other hand, in LTD approaches based on maximisation of single-hop traffic the traffic routing step (TR) is neglected and the algorithms typically compute TD and RWA. The single-hop approximation is discussed in more detail in the next section.

Single-Hop Approximation

Single-hop traffic corresponds to traffic flows that reach their final destination in one logical hop, i.e. there is a direct lightpath from the source node to the destination node.

[1] In [PKG02a, PKG02b] authors refers to joint TD and TR problem as “LTD” problem.

In [LA91] and later, e.g., [ZA95, RS96, BYC97, BM00b], the authors have proposed maximisation of the single-hop traffic as an alternative objective function. This simplifies the joint problem considerably by allowing one to neglect the traffic routing at the logical layer when deciding on the set of lightpaths. In [ZA95] the single-hop traffic maximisation problem is referred to as the *CP* problem[2]:

Problem: Single-Hop Maximisation [CP]

> *For a given,*
>
> – *physical network* $\mathcal{G} = (V, E)$
>
> – *traffic matrix* T
>
> *find*
>
> – *a set of lightpaths (TD)*
>
> – *routing and wavelength assignment of the lightpaths (RWA)*
>
> *that maximise the total volume of packet traffic reaching its destination using a single lightpath.*

In [ZA95] also a simple greedy heuristic algorithm, called *CP1*, to solve the single-hop maximisation problem is presented. The algorithm creates one lightpath at a time between such nodes where the volume of traffic on single-hop increases the most.

In Publication 2, we propose two alternative greedy heuristics. The first heuristic algorithm, iterative *CP1*, is a combination of *CP1* and any static RWA algorithm. The static RWA algorithm is used to "pack" the current lightpaths more efficiently when needed in order to allow *CP1* to establish additional lightpaths in the network. The second heuristic algorithm, denoted with *CP1e*, is similar to *CP1* but uses a novel dynamic order in which it sets up the lightpaths. Both proposed algorithms are shown to be superior to *CP1*.

Multihop Traffic and Other Related Approaches

If the solution to LTD problem is based on single-hop traffic maximisation and the proportion of multihop traffic is considerable, the result may be far from optimal. In [BM00b, LMM00] the authors have proposed heuristic algorithms which also take into account the multihop traffic. The shortcoming of these algorithms is that they allow only a single lightpath to be established between any $s - d$ pair.

Furthermore, in [RS96] an algorithm called TILDA is proposed which creates a logical topology without any a priori knowledge of the actual traffic matrix. Instead it is assumed that the traffic matrix exhibits a high level of locality and thus the algorithm starts by creating a lightpath from each node to all neighbouring nodes. After that the algorithm tries to establish lightpaths to nodes which are two hops away etc.

Alternatively one can relax the MILP formulation. In [RS96, LMM00, KS01] it is proposed that one first solves a relaxed LP problem and then rounds the solution variables to the nearest integers according to some

[2]CP is assumed to refer to "connectivity problem", as in [LA91].

procedure. The resulting algorithm is referred to as a *rounding heuristic*, *LPLDA*, or *LPLTDA*, depending on context.

Summary of Solutions to LTD Problems

In summary, the proposed solutions to LTD problem can be categorised as follows:

- **MILP formulations**: A complete formulation of the whole problem as an MILP problem [RS96, DR00, LMM00, BM00b, KS01, MGL$^+$02, PKG02a, PKG02b]. The most formulations make assumptions which lead to somewhat easier problems, while in [KS01] an exact formulation taking into account multiple fibres and logical links as well as RWA is presented. Generally MILP formulations, however, lead to intractable problems.
- **Relaxed MILP formulation**: The integer constraints on b_{ij}, $c_{ij}^{(k)}$ and $c_{ij}^{(k)}(l, m)$ are relaxed and the problem is first solved as a normal LP-problem and then the solutions variables are rounded using some procedure to integer values [RS96, KS01].
- **Neglecting RWA**: Assuming that the logical degrees are the real constraint and one can neglect RWA constraints (3.2i)-(3.2o) first and then later find a feasible set of routes and wavelength assignment [RS96, LMM00, MGL$^+$02] (and also [LA91]). This is a valid approach when the resulting static RWA problems can be solved.
- **Single-hop formulation**: Consider first only the amount of single-hop traffic and solve the traffic routing problem [ZA95, RS96, BYC97, BM00b]. In MILP formulation, this means that TR subproblem and the objective function are replaced by a simpler set of linear equations.
- **Multihop heuristics**: Multihop heuristic algorithms try to set up such lightpaths that the multihop traffic also benefits from [BM00b, LMM00].

3.5 Maximisation of Single-Hop Traffic

One possibility to simplify the MILP formulation is to consider only the total volume of the traffic carried with one optical hop as proposed in [LA91, ZA95]. That is, we assume that one is only interested in the traffic which reaches its destination without electronical processing at the intermediate nodes. Then we do not have to determine the routes at the logical layer, which simplifies the problem considerably. In this case, the constraints (3.2d)-(3.2h) defining the traffic routing (TR) can be replaced with,

$$\text{(single-hop traffic)} \qquad \lambda_{\text{one}} = \sum_{(s,d)} \lambda^{(sd)} \cdot b_{sd}. \tag{3.5}$$

As a new objective we try to maximise the single-hop traffic, i.e.

$$\max \lambda_{\text{one}},$$

which is clearly a linear function. Note that this formulation leaves the traffic routing (TR) at the logical layer open, but fixes the logical topology (LT) and routing and wavelength assignment (RWA). Thus, solving the "single-hop maximisation" problem gives a logical topology for which one must solve, exactly or approximately, the traffic routing problem.

This approach of decomposing the joint problem was first proposed in [ZA95], where the MILP formulation assumed a fixed set of paths consisting of (one of) the shortest paths for each node pair. The authors named the problem as the *CP* problem. When the set of lightpaths is determined in order to maximise the total volume of single-hop traffic one can use the following principle:

Principle 1 (Set up fully used direct lightpaths first)
Let t_{sd} be the traffic flow $s \to d$ and let t'_{sd} denote the traffic which requires exclusively the total bandwidth of one lightpath, i.e.,[3]

$$t'_{sd} = \lceil t_{sd} \rceil \,.$$

It is clearly advantageous to route the traffic corresponding to the t'_{sd} directly to its destination if possible. This can be seen as a traditional static RWA problem, where the objective is to minimise the number of used wavelengths. These lightpaths are already fully used and thus can be neglected when routing the remaining traffic in the logical layer.

Thus, the solution of the RWA problem can be used as a starting point for any heuristics. The remaining problem is to determine how to route the (residual) traffic matrix,

$$t^{(r)}_{sd} = t_{sd} - \lceil t_{sd} \rceil \,,$$

where each component is less than one. Typically the small traffic flows are combined at some node (traffic grooming) and routed further together.

As was mentioned earlier, the maximisation of the single-hop traffic in context of WR optical networks was initially proposed by Zhang et al. in [ZA95]. First the authors formulated the problem as an MILP problem with some additional restrictions (e.g. the routes were fixed beforehand), which was named as the *CP* problem. Even though the *CP* problem is considerably easier to solve than the whole joint problem, the formulation can still lead to an intractable problem when the number of the network nodes is large. Thus, the authors in [ZA95] also suggested an approach where lightpaths are established on one wavelength layer at a time. The problem of maximising single-hop traffic at one wavelength layer was named as *CP1* and it is, by definition, equivalent to *CP* with one wavelength layer. Once no new lightpath can be set up to the current wavelength layer the traffic matrix is updated by subtracting the traffic carried at the current layer. Then the same steps can be taken for the next layer, if available.

CP1 Heuristics

The authors of [ZA95] also suggested a heuristic algorithm to solve the *CP1* problem, i.e. a greedy heuristic algorithm which assigns lightpaths at

[3] or $\lceil t_{sd} + \delta \rceil$, where δ is some small constant.

Algorithm 5 Heuristic Algorithm for *CP1* [ZA95]

let $\mathcal{N}$ be the set of network nodes
set $t_{sd} \leftarrow \lambda^{(sd)}$ {the initial traffic intensity $s \to d$}
$\mathcal{X} \leftarrow$ (one of) the shortest path p for each node pair $(sd) \in \mathcal{N} \times \mathcal{N}$
set $W \leftarrow 0$
while $W < W_{\max}$ **do**
  set $W \leftarrow W + 1$
  sort $\mathcal{X}$ in the descending order of traffic intensity t_{sd}
  for each $p \in \mathcal{X}$ **do**
    if path p is free at layer W **then**
      assign a lightpath to path p at layer W
      $t_{sd} \leftarrow \max\{0, t_{sd} - C\}$ {remaining traffic}
    end if
  end for
end while

the current wavelength layer in the order defined by the (residual) traffic matrix.

Note that this order agrees with Principle 1, namely the greedy algorithm sets up first such lightpaths $s \to d$ which will be used solely by the single-hop traffic $s \to d$. Once no more lightpaths can be established to the current layer the traffic matrix is updated and the algorithm moves to the next layer. This is repeated until the last available wavelength layer is reached. Formally the heuristic algorithm is described in Algorithm 5. Note that the Algorithm 5, as presented in [ZA95], assumes $\Delta_{\max}^{(\text{in})} = \Delta_{\max}^{(\text{out})} = \infty$, but it is straightforward to extend the algorithm to the case of finite $\Delta_{\max}^{(\text{in})}$ and $\Delta_{\max}^{(\text{out})}$.

Heuristic LTD Algorithm HLDA

In [RS96], another heuristic LTD algorithm called HLDA is proposed with the aim of maximising the amount of single-hop traffic. Instead of creating lightpaths to one wavelength layer at a time, at each step HLDA assigns the lightpath the lowest available wavelength, if any. Furthermore, in its original form, it uses a heuristic rule to decrease the "remaining traffic" by amount of the next highest traffic intensity. One could as well decrease the amount of remaining traffic by the capacity of one wavelength channel, similarly as *CP1* does. The last step, which uses the remaining resources, can be performed with other algorithms as well.

Iterative *CP1*

In Publication 2, we propose an iterative approach to improve the solutions obtained by *CP1* algorithm. In particular, the *CP1* heuristics can be improved by using a static RWA algorithm to "pack" the current LT when the *CP1* algorithm is no longer able to set up more lightpaths. Several good and fast algorithms for the static RWA problem are presented in Chapter 2. If the used static RWA algorithm manages to find more "space", then the *CP1* heuristic can continue and establish additional lightpaths. This can be repeated until no new lightpath can be established in the network.

Algorithm 6 Heuristic LTD algorithm *HLDA* [RS96]

```
let $\mathcal{N}$ be the set of network nodes and $\lambda^{(sd)}$ the traffic intensities $s \to d$
set $t_{sd} \leftarrow \lambda^{(sd)}$ {the residual traffic intensities $s \to d$}
set $\delta_O^s = \Delta_{\max}^{(\text{out})}$ and $\delta_I^d = \Delta_{\max}^{(\text{in})}$ for all $s$,$d$
loop
    find $(s,d) = \arg\max_{sd} t_{sd}$
    if $t_{sd} = 0$ then
        break
    end if
    if $\delta_O^s \neq 0$ and $\delta_I^d \neq 0$ then
        find the lowest available wavelength on the shortest (propagation-delay)
        path between $s$ and $d$ (scan paths sequentially if several)
        if wavelength available then
            establish lightpath $s \to d$
            find the next highest traffic intensity, $(s'd') = \arg\max_{s'd' \neq sd} t_{s'd'}$,
            set $t_{sd} = t_{sd} - t_{s'd'}$
            set $\delta_O^s = \delta_O^s - 1$ and $\delta_I^d = \delta_I^d - 1$
        else
            set $t_{sd} = 0$
        end if
    else
        set $t_{sd} = 0$
    end if
end loop
establish as many lightpaths as possible at random
```

The idea is simple and is formally presented in Algorithm 7. It is clear, that as the iterative version only sets up additional lightpaths to the solution(s) of the basic *CP1*, the amount of single-hop traffic will never be less than what *CP1* alone would manage to set up. Note that, as the RWA algorithm only creates lightpaths between the given set of $s - d$ pairs, the solution will never violate the logical degree constraints as long as the original solution is feasible.

Enhanced *CP1*

In addition, in Publication 2 we propose an enhanced version of the *CP1* algorithm which is presented in Algorithm 8. The main difference is that *CP1e* uses k shortest paths instead of one. Also the order in which light-

Algorithm 7 Iterative *CP1*

```
determine a set of lightpaths using CP1
repeat
    re-establish the current LT using, e.g., the static RWA Algorithm 3 ("pack")
    if more than $W$ wavelength layers is used then
        return the previous set of established lightpaths
    end if
    continue with CP1
until no new lightpaths can be established
```

Algorithm 8 Enhanced *CP1*

$\mathcal{X} \leftarrow k$ shortest routes for each $s - d$ pair
let t_{ij} be the (residual) traffic from i to j
let $z(p) = (\,\Delta(p),\ -\min\{1, t_{a_0 a_{\ell(p)}}\},\ -\ell(p),\ a_0,\ \ldots, a_{\ell(p)},\ 0,\ \ldots)$
$W \leftarrow 1$
while $W \leq W_{\max}$ **do**
    $\mathcal{X}' \leftarrow \mathcal{X}$
    repeat
        take path $p \in \mathcal{X}'$ with the smallest $z(p)$ (c.f. Def. 2.2)
        if path p is free at layer W **then**
            set up a lightpath p at layer W
            update: $t_{ij} \leftarrow \max\{0, t_{ij} - 1\}$, where $i = a_0$ and $j = a_{\ell(p)}$
        end if
    until $\mathcal{X}'$ is empty
    $W \leftarrow W + 1$
    $\mathcal{X} = \{p \in \mathcal{X}\ :\ t_{a_1 a_{\ell(p)}} > 0\}$
end while

paths are established is slightly different and resembles closely the ideas behind the layered RWA Algorithm 3.

In particular, we sort the path candidates using the lexicographic order according to Def. 2.2 by defining a unique number sequence for each path $p = (a_0,\ a_1,\ \ldots,\ a_{\ell(p)})$ as follows:

$$S(p) = (\Delta(p),\ -\min\{1, t_{a_0 a_{\ell(p)}}\},\ -\ell(p),\ a_0,\ \ldots, a_{\ell(p)},\ 0,\ \ldots),$$

where $\Delta(p)$ is the number of additional hops the path p uses when compared to the shortest path in the sense of number of hops, t_{ij} is the residual traffic from i to j, $\ell(p)$ is the length of path p (in hops), and the a_i are the node numbers along the path. Note that the term corresponding to the residual traffic matrix is modified to be $\min\{1, t_{ij}\}$ to reflect the fact that one wavelength can carry at most one unit of traffic (i.e. it does not matter which connection is established as long as the lightpath channel is fully utilised). The order of two different paths p_1 and p_2 is clearly well-defined by using the lexicographic order for sequences $S(p_1)$ and $S(p_2)$.

The fact that Algorithm 8 sets up the longer paths (which need more resources) first instead of shorter paths may lead to worse overall results when the problem itself is ill-posed in that the available resources are inadequate with regard to the traffic demand. Otherwise, setting up the longer lightpaths first seems to be a good strategy as they are clearly harder to set up than the shorter lightpaths at later steps of the algorithm.

Note that for each layer the order is dynamical which is not the case for the *CP1* algorithm. Furthermore, as the order of the paths resembles closely the order used in the layered RWA Algorithm 3 the iterative version using Algorithm 3 to reconfigure the current LT is unlikely to give any improvement. However, using a more sophisticated RWA algorithm may turn out to be successful.

Connectivity
When blindly maximising the single-hop traffic, the solution does not necessarily result in a connected logical topology. The following approaches have been suggested in the literature to guarantee a connected logical topology:

i) In [ZA95] it is proposed that before filling the last wavelength layer one makes sure that the graph is connected by adding necessary lightpaths. Then the algorithm proceeds normally and fills the last wavelength layer normally.

ii) In [RS96] and [BM00a] it is proposed that if the (maximum) logical degree is greater than the physical degree then one initially sets up a lightpath to each physical link before establishing any other lightpaths. This clearly ensures a fully connected logical topology. If the traffic is highly localised this should not hinder the solution much. Note that if the heuristic algorithm solving LT includes RWA, as is the case with, e.g. [RS96], one should not assign a wavelength for these short lightpaths yet but instead just reserve one channel, and fix the wavelength when the number of free channels in the corresponding link decreases to one.

3.6 Summaries

Summary of Publication 2
The topic of this publication is logical topology design (LTD) problems, where the emphasis is on studying alternative greedy heuristic algorithms in order to maximise the single-hop traffic. The MILP formulation of LTD leads to computationally intractable problems for any network of reasonable size and leaves approximate and heuristic approaches as the only possible practical solution. Maximisation of the single-hop traffic resembles closely the objectives set by the logical topology design problem and seems to be a good alternative design objective.

In [ZA95] a simple and robust algorithm, *CP1*, was presented for the maximisation of the single-hop traffic. In Publication 2, the present author proposes two improved versions of the *CP1* algorithm: *CP1i* (iterative) and *CP1e* (enhanced). The iterative version combines *CP1* with any static RWA algorithm, while *CP1e* incorporates new heuristic rules which improve the performance of the algorithm when compared to the basic *CP1*. By means of numerical simulations both versions are shown to improve the performance of *CP1*.

Author's Contribution to Publication 2
Publication 2 is the sole work of the present author.

4 DYNAMIC ROUTING AND WAVELENGTH ASSIGNMENT PROBLEM

4.1 Introduction

In this chapter, it is assumed that traffic is not static but that the lightpath requests arrive randomly according to some traffic process. Hence, the routing and wavelength assignment constitutes a typical stochastic decision-making problem. When a certain event occurs, one has to decide on some action. In our case, the set of possible actions is finite: either reject the call or accept it and assign a feasible combination of route and wavelength (RW) to it. A feasible RW combination is such that along the route from the source to destination the wavelength is not already in use on any of the links. If no feasible RW combination exists, the call is unconditionally rejected. Furthermore, the accepted connections cannot be interrupted.

In general, one is interested in the optimal policy which maximises or minimises the expectation (infinite time horizon) of a given objective function. Here we assume that the objective is defined in terms of maximising or minimising some revenue or cost function. The cost may represent e.g. the loss of revenue due to blocked calls, where different revenue may be associated to each type of call.

The schemes considered under dynamic traffic can be divided into two cases: reconfigurable and non-reconfigurable. If it is possible to reconfigure the whole network when blocking would occur, the blocking probability can be considerably reduced. Such an operation, however, interrupts all (or at least many) active lightpaths and requires a lot of coordination between all the nodes. Thus, the reconfiguration seems infeasible in large networks. In any case, the reconfiguration algorithm should try to minimise the number of reconfigured lightpaths in order to minimise the amount of interruptions in the service [MM99].

The other case occurs when active lightpaths may not be reconfigured. In this case, it is important to decide which route and wavelength are assigned to an arriving connection request in order to balance the load and minimise the future congestion in the network.

As in the previous chapter, we consider wavelength-routed networks (WRN) where the network nodes are assumed to be WSXC's, hence without any capability to do wavelength conversions. Furthermore it is assumed that reconfiguration of network is not allowed, i.e. once a request has been accepted and a lightpath set up we are not allowed to tear it down beforehand.

Many reasonably well working heuristic policies have been proposed in the literature, such as the first-fit wavelength and most-used wavelength policies combined with shortest path routing or near shortest path routing, see e.g. in [KA98, MA98, RS95], while in [SB97] a wavelength assignment algorithm for a fixed routing is presented. Furthermore in [BH96, SB99, RRP99, ZRP98a, ZRP98b, Bir96, SB97] several approximate methods are proposed for estimating the blocking probability with a fixed routing. Some of the heuristic algorithms are described in Section 4.3. Common to all these heuristic policies is that they are rather simple. The choice of the action to be taken at each decision epoch can usually be described in simple terms and does not require much computation. These algorithms, however, do not take into account the traffic characteristics, such as unequal costs of different requests or inhomogeneous arrival rates.

4.2 Problem Formulation

Formally, the dynamic RWA problem can be stated as follows:

Problem: Dynamic Routing and Wavelength Assignment [D-RWA]

> *For a given,*
>
> – *physical network $\mathcal{G} = (V, E)$, where each link has certain number of bidirectional fibres (i.e. fibre pairs), and*
>
> – *lightpath requests pattern (either unidirectional or bidirectional connections)*
>
> *where active connections yield revenue, (or alternatively the blocked connection requests generate costs)*
>
> *find dynamical control for the network such that the expected revenues are maximised (or alternatively the costs are minimised).*

Connection requests between a given source-destination pair with a certain revenue per a carried call constitute a *traffic class*, which is indexed by k, $k \in \mathcal{K}$, where $\mathcal{K}$ is the set of all traffic classes. In the basic form of the problem all traffic classes have the same unit revenue per a carried call and one can as well try to minimise the mean blocking probability, but in a general case the revenues may be different and correspond to, e.g. different levels of QoS or other similar factors.

When the arrival process of class-k calls is a Poisson process with intensity λ_k, the holding times of those calls are distributed exponentially with mean $1/\mu_k$, and the revenue rate per active class-k connection is w_k, the system constitutes a Markov Process and the problem of determining the optimal policy belongs to the class of Markov Decision Processes (MDP) described, e.g. in [Tij94] and [Dzi97]. Section 4.4 contains a brief introduction to the MDP theory.

A *policy*, usually denoted with α, defines for each possible state of the system and for each traffic class k of an arriving call which of the possible actions is taken. Unless there are not enough resources available and the request is blocked, the RWA algorithm typically sets up a lightpath in the network, i.e. the algorithm unconditionally accepts requests whenever possible. This is, however, not always the optimal policy.

In this thesis, the dynamic routing and wavelength assignment problem (D-RWA) is studied mainly in the setting of Markov Decision Processes (MDP). The application of MDP theory in the context of routing problems is not new. For example, Krishnan and others have applied the MDP theory with traditional circuit switched networks [Kri90, ZAA$^+$97, Rum00, Ahl00]. The same problem arises in the context of WDM networks as with traditional circuit switched networks, i.e. the astronomical size of the state space makes it impossible to solve the optimal policy in most cases. Hence, more or less heuristic algorithms are the only option in practice.

Within MDP framework the optimal policy can be obtained using so-called policy iteration, where, as the name says, one tries to find the optimal policy by starting from some policy and iteratively improving it (see e.g. [Tij94, Dzi97]). The policy iteration is known to converge rather quickly to the optimal policy. Even the first iteration often yields a policy which is rather close to the optimal one. In practice, it is seldom possible to go beyond the first iteration. Also in this work, we will restrict ourselves to the first policy iteration. In order to avoid dealing with the huge size of the state space in calculating the relative state costs needed in the policy improvement step, in Publications 3 and 4 we suggest to estimate these costs on the fly by simulations for the limited set of states that are relevant at any given decision epoch. Informally the first policy iteration proceeds as follows. Upon a lightpath request arrival we make a decision analysis for a small number of alternative actions. Alternative actions are evaluated by a set of numerical simulations for a finite time interval T after making the initial action. During the simulation the consecutive decisions are made according to an arbitrary heuristic policy. The same set of simulations is run for each action and from the simulations we obtain estimates for future costs which corresponds to so-called relative value of state. Then, based on the simulation results, the most promising action is chosen.

The main concern with the first policy iteration is its computational complexity. This can be alleviated to some degree by using so-called importance sampling (IS) technique. The crucial step in IS in general is to find an even nearby optimal twisting for sample probabilities. In Publications 5 and 6, we propose an adaptive importance sampling (AIS) algorithm, which, as the name suggests, adaptively adjusts the chosen twisting parameters towards the global optimum.

4.3 Heuristic Algorithms

Several quick heuristic algorithms for the dynamic RWA problem have been proposed in the literature. Here we briefly present some of them. The first set of algorithms assumes that a fixed set of possible routes for each connection is given in advance. Some papers refer to this as an alternate routing strategy. In practice, the set of routes usually consists of the shortest or nearly shortest paths. These algorithms are greedy and accept the first feasible RW pair they find (first-fit).

- The *basic* (or FF-RW) algorithm goes through all the routes in a fixed order and for each route tries all the wavelengths in a fixed order. The routes are sorted in the shortest route first order. The new connection is routed on the first path on which a wavelength channel is available. Among the available wavelength channels the first feasible channel is selected (see e.g. [KA96, RS95]).

- The *porder* (or FF-WR) algorithm is similar to *basic*-algorithm, but it goes through all the wavelengths in a fixed order and for each wavelength tries all the routes in a fixed order.

- The *pcolour* (or FF-W$^+$R) algorithm works like *porder*, but wavelengths are searched in the order of their current usage instead of a fixed order, so that the most used wavelength is tried first.

- The *lpcolour* (or FF-W$^+$R$^+$) algorithm is the "smartest" algorithm in this set. It packs colours, but the primary target is to minimise the number of used links. The algorithm first tries the most used wavelength with all the shortest routes, then the next often used wavelength and so on. If no wavelength works, the set of routes is expanded to include routes having one link more and wavelengths are tried again in the same order.

The above set of heuristic algorithms in slightly different forms are presented, e.g. in [SB97, MA98, KA98].

Another set of heuristic algorithms, referred to as adaptive unconstrained routing (AUR) algorithms, are described in [MA98]. These algorithms search a route based on the current state of the network (dynamic routing) instead of relying on a fixed set of routes, and are thus a little bit slower.

- The *aurpack* is similar to *pcolour*, but without the limitations of a fixed set of routes, i.e. routes of any length are acceptable.

- The *aurexhaustive* finds a shortest route for each wavelength (if possible) and chooses the shortest RW pair among them, i.e. it is identical to *lpcolour* except that the set of routes is not fixed.

Thus, AUR-algorithms will search for a free route dynamically based on the current state of the network. There is no need to store possible routes (which without any limitations can form a very large set) in advance. Other heuristics are also given in [MA98], e.g. *random* (tries wavelengths in random order) and *spread* (tries least used wavelength first), but they were reported to work worse than the ones described above.

4.4 Markov Decision Processes

The theory of Markov decision processes (MDP) is fundamental to the analysis of many stochastic systems. It deals with stochastic systems where decisions are made and the aim is to find an optimal strategy in some sense. In this section a brief introduction to the MDP theory with main results is given. A more thorough treatment can be found for example in [Ros70, Tij94, Dzi97].

Markov Chain

Stochastic processes with a discrete state space can generally be classified into two categories, namely discrete time and continuous time processes [Law95, Coo81, Saa61]. Assume that the set of states where the system can be is finite or at least countable. A discrete time Markov process (or Markov chain) is a stochastic process whose state at time $t+1$ only depends on its current state at time t. This *memoryless property* is fundamental to Markov processes. Formally, let X_t denote the state of the process at time t. Process

X_t is a Markov chain if

$$\mathrm{P}\{X_{t+1} = x | X_t = x_t, X_{t-1} = x_{t-1}, \ldots\} = \mathrm{P}\{X_{t+1} = x | X_t = x_t\}.$$

The Markov chain is defined by the initial state distribution $\boldsymbol{\pi}^{(0)}$ and the transition probability matrix $\mathbf{P}$,

$$\mathbf{P} = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots \\ p_{2,1} & p_{2,2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix},$$

where $p_{i,j}$ is the probability that the system being in state i moves to the state j in the next step. Note that the row sums $\sum_j p_{i,j}$ are equal to one.

The steady state probability distribution, denoted by a row vector $\boldsymbol{\pi}$, satisfies the following equation,

$$\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P},$$

which together with the normalisation requirement, $\sum_i \pi_i = 1$, defines the steady state distribution of the system.

Markov Process

The continuous time Markov process is defined similarly as Markov chains. Let $p(j,t;i,s)$ be the conditional probability that the system will be in state j at time t if it was in state i at time s. Then the system is said to be a Markov process if the Chapman-Kolmogorov equations hold:

$$p(j,t;i,s) = \sum_k p(j,t;k,u)\,p(k,u;i,s),$$

where $s < u < y$, $i,j \in \mathcal{S}$ and $\mathcal{S}$ is the set of possible states. A similar memoryless condition holds also for the continuous time Markov process, i.e. the future of the system depends only on the current state.

The infinitesimal generator or transition rate matrix $\mathbf{Q}$ of the process,

$$\mathbf{Q} = \begin{pmatrix} q_{1,1} & q_{1,2} & \cdots \\ q_{2,1} & q_{2,2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix},$$

where $q_{i,i} = -q_i = -\sum_{j \neq i} q_{i,j}$ defines the transition probabilities per time unit. The state probability distribution satisfies the differential equation

$$\frac{d}{dt}\boldsymbol{\pi}(t) = \boldsymbol{\pi}(t)\mathbf{Q},$$

where $\boldsymbol{\pi}(t)$ defines the probability distribution at time t. The steady state distribution $\boldsymbol{\pi}$ of Markov process defined by matrix $\mathbf{Q}$ can obtained from

$$\boldsymbol{\pi}\mathbf{Q} = 0,$$

augmented with the normalisation condition. The *sojourn* times in each state i are exponentially distributed with mean

$$\tau_i = 1/\sum_{j \neq i} q_{i,j}.$$

Poisson Process

The Poisson process is probably the most used traffic (arrival) model in data communications as well as in many other areas. The Poisson process is a pure birth process with Markovian property. In this context we talk about arrivals. Briefly, the probability of an arrival during a small time interval of Δt is proportional to the length of time interval and the arrivals in different time intervals are independent, i.e.

$$\mathrm{P}\{\text{one arrival during } \Delta t\} = \lambda \cdot \Delta t + o(\Delta t), \tag{4.1}$$

where constant λ is called arrival rate. From (4.1) it follows that the inter arrival times are exponentially distributed with parameter λ. This can also be expressed by saying that during a fixed length time interval t the number of arrivals N_t obeys Poisson distribution with parameter λt,

$$\mathrm{P}\{i \text{ arrivals during time } t\} = \mathrm{P}\{N_t = i\} = \frac{(\lambda t)^i}{i!} e^{-\lambda t}.$$

Markov Decision Processes

A Markov decision process (MDP) in discrete time is a stochastic process on which a user has some control, i.e. the user can make decisions at certain time steps of the process. Typically, it is assumed that the decision is made after each transition. The chosen decision affects the transition probabilities from the current state and furthermore may incur an immediate cost (or revenue) associated with the decision [Ros70, Tij94, Dzi97]. Formally,

1. After a transition to state i a decision $a \in \mathcal{A}$ is made
2. Each decision,
   - generates an immediate cost $C(i, a)$, and
   - defines the transition probabilities $p_{ij}(a)$.

Typically it is also assumed that the immediate costs are bounded, $C(i, a) < M, \ \forall \ i, a$. Due to the memoryless property the actions taken in some state i can also be assumed to be time independent, i.e. at a given state the same decision is always made. Decisions in each possible state together form a (stationary) policy α,

$$\alpha = \{a_i\}_{i \in \mathcal{S}}.$$

Once a policy α is fixed, the resulting stochastic process is a Markov process (or Markov chain) X_t with some mean cost (or revenue) rate r_i in each state i. The problem is to find the optimal policy α which minimises the expected cost rate r, (or maximises the revenue rate)

$$r = \sum_i \pi_i \cdot r_i.$$

Note that in practice the costs originate, e.g. from the blocked calls and in such a case the immediate cost for decision a in state i corresponds to the expected number of blocked calls before the system moves to another state.

Semi-Markov Decision Processes

A more general framework for decision processes is known as semi-Markov decision processes (SMDP) [Ros70, Tij94, Dzi97]. In SMDP, the durations between the transitions are no longer fixed but obey some probability distribution. Formally, [Ros70]

1. A process starts at time $t = 0$ at some state $i = 0, 1, 2, \ldots$ and some action a is chosen
2. The next state of the process is chosen according to the transition probabilities $p_{ij}(a)$
3. Conditional on the event that the next state is j, the time until the transition $i \to j$ occurs is a random variable with probability distribution $F_{ij}(\cdot\,|a)$.
4. After the transition, steps 2 and 3 are repeated indefinitely.

Similarly, the cost model is extended appropriately, i.e. action a in state i incurs an immediate cost $C(i, a)$ and, in addition, a cost rate $c(i, a)$ until the next transition.

Note that when distributions $F_{ij}(\cdot\,|a)$ are exponential distributions, the resulting system with a fixed policy α is a continuous time Markov process.

Howard's Equations

Howard's equations provide a systematic procedure to obtain the average revenue of a Markov chain or process (i.e. a policy α has been fixed) without first solving the steady state probability distribution.

The *relative value*, or the relative cost, of state i, denoted with v_i, is the difference in the expected costs between a process that starts from state i and another process that starts from the equilibrium. Formally,

$$\begin{aligned} v_i &= \sum_{t=0}^{\infty} \left(\mathrm{E}\left[r_{X_t} | X_0 = i\right] - r\right) \\ &= \sum_{t=0}^{\infty} \left(\sum_{j=1}^{n} \mathrm{P}\{X_t = j | X_0 = i\} r_j - r \right) \\ &= \sum_{t=0}^{\infty} \sum_{j=1}^{n} \left(\mathrm{P}\{X_t = j | X_0 = i\} - \pi_j\right) r_j, \end{aligned}$$

which can be assumed to be finite, as

$$\mathrm{E}\left[r_{X_t} | X_0 = i\right] \overset{t\to\infty}{\to} r \qquad \forall\, i.$$

The difference in costs between the two processes is essentially collected during a transition phase, when the system tends towards the equilibrium from the initial state i.

The discrete time Howard's equation for state i is

$$v_i(\alpha) = r_i(\alpha) - r(\alpha) + \sum_j p_{ij}(\alpha) \cdot v_j(\alpha). \tag{4.2}$$

The formula can be explained in the following way. In the current state, before the departure, the immediate relative value is $r_i - r$. After that the system moves to state j with probability of p_{ij} and from that point onwards the incurred relative values are v_j (due to the lack of memory property). Taking the sum over j includes all the possible cases. Formally,

$$\begin{aligned}
v_i &= \sum_{t=0}^{\infty} \left(\mathrm{E}\left[r_{X_t} | X_0 = i\right] - r\right) \\
&= r_i - r + \sum_{t=1}^{\infty} \left(\mathrm{E}\left[r_{X_t} | X_0 = i\right] - r\right) \\
&= r_i - r + \sum_{t=1}^{\infty} \left(\left(\sum_j p_{ij} \mathrm{E}\left[r_{X_t} | X_1 = j\right] \right) - r \right) \\
&= r_i - r + \sum_j p_{ij} \sum_{t=1}^{\infty} \left(\mathrm{E}\left[r_{X_t} | X_1 = j\right] - r\right) \\
&= r_i - r + \sum_j p_{ij} v_j .
\end{aligned}$$

Hence, there are N linear equations and $N+1$ unknown variables. Any of the relative values v_i can be fixed to be 0 (or any other finite value). If a constant C is added to each relative value, they still satisfy equation (4.2). Hence, a constant offset in relative costs $\{v_i\}$ has no effect on r. Once one of the relative values is fixed, we are left with N unknown variables so that the interesting quantity, the average relative cost r, can be obtained.

Continuous Time Howard's Equations

Howard's equations can also be used with continuous time processes. Denote the relative costs again with v_i, i.e.

$$v_i = \lim_{T \to \infty} \int_0^T \left(\mathrm{E}\left[r_{X_t} | X_0 = i\right] - r\right) dt,$$

where r is the average cost rate in the long run and $\mathrm{E}\left[r_{X_t} | X_0 = i\right]$ is the expected cost rate at time t when the process starts initially from state i (see Fig. 4.1). We can assume that the above limit exists and is finite for each i. We proceed by considering the embedded Markov chain (so-called jump chain). The transition probabilities of the embedded Markov chain are

$$p_{ij} = \begin{cases} \frac{q_{ij}}{q_i}, & \text{when } i \neq j, \\ 0, & \text{when } i = j. \end{cases}$$

Assume that while the system is in state i the rate at which costs are incurred is r_i. Then the equivalent immediate relative value of state i in embedded Markov chain is

$$\frac{r_i - r}{q_i}.$$

Substituting these into (4.2), gives

$$v_i = \frac{r_i - r}{q_i} + \sum_{j \neq i} \frac{q_{ij}}{q_i} v_j .$$

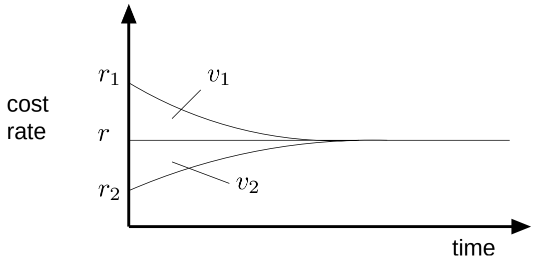

Figure 4.1: Illustration of relative costs v_i of a continuous time MDP.

Recalling that $q_i = \sum_{j \neq i} q_{ij}$, we get the continuous time Howard's equations:

$$r_i - r + \sum_{j \neq i} q_{ij}(v_j - v_i) = 0, \qquad \forall\, i \tag{4.3}$$

Since $q_{ii} = -\sum_{j \neq i} q_{ij}$, the Howard's equations can be expressed as

$$r_i - r + \sum_{j} q_{ij} v_j = 0, \qquad \forall\, i \tag{4.4}$$

As in the discrete time case, one of the relative cost v_i can be arbitrarily fixed, after which the rest of the relative costs and the average cost rate r can be solved.

Equation (4.4) can be explained in the following way. The difference in income rates in the current state equals to $r_i - r$, and the summation gives the transition rates to other states weighted with the appropriate relative values.

As one of the relative values can be given an arbitrary value, we can set $v_0(\alpha) = 0$. Then the set of linear equations can be written in the familiar form,

$$\begin{pmatrix} 1 & -q_{1,2} & -q_{1,3} & \cdots & -q_{1,n} \\ 1 & -q_{2,2} & -q_{2,3} & \cdots & -q_{2,n} \\ \vdots & & \ddots & & \vdots \\ 1 & -q_{n,2} & -q_{n,3} & \cdots & -q_{n,n} \end{pmatrix} \cdot \begin{pmatrix} r \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix},$$

which is an equation of the form $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{x}$ is an unknown vector. It has the formal solution

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b},$$

but the inverse of a huge matrix is difficult to compute. If $n \times n$ matrix $\mathbf{A}$ is sparse and of moderate size then the computation is feasible.

Policy Iteration

Next a systematic procedure to obtain the optimal policy iteratively is presented. The procedure starts from an arbitrary initial policy α and in each round a new better policy is obtained by using the so-called policy improvement step [Dzi97, Tij94]. Recall that the relative values v_i represent

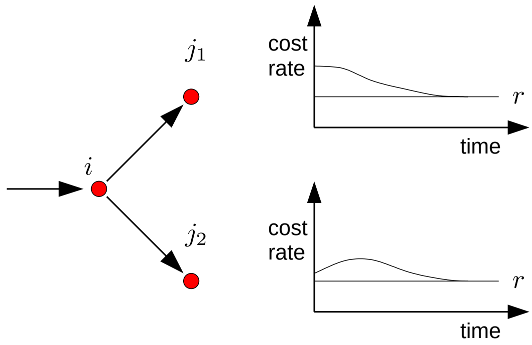

Figure 4.2: The two possible decisions depicted. After the decision is made (and paid the immediate cost) the future (relative) costs, assuming policy α, are given by v_{j_1} and v_{j_2} accordingly.

the difference in the expected future costs for the system starting from certain state i rather than from the equilibrium. The decisions the user makes define the policy.

Consider that the alternative policy α' is used until the next transition occurs. In particular, define the new policy so that the action to be taken in state i is

$$a(i) = \arg\min_{\alpha'} \left\{ r_i(\alpha') + \sum_j q_{ij}(\alpha') \cdot v_j(\alpha) \right\}, \qquad \forall\, i. \tag{4.5}$$

By taking $\arg\min$ over all the possible "temporary" policies α', we get a new policy which is never worse than the current policy. This so-called policy improvement step is repeated until the policy does not change (or the average cost r does not decrease in case there are two or more optimal policies).

Event Model

When MDP is applied to the dynamic RWA problem a decision means deciding whether to accept or reject a lightpath request, and if accepted, which route and wavelength to use. This can be expressed explicitly by defining a policy α as a mapping

$$\alpha : \mathcal{S} \times \mathcal{K} \to \mathcal{S},$$

where $\mathcal{K}$ is a set of possible requests (a stationary policy is implicitly assumed). We require that $\alpha(i,k) \in \mathcal{A}_{i,k}$ where the subset $\mathcal{A}_{i,k} \subset \mathcal{A}$ corresponds to the possible decisions for a type k request in state i. Note that this is just a more convenient notation for MDP in this case.

The policy α defines decisions in each state i for every event k, where a decision may incur some immediate cost $C(i,k,a)$ and possibly a transition $i \to j$ if request k is accepted. Note that in our case a policy is explicitly defined by the state the system is in after the decision. Thus, we can denote with $c_{i,j}(k)$ the cost of the decision to move to state j when type k event occurs in state i.

Furthermore, assuming that lost calls incur the costs the immediate cost of class-k request is w_k if the request is blocked and otherwise zero, i.e.

$$c_{i,i}(k) = \beta_k,$$

where β_k is the average loss of revenues β_k per class-k call. Similarly, the cost rate in state i is

$$r_i(\alpha) = \sum_{k:\alpha(i,k)=i} \lambda_k \cdot \beta_k,$$

i.e. the sum of the arrival rates of the traffic classes which are blocked in given state multiplied with appropriate weights β_k.

The first step in policy iteration in this case goes as follows. When in state i an event k occurs the action $a_{i,k}$ to be taken should be the one which minimises the expected future costs. Assuming that once the action is taken the system reverts back to the standard policy α, the optimal action is clearly

$$a_{i,k} = \operatorname*{arg\,min}_{j \in \mathcal{A}_{i,k}} \{c_{i,j}(k) + v_j(\alpha)\} = \operatorname*{arg\,min}_{j \in \mathcal{A}_{i,k}} \{1_{i=j} \cdot \beta_k + v_j(\alpha)\}, \quad \forall\, i, k, \tag{4.6}$$

where the immediate cost $c_{i,j}(k) = 1_{i=j} \cdot \beta_k$ from blocking the class-k request is explicitly added as it is not included in the relative costs v_j

The equation defines the action for each possible state i and event k, i.e. a new policy α'. For the original policy α the expected relative future costs are known. So, taking a minimum over all the possible actions, a better or at least equal policy is obtained. Repeating the iteration the optimal policy will be finally reached. The policy iteration algorithm is presented in Algorithm 9. Note that the same algorithm holds for the discrete and continuous MDP cases.

Algorithm 9 Policy Iteration for Event Model

```
1: Let α be an arbitrary policy
2: loop
3:    solve Howard's equations for the current policy α ⇒ relative values v_i and
      the average cost rate r for the current policy
4:    if average cost rate r did not improve then
5:       break
6:    end if
7:    determine a new policy α', for each state i and event k as
              arg min_{j∈A_{i,k}} {c_{i,j}(k) + v_j(α)} .
8:    α ← α'
9: end loop
```

Exact Calculation of the Optimal Policy

In [Hyy01] the present author describes a procedure to obtain the optimal policy for dynamic RWA problem. The steps taken can be summarised as follows:

1. First enumerate all possible routes for each traffic class $k \in \mathcal{K}$.
2. Considering one wavelength layer, form so-called connection graph $\mathcal{G}$, where each node corresponds to one route and routes sharing a common link are set as neighbours.
3. Identify the set of states $\mathcal{S}_0$ one wavelength layer can be by enumerating all independent sets of $\mathcal{G}$.
4. Thus, each state $s \in \mathcal{S}$ corresponds to a certain set of active lightpaths.
5. The (global) system state space is a Cartesian product, $\mathcal{S} = \mathcal{S}_0 \times \ldots \times \mathcal{S}_0 = \mathcal{S}_0^W$, from what some states can be pruned by combining them with other states due to symmetry (layers are identical).
6. By applying the event model and policy iteration as described in the previous section the optimal policy can be determined.

However, in practice the huge size of the state space prohibits the determination of the optimal policy. For further details see [Hyy01].

4.5 First Policy Iteration Applied in the Dynamic RWA Problem

In Section 4.4 the D-RWA problem was discussed in the context of MDP theory. First Howard's equations were solved for the current policy and then a method called policy iteration was used to obtain a better policy α'. This was repeated until the optimal policy was obtained and the average revenue rate no longer improved.

As observed in Section 4.4 the size of the state space of any realistic size network is intractable, and though Howard's equations are just a set of linear equations for relative costs v_i and the average cost rate $r(\alpha)$ of the standard policy, their solution cannot be obtained. Hence, it is practically impossible to determine the optimal policy for any realistic size network and other solutions must be sought.

In Section 4.3 several heuristic algorithms were presented. A deficiency in all the presented heuristic algorithms is that they do not take into account the possible additional information about the arrival rates, the distribution of holding times, or the priorities of traffic classes (different costs/revenues). Also the duration of the call when it arrives could be known (for example one channel is reserved for a certain event which lasts exactly two days), which conflicts slightly with the original assumptions about the traffic process (memoryless property). We could of course try to come up with better heuristics which would somehow take into account the additional information, but that means that we would need a new heuristic policy for each new case.

The *first policy iteration* (FPI) algorithm we propose in Publication 3 relies on the MDP theory. In FPI, we take one of the heuristic policies (e.g. one of those presented in Section 4.3) as a starting point and call it the *standard policy*. Then the first round of the policy iteration is taken to make the actual decision. The policy resulting from the first policy iteration is referred to as the *iteration policy*. As stated before, it is not possible to solve all the relative values v_i due to the prohibitive size of the state space. However, at any decision epoch the relative values v_i are needed only for the small set of states $\mathcal{A}_{i,k}$ reachable from the current state (linear function of the number of traffic classes) when class-k arrival has occurred. In Publication 3, we propose *to estimate these values on the fly by means of simulations*.

Briefly, our idea in the FPI is the following: at each decision epoch we make a decision analysis of all the alternative actions. For each of the possible actions, i.e. decision alternatives, we estimate the future costs by simulation. Thus, assuming that a given action is taken we let the system proceed from the state where it is after that action and use the standard policy to make all subsequent decisions in the simulation. The iteration policy is the policy which is obtained when at each decision epoch the action is chosen for which the estimated cost is the minimum. The procedure to obtain the iteration policy, i.e. the policy resulting from the FPI, is presented in Algorithm 10.

By doing the FPI we have two goals in mind. 1) Finding a better D-RWA algorithm which, being computationally intensive, may or may not be calculable in real time, depending on the time scale of the dynamics of the system. 2) Estimating how far the performance of a heuristic algorithm is from the optimal one, even in the case the algorithm is not calculable in real time.

In summary, our contribution in this context is as follows. In Publication 3, we propose using the FPI algorithm for dynamic RWA problems. Publication 4 is a continuation of the same work in which we demonstrate the applicability of the FPI to dynamic RWA problem and show how well the resulting heuristic algorithm automatically adapts to the new situations and takes into account all the peculiarities of the system. Thus, even though the FPI approach is very simple, it is very powerful due to its flexibility.

Relative Costs of States

In the MDP theory, the FPI consists of the following steps: With the standard policy one solves the Howard's equations (see, e.g. [Tij94, Dzi97]) to obtain the relative costs of the states, v_i, which for each possible state i of the system describe the difference in the expected cumulative cost from time 0 to infinity, given that the system starts from state i rather than from the equilibrium. Assume that we are considering costs from the blocked connection requests instead of revenues from accepted connections. Then, given that the current state of the system is i and a type k call is offered, one calculates the costs for different actions using Eq. (4.6), i.e. $\beta_k + v_i$, where $\beta_k = w_k/\mu_k$, for the action that the call is rejected, and v_j, where $j \in \mathcal{A}_{i,k}$ and $j \neq i$, for the case the call is accepted. The set $\mathcal{A}_{i,k}$ is the set of possible states after decision when the current state is i and type k connection is assigned a feasible RW pair or rejected. By choosing always the action which minimises the expected cost, one gets the iteration policy, i.e. the policy resulting from the FPI.

Given that the system starts from state i at time 0, i.e. $X_0 = i$, and the standard policy α is applied for all the decisions, the cumulative costs are accrued at the expected rate $c_t(i)$ at time t,

$$c_t(i) = \mathrm{E}\left[r_{X_t} | X_0 = i\right] = \sum_k \lambda_k \beta_k \mathrm{P}\{X_t \in \mathcal{B}_k | X_0 = i\}, \tag{4.7}$$

i.e. the expected rate of lost revenue, where β_k is average revenue of carried class-k connection and $\mathrm{P}\{X_t \in \mathcal{B}_k\}$ is the probability that at time t the state X_t of the system is a blocking state for class-k calls under the standard policy. When $X_t \in \mathcal{B}_k$, class-k calls arriving at time t are blocked by the standard policy because, either no feasible RW pair exists, or the policy otherwise deems the blocking to be advantageous in the long run. The expected cost rate $c_t(i)$ depends on the initial state i. However, no matter what the initial state is, as t tends to infinity, the expected cost rate tends to a constant r, which is specific to the standard policy, and corresponds to (4.7) with steady state blocking probabilities $\mathrm{P}\{X_t \in \mathcal{B}_k\}$.

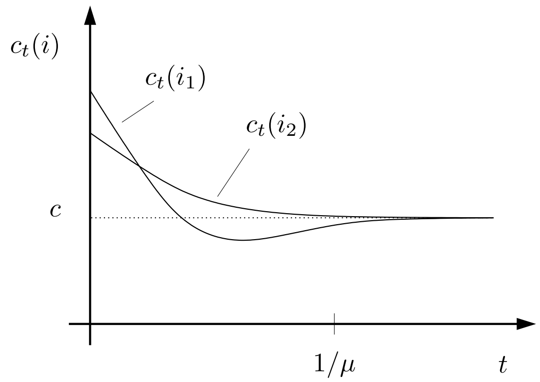

Figure 4.3: Expected costs with different initial choices as a function of time.

The behaviour of the function $c_t(i)$ is depicted in Fig. 4.3 for two different initial values i_1 and i_2. The relative cost v_i is defined as the integral

$$v_i = \int_0^\infty \left(c_t(i) - r\right)\, dt,$$

i.e. the area between the curve $c_t(i)$ and the line at level r. So we are interested in the transient behaviour of $c_t(i)$; after the transient no contribution comes to integral. The length of the transient is of the order $1/\mu$, where $1/\mu$ is the maximum over $\{1/\mu_k\}$, $k \in \mathcal{K}$. After this time the system essentially forgets the information about the initial state. So we can restrict ourselves to an appropriately chosen finite interval $(0, T)$. The actual choice of T is a tradeoff between various considerations as will be discussed later.

One easily sees that in the policy improvement step Eq. (4.6), only the differences of the immediate costs and the values v_i between different states are important. Therefore, in Publication 3 we neglect the constant r in the integral, as it is common to all states, and consider the expected cumulative cost in the interval $(0, T)$ including the immediate cost $c_{ij}(k)$ in the initial decision a to move from state i to state j,

$$V_a = c_{ij}(k) + \int_0^T c_t(j)\, dt. \tag{4.8}$$

Note that the relation between the cumulative costs V_a and the relative values v_j is simply,

$$V_a \approx c_{ij}(k) + v_j + rT,$$

which is accurate for large values of T.

Estimation of the Cumulative Costs by Simulation

In practice, it is not feasible to calculate the cost rate function $c_t(i)$ or the cumulative costs V_a analytically even for the simplest policies. Therefore, we estimate the mean cumulative costs V_a by simulations. In each simulation, the system is initially set in state i and then the evolution of the system is followed for the period of length T, making all the RWA decisions according to the standard policy.

In collecting the statistics, one has two alternatives. Either one records the time intervals when the system is in a blocking state of class-k calls, for all $k \in \mathcal{K}$. If the cumulative time within interval $(0, T)$ when the system is in the blocking state of class-k calls is denoted by $\tau_k(i)$, then the integral in (4.8) is simply

$$\sum_k \lambda_k \beta_k \tau_k(i). \tag{4.9}$$

Alternatively, one records the number $\nu_k(i)$ of blocked calls of type k in interval $(0, T)$. Then we have

$$\sum_k \beta_k \nu_k(i). \tag{4.10}$$

In these equations, we have written explicitly $\tau_k(i)$ and $\nu_k(i)$ in order to emphasise that the system starts from the state i. Both (4.9) and (4.10) give an unbiased estimate for $v_i(T)$. In either case, the simulation has to be repeated a number of times in order to get an estimator with a small enough confidence interval.

The estimates of future costs obtained in the nth simulation replication, being denoted by $\hat{V}_a^{(n)}$, are obtained using (4.9) or (4.10) as the case may be. Then our final estimator for V_a is

$$\hat{V}_a = \frac{1}{N} \sum_{n=1}^{N} \hat{V}_a^{(n)}, \tag{4.11}$$

where N is the number of simulation replications. For policy improvement, the interesting quantity is the difference

$$d(a_1, a_2) = V_{a_2} - V_{a_1},$$

(when $T \to \infty$) for which we have the obvious estimate

$$\hat{d}(a_1, a_2) = \hat{V}_{a_2} - \hat{V}_{a_1}. \tag{4.12}$$

From the samples $\hat{V}_{a_1}^{(n)}$ and $\hat{V}_{a_2}^{(n)}$, $n = 1, \ldots, N$, we can also derive an estimate for the variance $\hat{\sigma}^2(a_1, a_2)$ of the estimator $\hat{d}(a_1, a_2)$

$$\begin{aligned}\hat{\sigma}^2(a_1, a_2) &= \frac{N \sum_n (\hat{V}_{a_2}^{(n)} - \hat{V}_{a_1}^{(n)})^2 - \left(\sum_n \hat{V}_{a_2}^{(n)} - \hat{V}_{a_1}^{(n)}\right)^2}{N^2(N-1)} \\ &= \frac{\hat{S}^2_{a_1,a_2} - (\hat{d}(a_1, a_2))^2}{N-1},\end{aligned}$$

where $\hat{S}^2_{a_1,a_2} = \frac{1}{N} \sum_n \left(\hat{V}_{a_2}^{(n)} - \hat{V}_{a_1}^{(n)}\right)^2$.

The choice between the alternative statistics collection methods is based on technical considerations. Though estimator (4.9) (blocking time) has a lower variance per one simulation replication, it requires much more bookkeeping and the variance obtained with a given amount of computational effort may be lower for estimator (4.10) (blocking events).

The important parameters of the simulation are the length of the simulation period T and the number of simulation replications N used for the estimation of each V_a. In practice, we are interested in the smallest possible values of T and N in order to minimise the simulation time. However, making T and N too small increases the simulation noise, i.e. error in the estimates $\hat{V}_a$, occasionally leading to decisions that differ from that of the true iteration policy, consequently deteriorating the performance of the resulting algorithm.

No matter how the parameters are selected, some uncertainty in the estimators $\hat{V}_a$ is unavoidable. In order to deal with this uncertainty of the estimators $\hat{V}_a$, we do not blindly accept the action with the smallest estimated cost, but give a special status for the decision which would be chosen by the standard policy. Let us index this action with a_0. Based on the simulations we form estimates $\hat{d}(a_0, a)$ for each possible action a.

Eq. (4.6) defines the policy iteration step for state i and arrival k,

$$a(i, k) = \underset{j \in \mathcal{A}_{i,k}}{\arg\min} \left\{1_{i=j} \cdot \beta_k + v_j(\alpha)\right\}, \qquad \forall\, i, k.$$

where th term $1_{i=j} \cdot \beta_k$ corresponds to the blocking and is so-called immediate cost of the action $i \to j$. There is *no uncertainty* involved in the immediate costs, which are either zero if the connection request is accepted, or β_k if the connection request is blocked.

When two alternative actions leading to states j_1 and j_2 are compared, only the difference in immediate costs, $\beta_k \cdot (1_{j_2=i} - 1_{j_1=i})$, is important. Note that the information indicating whether either action blocked the connection request is included in the destination states j_1 and j_2 (the knowledge of i is not necessary). The difference in the number of active connections between the states j_1 and j_2 determines possible blocking (is there one connection less in either state). Hence, the difference in immediate costs can be expressed as a function of j_1 and j_2.

Then, as the decision we choose the action $a = a_{i,k}$ leading to state j which minimises the quantity

$$D(a) = \hat{d}(a_0, a) + \kappa \cdot \hat{\sigma}(a_0, a), \tag{4.13}$$

Algorithm 10 First policy iteration in D-RWA

a connection request between some nodes arrives
let a_0 be the standard policy action, and $a_1, \ldots, a_M$ alternative actions
generate N independent sets of future arrivals, A_j
$D_0 \leftarrow 0$
for $j = 1$ to N **do**
  make initial action $a = a_0$
  run simulation with arrivals A_j
  store the total costs to $V_0^{(j)}$ {Reference costs}
end for
for $i = 1$ to M **do**
  for $j = 1$ to N **do**
    make initial action a_i
    run simulation with arrivals A_j
    store the difference $d^{(j)} = V_i^{(j)} - V_0^{(j)}$
  end for
  $D \leftarrow \text{mean}(\mathbf{d}) + \kappa \cdot \text{std}(\mathbf{d})$
  if $D < D_0$ **then**
    $a \leftarrow a_i$ {set new action}
    $D_0 \leftarrow D$
  end if
end for
a is the action resulting from FPI

where κ is an adjustable parameter. Note that for $a = a_0$ this quantity is equal to 0. Thus, in order for action $a \neq a_0$ to replace the action a_0 of the standard policy, we must have $\hat{d}(a_0, a) < -\kappa \cdot \hat{\sigma}(a_0, a)$, i.e. we require a minimum level of confidence for the hypothesis that a indeed is a better action than a_0. An appropriate value for κ has to be determined experimentally.

If κ is too small, one often switches to an action which is worse than standard action. On the other hand, a too high value of κ prevents the choice of other alternative action totally.

4.6 Importance Sampling in Policy Iteration

In this section we study the possibility to accelerate the cost estimation in first policy iteration (FPI). In the first policy iteration, N different future realisations are generated. The set of possible future events is huge (or actually non-countable) and only a very small number of them can be included in the simulations. Also the revenues or the costs of different realisations may differ a lot. Assuming we are estimating the cost due to lost calls, we would like to favour such realisations which are more likely to cause costs. In particular, it is not useful to sample such realisations which do not give any contribution to the costs. Therefore, in applying importance sampling, we try to make the more important events more probable in the simulation.

Next a brief introduction to the importance sampling is presented, and then the technique is applied to the FPI approach in order to reduce the number of future replications (=samples) required in the simulations[1].

Importance Sampling

Importance sampling (IS) is a well-known method to reduce the variance in Monte Carlo simulations, see e.g. [Hei95, Hee97, Ros00, Rub97]. Suppose we are trying to estimate the expectation of some random variable. In a problematic case, the samples can have a high variance, which leads to a need of many samples or a poor estimate. Such a problem can be avoided to some degree with appropriate variance reduction methods presented e.g. in [Ros00].

Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ be a set of random variables and $h(\mathbf{x})$ some function of them. We are interested in the mean of $A = h(\mathbf{X})$:

$$\theta = \mathrm{E}\left[A\right] = \mathrm{E}\left[h(\mathbf{X})\right].$$

Assume that it is not possible to analytically calculate the above mean, but it is still possible to run simulations in order to get samples of A.

The obvious estimator for θ is obtained by directly taking k samples from the given probability distribution and averaging them,

$$\hat{A} = 1/k \cdot \sum_{i=1}^{k} A_i,$$

where A_i are i.i.d. random variables, $A_i \sim h(\mathbf{X})$. Then,

$$\begin{aligned}
\mathrm{E}\left[\hat{A}\right] &= \theta, \\
\mathrm{V}\left[\hat{A}\right] &= \mathrm{E}\left[(\overline{A} - \theta)^2\right] = \mathrm{V}\left[A_i\right]/k.
\end{aligned}$$

Generally, the smaller the variance of the estimator, the better the estimate is and, e.g. fewer samples are sufficient to achieve a given confidence interval. Several techniques to reduce the variance of the estimator are presented, e.g. in [Ros00]. In this work, we concentrate on the importance sampling where the events having a greater contribution to the expectation are made more probable in the sampling and vice versa. This technique is supposed to give a better estimate with the same number of samples (or even fewer) than the direct estimator. Especially, those $\mathbf{x}$ for which $h(\mathbf{x}) = 0$ can be excluded from the sample space as their contribution to final estimate is zero.

Let $f(\mathbf{x})$ be the probability density function of $\mathbf{X}$. Then,

$$\theta = \mathrm{E}\left[h(\mathbf{X})\right] = \int h(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}.$$

The discrete case is treated identically, but instead of integration an n-fold summation is taken.

Let $g(\mathbf{x})$ be another probability density function for which it holds[2]

$$g(\mathbf{x}) = 0 \qquad \Rightarrow \qquad f(\mathbf{x}) = 0.$$

[1] The importance sampling technique is also applicable when the performance of an arbitrary heuristic algorithm is evaluated.

[2] It would be even better to have, $g(\mathbf{x}) = 0$ iff $h(\mathbf{x})f(\mathbf{x}) = 0$, as it is useless to take samples with no contribution to the estimate.

The requirement guarantees that every possible event in the original distribution, i.e. an event that occurs with a non-zero probability, is also taken into account under the new probability distribution. Denote by $\mathbf{Y}$ a random variable whose pdf is $g(\mathbf{x})$.

Then the quantity θ can be expressed as

$$\theta = \int \frac{h(\mathbf{x})\, f(\mathbf{x})}{g(\mathbf{x})} g(\mathbf{x})\; d\mathbf{x} = \mathrm{E}\left[\frac{h(\mathbf{Y})\, f(\mathbf{Y})}{g(\mathbf{Y})}\right] \tag{4.14}$$

where the quantity $q(\mathbf{x}) = f(\mathbf{x})/g(\mathbf{x})$ in the equation (4.14) is so-called *likelihood ratio*. Thus, the interesting quantity θ can be estimated by generating samples of random variable $\mathbf{Y}$ with pdf $g(\mathbf{x})$ and estimating the expected value of

$$B = q(\mathbf{Y}) \cdot h(\mathbf{X}),$$

where B is so-called *observed random variable*. In other words, when taking the samples of $\mathbf{Y}$ with the new pdf $g(\mathbf{x})$ each outcome $h(\mathbf{x})$ is simply multiplied with the appropriate likelihood ratio $q(\mathbf{x})$ in order to get an unbiased estimator:

$$\theta = \mathrm{E}\left[h(\mathbf{X})\right] = \mathrm{E}\left[q(\mathbf{Y}) \cdot h(\mathbf{Y})\right] = \mathrm{E}\left[B\right]. \tag{4.15}$$

Assuming we have k i.i.d. samples $B_i \sim B$, then an obvious estimator for θ is,

$$\hat{B} = \frac{1}{k} \sum_i B_i.$$

The variance of the observed random variable B is,

$$\mathrm{V}\left[B\right] = \mathrm{E}\left[B^2\right] - \theta^2 = \int_\Omega (q(\mathbf{x})h(\mathbf{x}) - \theta)^2 \cdot g(\mathbf{x})\, d\mathbf{x},$$

which is to be minimised by a proper choice of the importance function. Thus, the optimal choice for new pdf $g(\mathbf{x})$ minimising the variance $\mathrm{V}\left[B\right]$ is the one which also minimises the second moment of B. It is well-known that the optimal biasing is,

$$g(\mathbf{x}) = \frac{|f(\mathbf{x}) \cdot h(\mathbf{x})|}{\int_\Omega |f(\mathbf{x}) \cdot h(\mathbf{x})|\, d\mathbf{x}}.$$

In particular, if $h(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \Omega$, then using the above formula results in a zero variance estimator. This is, however, not practical as it requires the knowledge of the quantity one is interested in. Thus, we are looking for importance functions which are approximately optimal. From the Chebyshev inequality,

$$\mathrm{V}\left[Y^*\right] \geq a^2\, \mathrm{P}\{|Y^* - \theta| \geq a\},$$

it follows that it is important that for all $x \in \Omega$,

$$|p(x)h(x) - \theta p^*(x)| \gg p^*(x),$$

is not true, or otherwise the variance can become very large, even infinite.

Importance Sampling Applied to the FPI Approach

In Publication 6, we apply the IS technique to the FPI approach. The aim is to get a reasonably good estimate about the future costs with fewer samples leading to a faster decision-making process.

Assume that connection requests from each traffic class k, $k \in \mathcal{K}$, constitute a Poisson process with arrival rate λ_k. Let A_k be the random variable representing the number of class-k arrivals during the time interval $(0, T)$.

The Poisson process has the following nice property (see Section 4.4):

Theorem 4.1 *Given that a certain number of arrivals from a Poisson process have occurred during a time interval* $(0, T)$*, these arrivals are uniformly and independently distributed in the same interval.*

This property will be useful when characterising the likelihood ratio in importance sampling within the first iteration framework. The interesting quantity here are the average cumulative costs $\overline{c}$ during a finite time interval $(0, T)$:

$$\overline{c} = \mathrm{E}_f\left[c(\mathbf{X})\right],$$

where $f(\mathbf{x})$ is the pdf of the finite time future events (arrivals and departures) and $c(\mathbf{x})$ is the cumulative incurred costs during the time interval $(0, T)$ with the future realisation $\mathbf{x}$ of the process.

Altering Arrival Rates

The problem with simulating a typical network is that blocking is a rare event. An obvious idea is to increase the arrival rates for some or for all the traffic classes, i.e. instead of using the original arrival rates $\{\lambda_k\}_{k\in\mathcal{K}}$ a new set of arrival rates $\{\lambda_k^*\}_{k\in\mathcal{K}}$ is used, where $\lambda_k^* > 0\ \forall\, k \in \mathcal{K}$. The holding time distributions as well as the revenue rates/average losses of missed calls w_k are kept the same. Clearly the realisations generated with the new arrival parameters have new probabilities, or probability densities $g(\mathbf{x})$ to be more precise.

Let $g(\mathbf{x})$ be the pdf of the finite time future events with altered arrival rates. According to (4.15) the average cumulative costs during the time interval $(0, T)$ then become

$$\overline{c} = \mathrm{E}_g\left[q(\mathbf{X})c(\mathbf{X})\right],$$

where the $q(\mathbf{x})$ is the likelihood ratio. Thus, certain realisations with new arrival parameters are more likely to occur than they used to be, and vice versa. Hence, the cost estimate obtained in a direct way with the new arrival process would give false results. To correct this we must weight the cost from each realisation appropriately with the likelihood ratio, which will depend only on the number of arrivals as will be seen in the following.

For the Poisson process determining the likelihood ratio is indeed fairly easy. The arrival realisations can be classified according to the number of arrivals from each traffic class. Let n_k be the number of class-k arrivals in a given realisation, i.e. $A_k = n_k$. For the Poisson process these arrivals are uniformly distributed in the given time interval $(0, T)$, as stated by Theorem 4.1. Thus, as every realisation $\mathbf{x}$ with the same number of arrivals from each traffic class k are equally likely to occur, we can concentrate on the number of arrivals and neglect the actual arrival times. A more formal proof follows.

Number of Arrivals

The probability that there are n_k class-k arrivals from the original arrival process A_k is

$$\mathrm{P}\{A_k = n_k\} = \frac{(\lambda_k \cdot T)^{n_k}}{n_k!} e^{-\lambda_k \cdot T},$$

i.e. a Poisson distribution with parameter $\lambda_k \cdot T$. The arrivals from the different traffic classes are independent and thus the probability of having $\mathbf{n} = (n_1, \ldots, n_K)$ arrivals from the original arrival processes is simply the product

$$\mathrm{P}\{\mathbf{A} = \mathbf{n}\} = \prod_{k \in \mathcal{K}} \frac{(\lambda_k \cdot T)^{n_k}}{n_k!} e^{-\lambda_k \cdot T} = T^n e^{-\lambda \cdot T} \prod_{k \in \mathcal{K}} \frac{\lambda_k^{n_k}}{n_k!},$$

where $n = \sum n_k$ and $\lambda = \sum \lambda_k$.

Similarly, the same number of arrivals from each traffic class with the new arrival process $\mathbf{A}^*$ would occur with the probability

$$\mathrm{P}\{\mathbf{A}^* = \mathbf{n}\} = T^n e^{-\lambda^* \cdot T} \prod_{k \in \mathcal{K}} \frac{(\lambda_k^*)^{n_k}}{n_k!}.$$

The costs we want to estimate can be written as

$$\overline{c} = \mathrm{E}\left[c(\mathbf{X})\right] = \mathrm{E}\left[\mathrm{E}\left[c(\mathbf{X})|\mathbf{A}\right]\right] = \mathrm{E}\left[\tilde{c}(\mathbf{A})\right],$$

where $\tilde{c}(\mathbf{n}) = \mathrm{E}\left[c(\mathbf{X})|\mathbf{A} = \mathbf{n}\right]$ is the average cumulative costs during the time interval $(0, T)$, when n_k uniformly distributed class-k arrivals, for each $k \in \mathcal{K}$, occur during the given time period.

Similarly as in the continuous case, the importance sampling with arrivals $\mathbf{A}^*$ having a different point probability distribution becomes

$$\begin{aligned}\mathrm{E}\left[\tilde{c}(\mathbf{A})\right] &= \sum_{\mathbf{n}} \mathrm{P}\{\mathbf{A} = \mathbf{n}\} \cdot \tilde{c}(\mathbf{n}) = \sum_{\mathbf{n}} \frac{\mathrm{P}\{\mathbf{A} = \mathbf{n}\}\tilde{c}(\mathbf{n})}{\mathrm{P}\{\mathbf{A}^* = \mathbf{n}\}} \mathrm{P}\{\mathbf{A}^* = \mathbf{n}\} \\ &= \mathrm{E}_*\left[\frac{p(\mathbf{A}^*)}{p^*(\mathbf{A}^*)}\tilde{c}(\mathbf{A}^*)\right],\end{aligned}$$

where the subscript $*$ denotes that the expectation is to be taken with respect to the alternative point probability distribution of the arrivals $\mathbf{A}^*$. Similarly, the likelihood ratio $q(\mathbf{n})$ is

$$\tilde{q}(\mathbf{n}) = \frac{\mathrm{P}\{\mathbf{A} = \mathbf{n}\}}{\mathrm{P}\{\mathbf{A}^* = \mathbf{n}\}} = e^{-(\lambda_k - \lambda_k^*)T} \prod_{k \in \mathcal{K}} \left(\frac{\lambda_k}{\lambda_k^*}\right)^{n_k}.$$

Likelihood Ratio with Future Realisations

Next it will be shown that for any future realisation $\mathbf{x}$ the likelihood ratio $q(\mathbf{x})$ depends only on the number of arrivals from different traffic classes. To be exact, for each $\mathbf{n}$ it holds that

$$\forall\, \mathbf{x} \in \Omega_{\mathbf{n}} \qquad q(\mathbf{x}) = \tilde{q}(\mathbf{n}),$$

where $\Omega_{\mathbf{n}}$ denotes the class of future realisations with $\mathbf{n}$ arrivals from the respective traffic classes.

Formally, let $(\Omega, \mathcal{F}, \mu)$ be a probability triple, i.e. Ω is a sample space, $\mathcal{F}$ its σ-algebra and μ a probability measure $\mu : \mathcal{F} \to \mathbb{R}$ [Wil91, Dud89]. Here the sample space Ω consists of the possible arrivals and departures during the finite time interval $(0, T)$. The sample space is divided into subspaces $\Omega_{\mathbf{n}}$ according to the number of arrivals n_k from each traffic class k, $k \in \mathcal{K}$. According to Theorem 4.1, the probability measure (or rather its density) μ is constant within each $\Omega_{\mathbf{n}}$.

There are two probability measures here: $\int f$ and $\int g$, which, as stated before, have a constant density within each $\Omega_{\mathbf{n}}$, i.e. also the likelihood ratio $q(\mathbf{x})$ is constant: $q(\mathbf{x}) = f(\mathbf{x})/g(\mathbf{x}) = c_f/c_g$ where c_f and c_g are some constants. Furthermore, it holds that

$$\mathrm{P}\{\mathbf{A} = \mathbf{n}\} = \int_{\mathbf{x}\in\Omega_{\mathbf{n}}} f(\mathbf{x}) = \int_{\mathbf{x}\in\Omega_{\mathbf{n}}} c_f, \qquad \text{and}$$

$$\mathrm{P}\{\mathbf{A}^* = \mathbf{n}\} = \int_{\mathbf{x}\in\Omega_{\mathbf{n}}} g(\mathbf{x}) = \int_{\mathbf{x}\in\Omega_{\mathbf{n}}} c_g.$$

Thus, for each $\mathbf{x} \in \Omega_{\mathbf{n}}$ the likelihood ratio $q(\mathbf{x})$ becomes

$$q(\mathbf{x}) = \frac{c_f}{c_g} = \frac{\int_{\mathbf{x}\in\Omega_{\mathbf{n}}} c_f}{\int_{\mathbf{x}\in\Omega_{\mathbf{n}}} c_g} = \frac{\mathrm{P}\{\mathbf{A} = \mathbf{n}\}}{\mathrm{P}\{\mathbf{A}^* = \mathbf{n}\}} = \tilde{q}(\mathbf{n}).$$

Next the likelihood ratio is determined for some simple cases.

In Publications 5 and 6, we consider a constant increase in arrival rates by multiplying them with a common factor $\alpha > 1$

$$\lambda_k^* = \alpha \cdot \lambda_k.$$

Then the likelihood ratio $\tilde{q}$ is

$$\tilde{q} = \prod_{k\in\mathcal{K}} \left(\frac{\lambda_k}{\alpha \cdot \lambda_k}\right)^{n_k} e^{-(\lambda_k - \alpha\lambda_k)T} = C \cdot \left(\frac{1}{\alpha}\right)^n,$$

where $C = \prod_{k\in\mathcal{K}} e^{(\alpha-1)\lambda_k \cdot T} = e^{(\alpha-1)\overline{A}}$, $\overline{A} = \lambda T$ is the expected number of arrivals from the original process, and $n = \sum_k n_k$ is the total number of arrivals in a given realisation. Thus, the likelihood ratio can be written as

$$\tilde{q} = \frac{\left(e^{\alpha-1}\right)^{\overline{A}}}{\alpha^n}.$$

The advantage with this choice of biasing is its simplicity; there is only one constant to be determined. Alternative choices are also possible (see [Hyy01]).

4.7 Adaptive Importance Sampling

Generally, the problem with IS is finding a good biased distribution which minimises the variance of Y^*. In Publications 5 and 6, we develop a novel approach for adaptively adjusting the chosen biasing parameters in FPI with IS approach. We refer to this method as *adaptive importance sampling* (AIS). ,

It turns out that the variance of Y^* can be expressed in terms of the original random variable X and functions $q(x)$ and $h(x)$. Namely, it holds for any function $z(x)$ that

$$\mathrm{E}\left[z(X)\right] = \mathrm{E}\left[q(X^*) \cdot z(X^*)\right], \tag{4.16}$$

and hence,

$$\begin{aligned} \mathrm{V}\left[Y^*\right] &= \mathrm{E}\left[(Y^*)^2\right] - \mathrm{E}\left[Y^*\right]^2 \\ &= \mathrm{E}\left[q^2(X^*)h^2(X^*)\right] - \theta^2 \\ &= \mathrm{E}\left[q(X)h^2(X)\right] - \theta^2, \end{aligned} \tag{4.17}$$

where $q(x)$ is an arbitrary likelihood ratio to be chosen.

Generally, let $\{X_i^*\}$ be a set of biased random variables with the corresponding likelihood ratios $q_i(x)$. Then for any $z(x)$ and for all i it holds that,

$$\mathrm{E}\left[z(X)\right] = \mathrm{E}\left[q_i(X_i^*) \cdot z(X_i^*)\right], \tag{4.18}$$

and, in particular, we have the unbiased estimator for θ,

$$\hat{\theta} = \frac{1}{m}\sum_{i=1}^{m} q_i(X_i^*) \cdot h(X_i^*).$$

From equation (4.18) it follows for $z(x) = q(x)h^2(x)$ that,

$$\begin{aligned} \mathrm{E}\left[q(X)h^2(X)\right] &= \mathrm{E}\left[z(X)\right] \\ &= \mathrm{E}\left[q_i(X_i^*) \cdot z(X_i^*)\right] \\ &= \mathrm{E}\left[q_i(X_i^*) \cdot q(X_i^*) \cdot h^2(X_i^*)\right]. \end{aligned} \tag{4.19}$$

Combining (4.17) and (4.19) gives,

$$\mathrm{V}\left[Y^*\right] = \mathrm{E}\left[q_i(X_i^*) \cdot q(X_i^*) \cdot h^2(X_i^*)\right] - \theta^2. \tag{4.20}$$

This relation between the random variables can be used to adjust biasing adaptively. As only the first term in equations (4.17) and (4.20) depends on the chosen importance function $q(x)$ minimising the variance with respect to $q(x)$ is equivalent to minimising the first term.

Assume that the original random variable X belongs to some family of distributions parametrised by β and that we are trying to find the optimal pdf from the same family, i.e. we are looking for the optimal value for β. Furthermore, we assume that there are already m biased samples, each from a possibly different biased distribution $p(\beta_i, x)$, parametrised by $\beta_1, \ldots, \beta_m$, and want to find the optimal importance function $p(\beta, x)$ parametrised by β. From (4.20) it follows that minimising the variance of Y^* is equivalent to minimising,

$$\hat{r}(\beta) = \sum_i q(\beta, X_i^*) \cdot q(\beta_i, X_i^*) \cdot h^2(X_i^*).$$

If all the samples are from the original distribution the function to be minimised with respect to β reduces to,

$$\hat{r}(\beta) = \sum_i q(\beta, X_i) \cdot h^2(X_i).$$

The above equations are very similar to those proposed by Rubinstein in [Rub97]. He proposed a two-stage procedure where in the first stage one estimates the optimal sampling distribution and in the second stage obtains the final solution. Our approach in Publications 5 and 6, however, is slightly different. Namely, we are dealing with a dynamic system where decisions are made based on replications, and the biasing parameters are based on the samples from the previous decision epoch. By doing this we hope that the dynamically changing sampling distribution follows the changes in the system state appropriately.

Biasing poissonian arrivals

Recalling that when applying FPI to the RWA problem one is interested in estimating the blocking probability during a short time interval $(0, T)$, a natural idea is to increase the arrival rates (assuming blocking events are rare, as they usually are).

For Poisson arrivals the posterior distribution of arrival instants given the number of arrivals during a time interval, is uniform. Thus, the likelihood ratio is the same for any sample path with the same number of arrivals. Let λ be the original arrival intensity, i.e. distribution parameter, and $\alpha\lambda$ the biased arrival intensity where the parameter α is to be determined. Then, the likelihood ratio for a sample path x becomes

$$q(x) = \tilde{q}(n(x)) = e^{(\alpha-1)\lambda T} \cdot \alpha^{-n(x)}.$$

where $n(x)$ is the total number of arrivals. Assuming biased sample paths X_i^*, we need to find α which minimises,

$$\begin{aligned} \hat{r}(\alpha) &= e^{(\alpha-1)\cdot\lambda T} \sum_i q(\alpha_i, X_i^*) h^2(X_i^*) \cdot \frac{1}{\alpha^{n(X_i^*)}}, \\ &= e^{(\alpha-1)\cdot\lambda T} \sum_i Y_i^* \cdot \frac{h(X_i^*)}{\alpha^{n(X_i^*)}}. \end{aligned} \tag{4.21}$$

The factors $q(\alpha_i, X_i^*)$ are not expanded because the terms $Y_i^* = q(\alpha_i, X_i^*) \cdot h(X_i)$ are determined anyway when an estimate for θ is obtained. The optimal α minimising (4.21) can be easily found by using, e.g. the Newton-Raphson method.

Other approaches

It can be assumed that the correlation between the number of blocked customers and the number of arrivals is quite strong, e.g. something like

$$h(x) \approx b \cdot n(x),$$

where b is the blocking probability during $(0, T)$ starting from the current state. In the optimal biasing, the probability of outcome x should be proportional to the product $h(x)p(x)$. Hence, taking the samples could be first conditioned on the number of arrivals n. Let $\tilde{p}^*(n)$ be the probability that there are n arrivals. Then one could use,

$$\tilde{p}^*(n) = b \cdot \frac{(\lambda T)^n}{(n-1)!} \cdot e^{-\lambda T},$$

to draw the number of arrivals. In this thesis, however, we do not consider this kind of approach and leave it for future study.

Adaptive biasing of arrivals

As was suggested in Section 4.7 biasing the arrival rates in the simulation can be used to reduce the variance of the estimator. In particular, sample paths with no blocking events should be avoided. However, too high (or low) an arrival rate can lead to very high values for the likelihood ratio $q(x)$ and thus deteriorate the overall performance of the algorithm. The adaptive biasing technique, presented first in Publication 5, can be used to overcome this.

We choose to increase all arrival rates with a common multiplier α (see Section 4.7). At each decision epoch, we can use a constant biasing parameter α_t, initially set to 1.0. Assuming the system dynamics change only little between decisions epochs, the previous N samples used to make the previous decision can be used to estimate the next optimal α_{t+1}. Assume that the system is in state j and a class-k request arrives. Our algorithm becomes,

1. Run N samples using the previously computed biasing parameter α_t.
2. Choose the action using (4.13) where costs resulting from the biased sample paths must be multiplied with appropriate likelihood ratios.
3. Update the value of the biasing parameter α by minimising (4.21) based on the latest N samples for the chosen action a.

Example WDM network

In Publication 6, an example case is presented. Fig. 4.4 illustrates the hypothetical WDM-network which is assumed to have 8 wavelength channels available on each link. The offered traffic is uniform between all node-pairs with load $a = 0.4$. The used FPI parameters are:

- the length of simulation in each replica, T, equals one holding time and the number of replications, N, is 50,
- the confidence parameter in (4.13) is $k = 1.0$.

Note that, as explained above, the simulations are actually made on two levels. The upper level represents the real system where the actual lightpath requests arrive. Then for each request we are supposed to decide on some action, i.e. what RW-pair to choose or whether to reject the request. In order to make that decision, a set of simulations are run starting from each network state resulting from different initial actions. These lower level simulations will be run similarly in practice, i.e. the network management unit would run the simulations in order to determine the optimal action to

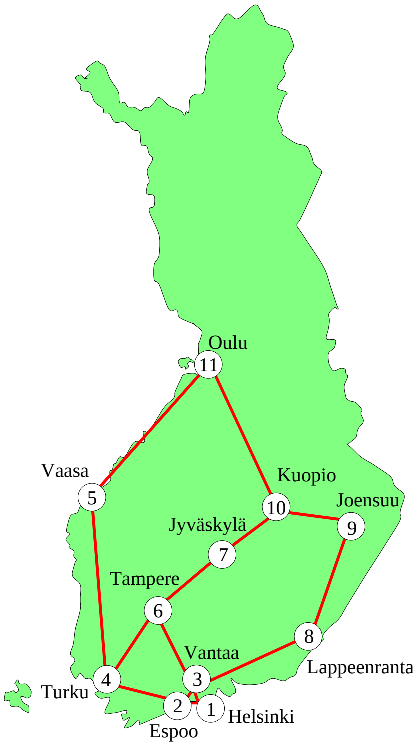

Figure 4.4: Hypothetical WDM-network in Finland.

the current request. The AIS is used in these inner replications to improve the quality of the cost estimates needed in FPI.

As the FPI algorithm can be used together with any (quick) heuristic algorithm, we do not compare the differences in performance between the different standard policies, but rather choose to use a simple standard policy *FF-RW* (see Section 4.3 for details). The set of routes was chosen to consist of all the shortest paths (one or more) in terms of number of hops.

The numerical results can be seen in Table 4.1. FPI reduces the blocking probability considerably, while the improvement using the adaptive importance sampling is less, from 4.90% to 4.79%. However, as using AIS does not increase computational effort, the improvement is "free" and

policy	blocking probability
standard	5.94%
FPI	4.90%
FPI with AIS	4.79%

Table 4.1: Simulation results with the hypothetical WDM-network of Fig. 4.4 residing in Finland. Using the AIS improves the quality of the cost estimates and leads to a better iteration policy. (Publication 6)

should not be left unused. Also, the number of samples $N = 50$ is quite small and the algorithm adjusting α could benefit from more samples. For example the next value for α could be based on the last two decisions epochs instead of considering only the last one.

4.8 Summaries

The topic of this chapter is the dynamic RWA problem in a WR optical network. For any moderate size or larger network the state space is huge and the exact optimal policy cannot be determined. In practice, the solution is to use heuristic algorithms and several reasonably good heuristic algorithms can be found in the literature (see, e.g. [KA96, RS95, SB97, MA98, KA98]).

In Publications 3, 4, 5 and 6 a systematic approach is developed to improve any given heuristics by applying the first policy iteration step of the MDP theory with on-the-fly simulation of the relative costs of states.

Summary of Publication 3
The weak point of most (if not all) heuristic algorithms is their inflexibility in adapting to new situations. In Publication 3, we propose a novel approach to improve any given heuristic RWA algorithm by applying so-called first policy iteration (FPI) step. In FPI, one has a set of possible actions to choose from and proceeds by estimating the expected cost due to each action. The cost estimates are obtained by numerical on-the-fly simulations of the system after each possible action where the consecutive decisions are made using some simple heuristic rule, i.e. so-called standard policy. As an example case we consider a mesh network with a homogeneous traffic pattern.

Summary of Publication 4
In Publication 4, we demonstrate the effectiveness of the FPI approach in the case of inhomogeneous traffic. By means of numerical examples it is shown that the policy resulting from FPI is superior to the considered standard policies. Hence, policy resulting from the FPI step is capable of choosing such routes which are unlikely to be congested with the current traffic matrix.

Summary of Publication 5
Importance sampling (IS) is a well-known technique for variance reduction. The key problem with IS is in determining a good bias. In Publication 5, we propose an algorithm for adaptively adjusting biasing parameters based on the past samples. The algorithm is shown to work well by means of numerical examples. As an example we consider biasing poissonian arrival process in the standard Erlang loss system. The objective is to obtain a better estimate for the mean number of blocked customers during a finite time interval $(0, T)$ when there is initially a certain number of customers in service.

Summary of Publication 6

The drawback with the FPI approach is the considerably longer running time when compared to the standard policy. This problem can be alleviated to some degree by using the IS method, in which one tries to favour events having the largest contribution to the estimated quantity. In Publication 6, we combine the IS technique of Publication 5 with the FPI approach. In particular, the relative values needed in FPI are evaluated using process simulation with biased arrival rates.

Author's Contribution to Publications 3, 4, 5 and 6

In Publications 3 and 4, we present a method to improve any given dynamic RWA algorithm by applying the first policy iteration. The idea of the first policy iteration with on-the-fly simulations of the relative values of states was developed in collaboration with the supervisor. The paper is jointly written by the present author and Virtamo. Publication 5 presents a novel approach where twisting parameters of importance sampling technique can be iteratively improved based on the previous samples. The presented updating formula for biasing parameters was first discovered by Virtamo and then reformulated and simplified by the present author. Publication 6 contains a summary of the previous work and presents numerical examples, where the adaptive IS technique is applied to dynamic RWA problem. The numerical examples as well as the simulator code are the work of the present author.

5 OPTICAL BURST SWITCHING

5.1 Introduction

In the previous chapters we have been dealing with wavelength- routed networks (WRNs), where optical lightpaths are established between certain node pairs. Each lightpath serves as a virtual link with a constant capacity between its endpoints. At the end of each lightpath, the optical signal is converted to the electronical domain and processed accordingly and transmitted further if necessary.

In this chapter, we consider optical burst switching (OBS) which is seen as an intermediate step on the way from the wavelength-routed networks towards the optical packet switched networks (OPSNs). The OBS scheme was originally proposed by Yoo and Qiao in [YQ97, QY99]. Since then, OBS has been a topic of active research, see e.g. [Tur99, DG01, XPR01, DGSB01, BP03]. Generally, it can be said that optical burst switching networks (OBSN) overcome the quasi-static nature of WRNs while taking into account the limitations imposed by the current technology.

In an OBS network, the data are transmitted in bursts consisting of several packets going to the same destination. An often made assumption is that the packets are IP packets, while the OBS scheme itself does not depend on this. At the edge nodes, the incoming packets are gathered into bursts and then transmitted together over the OBS network (see Fig. 5.1). The process where an edge node forms a burst is often referred to as the burstification or the burst aggregation process and can have a great impact on the overall performance.

One specific characteristic of OBS is the strong separation between the control and the data plane. The control plane, carrying the control packets, is typically realised by a low capacity out-of-band channel, e.g. a dedicated wavelength channel. At each intermediate node, the control packets are converted into the electronical domain and then processed electronically. The data plane, on the other hand, consists of one or more all-optical wavelength channels, i.e. the bursts travel throughout the network in the optical domain.

One weak point of many proposed OBS schemes is the fairness. In this thesis, we study possibilities to promote fairness in different kinds of OBS networks. Publication 9 deals with OBS in ring networks and Publications 7 and 8 deal with OBS in mesh networks.

5.2 OBS in Mesh Networks

Edge Nodes and Burst Aggregation

The OBSN consists of two kinds of nodes, edge nodes and core nodes (see Fig. 5.1). Edge nodes are devices which gather the incoming packets and sort them based on the destination address to different output queues. The edge node then forms a burst and transmits it to the core network. Analytical models for edge nodes have been developed in [XPR03b, XPR03a].

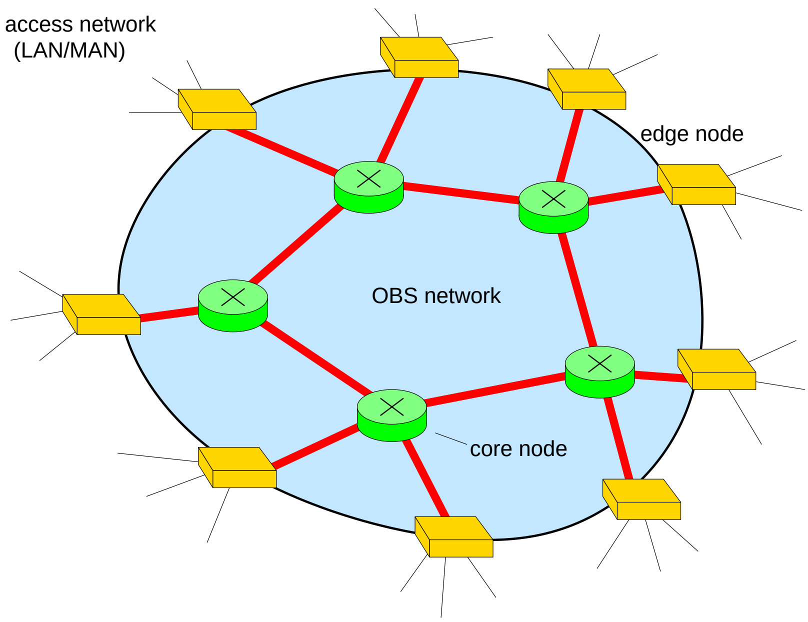

Figure 5.1: Optical burst switching (OBS) network. Edge nodes aggregate several packets going to the same destination and then send them through the OBS network in one burst.

The burst assembly process in each edge node is controlled by a timer which expires after a certain time interval [Dix03]. If the expiry interval is a constant the assembly process is deterministic and is referred to as the *constant assembly time* (CAT). Alternatively, the assembly time interval can be uniformly distributed with some parameters, which is referred to as the *variable assembly time* (VAT). Whenever the timer expires, the burst assembler checks whether any packets reside in the assembly buffer or not. If there are packets, the burst assembler moves the corresponding packets out of the assembly buffer and forms a burst.

The burstification process can have a great impact on the overall performance. The two key performance measures of the burst assembly process are the variance of the burst interarrival time and the mean burstification delay, and, in general, the aim is to minimise both of them at the same time [Dix03]. A smaller variance means that the offered traffic to the core OBS network is not as bursty as it would otherwise be. On the other hand, a small burstification delay is also desirable as it allows smaller buffer sizes and a higher throughput with TCP [DL02, CLCQ02, GSSC03].

Core Node Architectures

The core nodes have two main functions. Firstly, they schedule the resources, i.e. the use of outgoing links and possible FDLs. Secondly, they control the switching fabric accordingly.

Several slightly different protocol versions have been proposed for the resource scheduling. Firstly, the setup can be either explicit, i.e. the outgoing link is reserved from the current instant of time onwards, or estimated, where the outgoing link is reserved from the estimated time of burst arrival onwards. The latter option leads to a more efficient use of resources but is also more complex to implement, especially in hardware. Similarly for the release, two options exist, explicit release and estimated release. The functions the core nodes perform are strongly related to the used reservation scheme, which we will describe in Section 5.4.

5.3 Contention Resolution

The contention resolution plays an important role also in OBS networks. For example, assume that each burst consists of several (IP) packets. Then, each dropped burst corresponds to a loss of several consecutive TCP packets which can detoriate the performance of the TCP protocol [CLCQ02, DL02, GSSC03].

Thus, in order to avoid severe performance degradation, the blocking probability in the network must be kept low. This, however, means that the sustainable level of offered load, and hence also the throughput, must be considerably lower than what the network could handle with a more sophisticated reservation scheme. Especially important is the problem of contention resolution in OBS networks without wavelength conversion as the lack of wavelength conversion induces a high blocking rate on paths which consist of several links. In this section we will give a brief introduction to different contention resolution schemes proposed for OBS networks.

Fibre Delay Lines

One major deficiency with OBSN (and OPSN) is the lack of optical memory, which leaves us with two options. Either accept OEO-conversion and buffer the packets/bursts in the electronical domain, or use fibre delay lines (FDL) as a temporary memory. The latter option seems more appealing as it does not require OEO-conversion and is thus blind, e.g. to the used modulation and bit rate. FDL delays the actual burst and the OBS protocol must be revised accordingly. Generally, two alternative FDL scheduling mechanisms exist: PreRes and PostRes [Gau02, BP03]. In practice, the PreRes scheme has better performance and is typically chosen in the studies.

In the *PreRes scheme*, the node checks if suitable FDL lines exist for which also the respective output link will have a free channel at the time the burst exits the FDL and then chooses the shortest of them. The control packet with updated information is passed further as soon as it has been processed. Thus, in this scheme the offset time, which also corresponds to the priority, gets increased each time the burst is delayed.

In the *PostRes scheme*, the control packet is also delayed by the same amount of time as the burst. After the delay, the control packet is reprocessed and only at this point is a free channel on the output link checked. Thus, it is possible that a burst is guided to an FDL unnecessarily. On the other hand, in this scheme the offset times remain the same and possible QoS classes based on extended offsets are not affected.

Deflection Routing

The simplest approach to dealing with contention and avoiding burst losses is to deflect the otherwise dropped burst to some other direction with a free port. This scheme, proposed by Wang et al. in [WMA00b, WMA00a], is known as deflection routing protocol for OBS networks. The idea of using deflection to resolve contention, however, has been proposed earlier in the context of ATM, but also for optical networks. Note that the deflection routing and FDL are quite similar methods to resolve contention as in both cases a burst to be dropped is "delayed" by guiding it into a free fibre instead. Thus, loosely speaking, the idea with the deflection routing is to use the idle time of fibre links as optical memory. Also analytical models have been developed for evaluating the performance of deflection routing, see, e.g., [CWXQ03].

Generally, when the network load is reasonably low the use of deflection routing can improve the throughput considerably, but as the load increases above a certain threshold, the situation however, becomes the opposite and the throughput suddenly collapses [WMA00b, WMA00a]. Thus, with deflection routing additional care must be taken in order to keep the load under a certain limit and the network operational.

Burst Segmentation

An alternative method to deal with contention resolution is so-called burst segmentation proposed by Vokkarane et al. in [VJS02]. In burst segmentation, each burst can be divided into segments. When contention occurs, instead of dropping an entire burst, the node drops only the overlapping segments. By doing this, the non-overlapping parts of the bursts can be transmitted succesfully and a higher overall throughput is achieved. Typically, the burst segmentation scheme is combined with the deflection routing, which together have been shown to achieve a significantly reduced packet loss rate [VJS02]. The burst segmentation scheme can be further developed to take into account QoS aspects by burst priorisation [VJ03].

5.4 Reservation Schemes

In OBS networks temporary reservations are made for each fibre section a burst traverses by an out-of-band control packet, which is sent before the burst in a dedicated control channel. Several different reservation schemes have been proposed for OBS networks (see Fig. 5.2), which can be classified into centralised or distributed signalling schemes. In a centralised signalling scheme each reservation is processed by a centralised scheduler, which is responsible for resource scheduling for the whole OBS network. In distributed signalling schemes the resource scheduling for each fibre section is handled by the network core nodes in a distributed fashion. The most commonly proposed protocols employ *one-way reservation* schemes.

One-Way Reservation Schemes

The one-way reservation scheme is a tell-and-go protocol and is illustrated in Fig. 5.3. Prior to sending the actual burst the source node sends a control packet on a dedicated control channel along the same path as the burst will

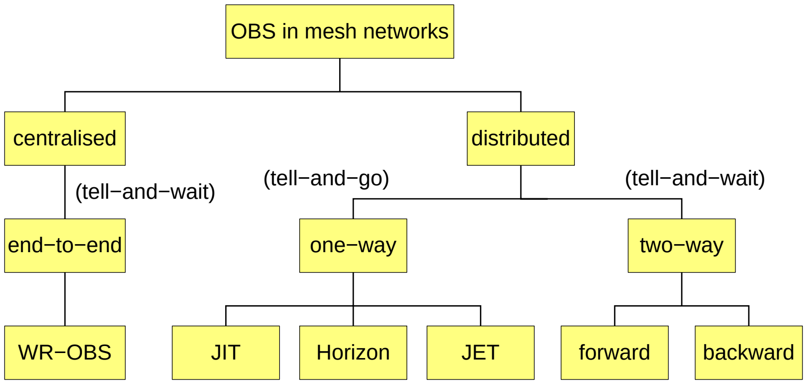

Figure 5.2: Classification of different OBS schemes for mesh networks.

traverse. The control packet reserves the resources and acknowledges the corresponding nodes about the coming burst so that the intermediate nodes have time to configure their switching fabrics before the actual burst arrives.

Meanwhile, after a certain offset time, the source node begins to send the actual burst without waiting for a positive acknowledgement message. The minimum possible offset time depends on the number of intermediate nodes along route as illustrated in Fig. 5.3. The offset time must be long enough to ensure that the control packet reaches the destination node in time before the actual bursts arrives. The basic idea behind the one-way reservation scheme is to share the available network resources efficiently without introducing any complex capacity reservation schemes. In order to do this, one accepts the possibility of bursts being blocked occasionally. The main advantage of the one-way reservation scheme, when compared to other schemes, is the reduced latency due to the fact that the burst does not have to wait for a positive acknowledgement.

Optionally, in cases when there is congestion on some link, a NAK packet is sent back to the source node along the same route. In each intermediate node, the NAK packet cancels the reservation and the burst is dropped at the point it meets the NAK packet.

JIT protocol

Just-In-Time protocol (JIT), proposed in [WPRT99], uses estimated setup and explicit release. Thus, the control packet contains information about the arrival time of the burst, but not about the burst length, and the reservation remains active until it is released by an in-band terminator at the end of each burst. Thus, each core node needs to maintain information only about which wavelength channels are reserved on each outgoing link [WPRT99, DGSB01].

Horizon

Horizon reservation scheme proposed by Turner in [Tur99], assumes a full wavelength conversion. In the Horizon concept, each core node maintains information about the time when each wavelength channel will be

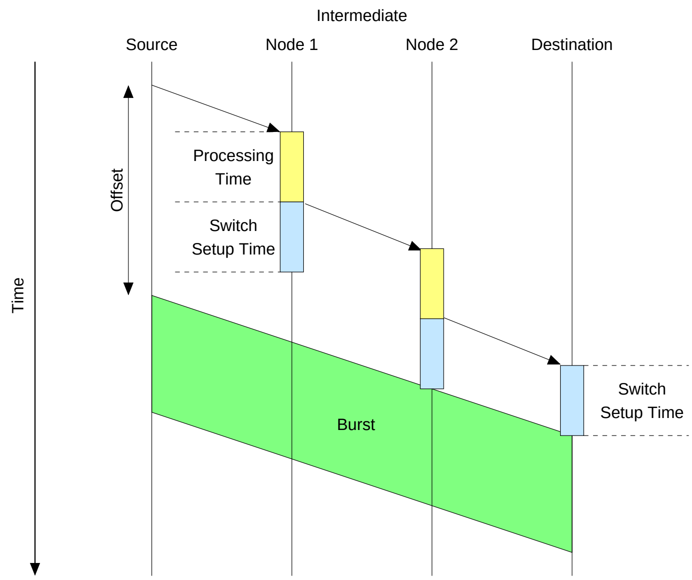

Figure 5.3: The one-way reservation scheme, e.g. JET (adopted from [BP03]).

free (horizon). Upon the arrival of a new burst, the scheduler allocates the burst to such a wavelength channel that has the largest horizon time less than the arrival time of the burst, i.e. the scheduling algorithm tries to pack the outgoing bursts so that the void time between consecutive bursts is minimised [DGSB01].

JET protocol

Just-Enough-Time protocol (JET), proposed by Qiao and Yoo in [QY99], is probably the most common one-way reservation scheme. In contrast to previous protocols, the JET protocol utilises estimated setup and estimated release resource reservation, which intuitively sounds efficient. Hence, in comparison to Horizon protocol, the JET protocol is capable of using the idle time between two already made reservations.

The initial offset time of burst must be at least equal to the sum of control packet processing times in intermediate nodes and the switch setup time at the destination node. As reservations are made in the same order as they arrive, the reservations made earlier have a higher probability of success. Thus, by using different additional offset times for different bursts, one can provide multiple classes of service (CoS) within the JET protocol, as proposed in [YQ99]. Note that the same approach could be applied with other one-way reservation schemes as well.

Two-Way Reservation Schemes

The *two-way reservation* schemes (i.e. a parallel reservation) by Ogushi et al. [OAMiK01] are distributed schemes where the control packet first travels from the source node to the destination node and "scans" the available wavelength channels on the route. Then, the destination node chooses one of the free wavelength channels, if any, and completes the reservation process by sending a control packet with the reservation back to the source along the same route. The two-way reservation schemes can be further classified into forward and backward reservation schemes. The difference is that the *forward reservation* scheme first reserves all free wavelength channels and then on its way back the control packet releases all except one of them. In the *backward reservation* scheme, on the other hand, the control packet first collects the information about the free channels on its way to the destination node, and then at the second step, it tries to reserve one of the free channels on its way back to the source node. In [AMMM99, OAMiK01], it is reported that the backward reservation scheme seems to be a better alternative.

Centralised Reservation Schemes

Other possible schemes include centralised signalling with end-to-end reservation, e.g. WR-OBS protocol proposed by [DB02]. The end-to-end reservation scheme uses a centralised scheduler which finds and allocates the necessary resources for the burst. The scheduler acknowledges the respective nodes and calculates the needed offset time. The WR-OBS protocol does not assume a wavelength conversion.

Burst Switching in Electronic Networks

It is worth noting that solutions similar to OBS have been also proposed earlier for both electronic and optical networks (see [HR96]). Especially, within ATM framework the so-called fast reservation protocol (FRP), i.e., the ATM block transfer mode in ITU terminology [ITU04], is very similar to OBS schemes in many respects. Firstly, in ATM block transfer mode the allocated capacity can be adjusted per block basis to match the instantaneous bandwidth requirements and thus it suits well for transmitting bursty traffic. Secondly, the (capacity) reservation process is similar to OBS. However, the separation between the data and the control plane in ATM block transfer mode is weaker than in the case of OBS.

In order to cope with different kind of situations two versions of FRP have been proposed, i.e., FRP with delayed transmissions (FRP/DT) and FRP with immediate transmissions (FRP/IT). In both versions, the source can initiate a request to increase the amount bandwidth to be allocated to the connection temporarily (peak cell rate). The request is then processed at each intermediate node, which checks if there is enough free capacity on the next link, and if so, allocate the requested capacity and send the request further. If some link has insufficient capacity, the request is dropped and a time-out mechanism takes care of clearing the capacity allocations already made on previous links. If the reservation process is succesful, the destination node sends a positive acknowledgement back to the source.

In the FRP/DT protocol, the source node has to wait for a positive acknowledgement before it is allowed to start the transmission with a new bandwidth. Hence, FRP/DT protocol is very similar to two-way reservation scheme for OBS networks proposed by Ogushi et al.

On the other hand, the other version of the FRP protocol, FRP/IT, tries to eliminate the need to buffer the increased traffic volume at the source waiting for the positive acknowledgement for the increased bandwidth request. In particular, in the FRP/IT protocol the source can start to use the requested capacity immediately after sending the request for the bandwidth increase. If there is not enough capacity available at some link along the path, the corresponding blocks may be discarded. Thus, FRP/IT is similar to one-way reservation schemes for OBS networks with the distinction that in FRP/IT there is no offset time between the request and the "burst".

5.5 Fairness

The OBS paradigm may lead to unfairness among the different connections, e.g. typically the long connections suffer from a higher loss probability than the short ones. In [QY99], the authors proposed the use of an additional offset time before sending the actual burst for higher priority traffic flows. When the offset difference is large enough (when compared to maximum burst length) the traffic classes become fully isolated, while with a smaller difference in offset the system has "soft priorities". Note that the same mechanism can be also used to balance the blocking probability among the different traffic classes. Similarly in [OAMiK01], Ogushi et al. have proposed a parallel wavelength reservation protocol, where only the longest connections are allowed to use the entire set of wavelength channels.

The author's contributions in this thesis deal mainly with the fairness issue. In Publication 7, we show that fibre delay lines (FDL) have a positive effect on the fairness in an OBS network. This is mainly due to the fact that the FDLs "shuffle" the priorities and actually, on average, raise the priorities of long connections along the route. In Publication 8, we propose an alternative method to ensure fairness by choosing the routes and wavelengths in a special way. The routing problem formulation enforces the resulting routes to be chosen so that no burst can get blocked after m optical hops. In this way we have managed to improve two factors: longer bursts do not suffer from a much higher blocking probability and also the possible NAK packet in case of burst loss reaches the source faster on average, allowing a faster retransmission cycle (if implemented).

5.6 OBS in Ring Networks

Typically, OBS is associated with mesh networks, but the use of OBS has been proposed for ring topology as well, see e.g. [XPR02, XPR03c] and Publication 9. The architecture proposed in [XPR02, XPR03c] is cost effective and suitable for metropolitan area networks (MANs). It is assumed that links are unidirectional and that each network node has a fixed transmitter on a dedicated wavelength channel and a tunable receiver. Thus, together with a control channel there are $W = N + 1$ wavelength channels, where N is the number of network nodes. In the control channel, K control packets circulate. They are processed electronically at each node. Prior to sending a burst, the node writes the respective information into a control packet (or frame). Thus, all intermediate nodes will be aware of the transmission as well, while the nodes residing before the source node and after the destination node in the ring will only know about the transmission afterwards. Note that operation of the network is slotted due to a fixed number of control frames circulating in the control channel.

Since the channels are dedicated, no bursts clash in the fibre. However, as there is only one receiver in each node, at most one burst can be received successfully at a time. Thus, when several concurrent bursts arrive at a node, it chooses (typically randomly) one of them and the others are blocked.

The proposed scheme in [XPR02, XPR03c] employs so-called only destination delay (ODD) protocol, where at each node the incoming bursts are delayed using an appropriate fibre delay line by an amount which is equal to the processing delay of the control packet. Thus, the offset time between the control packet and the burst is kept constant as they pass each node along the route. This is reported to achieve a shorter mean packet delay than using, e.g. JET [XPR02].

Several MAC protocols for this architecture have been proposed and studied in [XPR02, XPR03c]. Some of them use tokens to coordinate the transmissions between the nodes in order to avoid congestion in the receivers, while other protocols are based on the partial information they have collected from the past control packets. The proposed OBS protocols are as follows:

- **Round-Robin with Random Selection (RR/R)**: The outgoing packets are stored in queues based on the destination node. The queues are served in round-robin fashion. At the receiving side, a random burst is chosen in case there are concurrent transmissions (random selection).

- **Round-Robin with Persistant Service (RR/P)**: This protocol is similar to RR/R with the distinction that the nodes maintain a so-called "earliest-free-time" structure for each destination node based on the control frames. Before transmitting the chosen burst, the node, based on the partial information it has, waits until the receiver at the destination node is free.

- **Round-Robin with Non-Persistant Service (RR/NP)**: This protocol is a variant of the RR/P protocol, where instead of waiting for the destination receiver to become free, the sending node moves to the next queue in case the given receiver is known to be reserved (based on the same partial information as in the case of RR/P).

- **Round-Robin with Tokens (RR/T)**: This protocol is the only contention free protocol among these. In the RR/T protocol, n tokens circulate in the network and the nodes are allowed to send bursts to a given node only if they are holding the corresponding token.

The above access protocols have been evaluated by means of numerical simulations in [XPR02, XPR03c]. In Publication 9, we develop analytical models for estimating the burst blocking probability of the RR/R protocol, as well as, of the random-order-with-random-selection protocol, where each node serves the burst aggregation queues in random order instead of round-robin. It turns out that the round-robin order achieves a lower throughput than the random order protocol under the assumed high traffic load scenario.

Also other related studies have been published, see, e.g., [FK03, Whi02, WRSK03]. For example, in the Hornet project, a test bed of bidirectional ring network has been developed using OPS [Whi02, WRSK03]. Their protocol is tailored for transporting IP traffic. Fairness is guaranteed by applying *Distributed Queue Bidirectional Ring* , which essentially transforms the bidirectional ring into a distributed FCFS queue [BG92].

5.7 Summaries

Summary of Publication 7
In Publication 7 (and [NH02]), we study different delay line configurations and their effects on blocking probability and fairness in OBS networks using a revised JET protocol with no wavelength conversion. An OBS network without FDLs cannot sustain very high traffic loads as each link is essentially a blocking system without any waiting place with either W servers (with wavelength conversion) or one server (without wavelength conversion). Thus, by adding a few FDLs into each node, the blocking probability can be remarkably reduced and the overall performance of the network improved.

The main problem with protocols such as JET is that long connections have a high priority at the beginning of their journey but lose the edge as they progress further, while in fact the opposite behaviour would be desirable. However, when there is congestion and a burst gets delayed by an FDL, the time between its header and the actual burst increases, which corresponds to a higher priority. This phenomenon tends to compensate for the negative effect which is due to the offset time normally becoming shorter as the burst gets closer to its destination.

Summary of Publication 8
In Publication 8, we approach the fairness issue by altering the routing decisions from the standard shortest paths routes. The proposed OBS-aware routing formulation guarantees that no burst gets blocked after m hops, where m is an adjustable parameter.

The routing problem is formulated as an MILP problem where the objective function is to minimise the maximum channel load. Together with the m-hop clash constraint, it is shown to lead to considerably better overall configurations than the standard approach.

Summary of Publication 9
In Publication 9, we develop an analytical loss model for some MAC protocols proposed in [XPR02, XPR03c, Bat02] for OBS optical ring networks. The models have been verified and shown to be accurate by means of numerical simulations. The models consider both random and round robin order in burst transmission, of which the random order schedule achieves a higher throughput especially when the network load is high.

Author's Contribution to Publications 7, 8 and 9

The present author's contribution in Publication 7 has been in the analysis of the numerical results obtained by simulations together with the co-author. In Publication 8, the routing formulation and the analytical results are mainly a work by the present author, while the numerical results with mesh topology are a work by the co-author. Similarly, in Publication 9, the analytical models are derived by the present author and the numerical examples are a result of a joint effort.

REFERENCES

[AH77] Kenneth Appel and Wolfgang Haken. The solution of the four-color-map problem. *Scientific American*, 237(4):108–121, October 1977.

[Ahl00] Ulf Ahlfors. *Dynamic Traffic Control in Multiservice Networks*. PhD thesis, Lund Institute of Technology, Lund University, 2000.

[AMMM99] Shin'ichi Arakawa, Kentaro Miyamoto, Masayuki Murata, and Hideo Miyahara. Performance analyses of wavelength reservation methods for high-speed data transfer in photonic networks. In *ITC-CSCC*, pages 828–831, July 1999.

[Bar98] Stefano Baroni. *Routing and wavelength allocation in WDM optical networks*. PhD thesis, University College London, May 1998.

[Bat02] Tzvetelina (Lina) Battestilli. Optical burst switching: A survey. Technical Report TR-2002-10, North Carolina State University, July 2002.

[BB97] Stefano Baroni and Polina Bayvel. Wavelength requirements in arbitrarily connected wavelength-routed optical networks. *IEEE J. Lightwave Technol.*, 15(2):242–251, February 1997.

[BFG90] J.A. Bannister, L. Fratta, and M. Gerla. Topological design of the wavelength-division optical network. In *Infocom '90*, volume 3, pages 1005–1013. IEEE, June 1990.

[BG92] Dimitri Bertsekas and Robert Gallager. *Data Networks*. Prentice-Hall, 2nd edition, 1992.

[BGD01] David J. Bishop, C. Randy Giles, and Saswato R. Das. The rise of optical switching. *Scientific American*, 284(1):74–79, January 2001.

[BH96] R. A. Barry and P. A. Humblet. Models for blocking probability in all-optical networks with and without wavelength exchangers. *IEEE J. Select. Areas Commun.*, 14(5):858–867, June 1996.

[Bir96] Alexander Birman. Computing approximate blocking probabilities for a class of all-optical networks. *IEEE J. Select. Areas Commun.*, 14(5):852–857, June 1996.

[BM76] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. The MacMillan Press, 1976.

[BM96] Dhritiman Banaree and Biswanath Mukherjee. A practical approach for routing and wavelength assignment in large wavelength-routed optical networks. *IEEE J. Select. Areas Commun.*, 14(5):903–908, June 1996.

[BM00a] Dhritiman Banaree and Biswanath Mukherjee. Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study. *IEEE/ACM Trans. Networking*, 8(5):598–607, October 2000.

[BM00b] Dhritiman Banerjee and Biswanath Mukherjee. Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study. *IEEE/ACM Trans. Networking*, 8(5):598–607, October 2000.

[BP03] Tzvetelina Battestilli and Harry Perros. An introduction to optical burst switching. *IEEE Optical Communications*, pages S10–S15, August 2003.

[Bré79] Daniel Brélaz. New methods to colour the vertices of a graph. *Communications of the ACM*, 22(4), April 1979.

[BRM00] Paul Bonenfant and Antonio Rodriguez-Moral. Optical data networking. *IEEE Commun. Mag.*, 38(2):63–70, February 2000.

[BSSLB95] Krishna Bala, Thomas E. Stern, David Simchi-Levu, and Kavita Bala. Routing in a linear lightwave network. *IEEE/ACM Trans. Networking*, 3(4):459–469, June 1995.

[BYC97] Subrata Banerjee, Jay Yoo, and Chien Chen. Design of wavelength-routed optical networks for packet switched traffic. *IEEE J. Lightwave Technol.*, 15(9), September 1997.

[CLCQ02] X. Cao, J. Li, Y. Chen, and C. Qiao. TCP/IP packets assembly over optical burst switching network. In *Global Telecommunications Conference - (GlobeCom'02)*, volume 3, pages 2808–2812, 2002.

[Coo81] Robert B. Cooper. *Introduction to Queueing Theory*. Elsevier Science Publishing, 1981.

[CWXQ03] Yang Chen, Hongyi Wu, Dahai Xu, and Chunming Qiao. Performance analysis of optical burst switched node with deflection routing. In *IEEE International Conference on Communications (ICC)*, volume 2, pages 1355–1359, 2003.

[DB02] Michael Düser and Polina Bayvel. Analysis of a dynamically wavelength-routed optical burst switched network architecture. *IEEE J. Lightwave Technol.*, 20(4):574–585, April 2002.

[DG01] Klaus Dolzer and Christoph Gauger. On burst assembly in optical burst switching networks – a performance evaluation of just-enough-time. In *the 17th International Teletraffic Conference, ITC-17*, pages 149–161, September 2001.

[DGSB01] Klaus Dolzer, Christoph Gauger, Jan Späth, and Stefan Bodamer. Evaluation of reservation mechanisms for optical burst switching. *International Journal of Electronics and Communications (AEÜ)*, 55(1), 2001.

[Dix03] Sudhir Dixit, editor. *IP over WDM: Building the Next-Generation Optical Internet*. John Wiley & Sons, 2003.

[DL02] A. Detti and M. Listanti. Impact of segments aggregation on TCP Reno flows in optical burst switching networks. In *Infocom 2002*, pages 1803–1812, March 2002.

[DR00] Rudra Dutta and George N. Rouskas. A survey of virtual topology design algorithms for wavelength routed optical networks. *Optical Networks Magazine*, 1(1):73–89, January 2000.

[Dud89] Richard M. Dudley. *Real Analysis and Probability*. Wadsworth & Brooks/Cole, 1989.

[Dzi97] Zbigniew Dziong. *ATM Network Resource Management*. McGraw-Hill, 1997.

[FK03] A. Fumagalli and P. Krishnamoorthy. A low-latency and bandwidth-efficient distributed optical burst switching architecture for metro ring. In *IEEE International Conference on Communications (ICC)*, volume 2, pages 1340–1344, 2003.

[Gau02] Cristoph M. Gauger. Dimensioning of FDL buffers for optical burst switching nodes. In *the 6th IFIP Working Conference on Optical Network Design and Modeling (ONDM 2002)*, Torino, Italy, February 2002.

[GDW00] Nasir Ghani, Sudhir Dixit, and Ti-Shang Wang. On IP-over-WDM integration. *IEEE Commun. Mag.*, 38(2):72–84, February 2000.

[Gha00] Nasir Ghani. Lambda-labeling: A framework for IP-over-WDM using MPLS. *Optical Networks Magazine*, 1(2):45–58, April 2000.

[GSM97] M. Garnot, M. Sotom, and F. Masetti. Routing strategies for optical paths in WDM networks. *IEEE International Conference on Communications 1997*, pages 422–446, 1997.

[GSSC03] S. Gowda, R. Shenai, K. Sivalingam, and H. C. Cankaya. Performance evaluation of TCP over optical burst-switched (OBS) WDM networks. In *IEEE International Conference on Communications (ICC)*, volume 2, pages 1433–1437, 2003.

[HdW87] A. Hertz and D. de Werra. Using tabu search techniques for graph coloring. *Computing*, 39:345–351, 1987.

[Hee97] Poul E. Heegaard. A survey of Speedup simulation techniques. Workshop tutorial on Rare Event Simulation, Aachen, Germany, August 1997.

[Hei95] Phil Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modeling and Computer Simulation*, 5(1):43–85, January 1995.

[HR96] Gail C. Hudek and Ira Richer. Signaling protocol implementations on a burst-switched all-optical networking testbed. In *Optical Fibre Communication Conference (OFC)*, pages 315–317, 1996.

[HV98] Esa Hyytiä and Jorma Virtamo. Wavelength assignment and routing in WDM networks. In *the 14th Nordic Teletraffic Seminar, NTS 14*, pages 31–40, Lyngby, Denmark, August 1998.

[HV99] Esa Hyytiä and Jorma Virtamo. Wavelength assignment in multifibre WDM-networks. Technical Report TD(99)04, COST257, 1999.

[Hyy01] Esa Hyytiä. Dynamic control of all-optical WDM networks, February 2001. Licentiate thesis, Networking Laboratory, HUT.

[ITU98] ITU-T. *G.692, Optical interfaces for multichannel systems with optical amplifiers*, October 1998.

[ITU04] ITU-T. *ITU-T Rec. I.371, Traffic control and congestion control in B-ISDN*, March 2004.

[KA96] Milan Kovačević and Anthony Acampora. Benefits of wavelength translation in all-optical clear-channel networks. *IEEE J. Select. Areas Commun.*, 14(5):868–880, June 1996.

[KA98] Ezhan Karasan and Ender Ayanoglu. Effects of wavelength routing and selection algorithms on wavelength conversion gain in WDM optical networks. *IEEE/ACM Trans. Networking*, 6(2):186–196, April 1998.

[KPG02] Josué Kuri, Nicolas Puech, and Maurice Gagnaire. A tabu search algorithm to solve a logical topology design problem in wdm networks considering implementations costs. In *SPIE Asian Pacific Optical Conference*, Shangai (Chine), oct 2002.

[Kri90] K. R. Krishnan. Markov decision algorithms for dynamic routing. *IEEE Commun. Mag.*, 28(10):66–69, October 1990.

[KS01] Rajesh M. Krishnaswamy and Kumar N. Sivarajan. Design of logical topologies: A linear formulation for wavelength-routed optical networks with no wavelength changers. *IEEE/ACM Trans. Networking*, 9(2):186–198, April 2001.

[LA91] J.-F.P. Labourdette and A.S. Acampora. Logically rearrangeable multihop lightwave networks. *IEEE/Trans. Communications*, 39(8):1223–1230, August 1991.

[Law95] Gregory F. Lawler. *Introduction to Stochastic Processes*. Chapman & Hall Probability Series. Chapman & Hall, 1995.

[LH03] X. Lu and S. He. Wavelength assignment for WDM ring. *Electronic Letters*, 39(19):1400–1402, September 2003.

[LMM00] Emilio Leonardi, Marco Mellia, and Marco Ajmone Marsan. Algorithms for the logical topology design in WDM all-optical networks. *Optical Networks Magazine*, 1(1):35–46, January 2000.

[LS00a] Guangzhi Li and Rahul Simha. The partition coloring problem and its application to wavelength routing and assignment. In *Optical Networks Workshop*, Richardson, Texas, February 2000.

[LS00b] Gang Liu and Thomas E. Stern. An algorithm for optimal routing and wavelength assignment on a bidirectional ring. In *Optical Networks Workshop*, Richardson, Texas, February 2000.

[MA98] A. Mokhtar and M. Azizoğlu. Adaptive wavelength routing in all-optical networks. *IEEE/ACM Trans. Networking*, 6(2):197–206, April 1998.

[MBRM96] Biswanath Mukherjee, Dhritiman Banerjee, S. Ramamurthy, and Amarnath Mukherjee. Some principles for designing a wide-area WDM optical network. *IEEE/ACM Trans. Networking*, 4(5):684–696, October 1996.

[MGL+02] Marco Ajmone Marsan, Andrea Grosso, Emilio Leonardi, Marco Mellia, and Antonio Nucci. Design of logical topologies in wavelength-routed IP networks. *Photonic Network Communications*, 4(3/4):423–442, December 2002.

[Mit76] John Mitchem. On various algorithms for estimating the chromatic number of a graph. *The Computer Journal*, 19(2):182–183, 1976.

[MM99] G. Mohan and S. R. Murthy. A time optimal wavelength rerouting algorithm for dynamic traffic in WDM networks. *IEEE J. Lightwave Technol.*, 17(3):406–417, March 1999.

[MNG+01] Marco Mellia, Antonio Nucci, Andrea Grosso, Emilio Leonardi, and Marco Ajmone Marsan. Optimal design of logical topologies in wavelength-routed optical networks with multicast traffic. In *Globecom 2001*, San Antonio, Texas, November 2001.

[Muk97] Biswanath Mukherjee. *Optical Communication Networks*. McGraw-Hill series on computer communications. McGraw-Hill, 1997.

[NH02] Laura Nieminen and Esa Hyytiä. Delay line configurations in optical burst switching with JET protocol. In *Sixteenth Nordic Teletraffic Seminar, NTS-16*, pages 229–240, Espoo, Finland, August 2002.

[OAMiK01] Ikutaro Ogushi, Shin'ichi Arakawa, Masayuki Murata, and Ken ichi Kitayama. Parallel reservation protocols for achieving fairness in optical burst switching. In *2001 IEEE Workshop on High Performance Switching and Routing (HPSR)*, pages 213–217, May 2001.

[PKG02a] Nicolas Puech, Josué Kuri, and Maurice Gagnaire. Models for the logical topology design problem. In *Networking 2002*, pages 301–313, Pisa, Italy, May 2002.

[PKG02b] Nicolas Puech, Josué Kuri, and Maurice Gagnaire. Topological design and lightpath routing in WDM mesh networks: A combined approach. *Photonic Network Communications*, 4(3/4):443–456, 2002.

[PMMB00] A. Pattavina, M. Martinelli, G. Maier, and P. Boffi. Techniques and technologies towards all-optical switching. *Optical Networks Magazine*, 1(2):75–93, April 2000.

[QY99] C. Qiao and M. Yoo. Optical burst switching (OBS) - a new paradigm for an optical internet. *Journal of High Speed Networks (JHSN)*, 8(1):69–84, 1999.

[Ree95] Colin R. Reeves, editor. *Modern Heuristic Techniques for Combinatorial Problems*. Advanced Topics in Computer Science Series. McGraw-Hill, 1995.

[Ros70] Sheldon M. Ross. *Applied Probability Models with Optimization Applications*. Holden-Day Inc., 1970.

[Ros00] Sheldon M. Ross. *Introduction to Probability Models*. Academic Press, 7th edition, 2000.

[RRP99] Sridar Ramesh, George N. Rouskas, and Harry G. Perros. Computing blocking probabilities in multi-class wavelength routing networks. Technical Report TR-99-08, Department of Computer Science, College of Engineering, North Carolina State University, May 1999. http://www.csc.ncsu.edu/research/tech-reports/README.html.

[RS95] Rajiv Ramaswami and Kumar N. Sivarajan. Routing and wavelength assignment in all-optical networks. *IEEE/ACM Trans. Networking*, 3(5):489–500, October 1995.

[RS96] Rajiv Ramaswami and Kumar N. Sivarajan. Design of logical topologies for wavelength-routed optical networks. *IEEE J. Select. Areas Commun.*, 14(5):840–851, June 1996.

[RS98] Rajiv Ramaswami and Kumar N. Sivarajan. *Optical Networks, A Practical Perspective.* Morgan Kaufmann Series in Networking. Morgan Kaufmann Publishers, 1998.

[RSORS96] V.J Rayward-Smith, I.H. Osman, C.R. Reeves, and G.D. Smith. *Modern Heuristic Search Methods.* John Wiley & Sons, 1996.

[Rub97] Reuven Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, May 1997.

[Rum00] Hannu Rummukainen. On approximate markov control of multiservice telecommunication links. Master's thesis, Helsinki University of Technology, 2000.

[Saa61] Thomas L. Saaty. *Elements of Queueing Theory With Applications.* McGraw-Hill, 1961. Dover edition, 1983.

[SB97] Suresh Subramaniam and Richard A. Barry. Wavelength assignment in fixed routing WDM networks. *IEEE International Conference on Communications 1997*, pages 406–410, 1997.

[SB99] Thomas E. Stern and Krishna Bala. *Multiwavelength Optical Networks: a Layered Approach.* Addison Wesley, 1999.

[SK77] Thomas L. Saaty and Paul C. Kainen. *The Four-Color Problem, Assaults and Conquest.* McGraw-Hill, 1977. Dover edition, 1986.

[Tij94] Henk C. Tijms. *Stochastic Models, An Algorithmic Approach.* John Wiley & Sons, 1994.

[TJWW03] Maoning Tang, Weijia Jia, Hanxing Wang, and Jianxin Wang. Routing and wavelength assignment for anycast in WDM networks. In *International Conference on Wireless and Optical Communications (WOC)*, pages 301–306, Banff, Alberta, Canada, July 2003.

[TP95] T.K. Tan and J.K. Pollard. Determination of minimum number of wavelength required for all-optical WDM networks using graph colouring. *Electronic Letters*, 31(22):1895–1897, 1995.

[Tur99] Jonathan Turner. Terabit burst switching. *Journal of High Speed Networks (JHSN)*, 8:3–16, 1999.

[VJ03] V.M. Vokkarane and J.P. Jue. Prioritized burst segmentation and composite burst-assembly techniques for QoS support in optical burst-switched networks. *IEEE J. Select. Areas Commun.*, 21(7):1198–1209, September 2003.

[VJS02] Vinod M. Vokkarane, Jason P. Jue, and Sriranjani Sitaraman. Burst segmentation: An approach for reducing packet loss in optical burst switched networks. In *IEEE International Conference on Communications (ICC)*, pages 2673–2677, New York, NY, USA, April-May 2002.

[Whi02] Ian M. White. *A New Architecture and Technologies for High-Capacity Next-Generation Metropolitan Networks*. PhD thesis, Stanford University, US, 2002.

[Wil91] David Williams. *Probability with Martingales*. Cambridge University Press, 1991.

[Wil97] A. E. Willner. Mining the optical bandwidth for a terabit per second. *IEEE Spectrum*, 34(3):32–41, April 1997.

[WMA00a] Xi Wang, Hiroyuki Morikawa, and Tomonori Aoyama. Burst optical deflection routing protocol for wavelength routing WDM networks. In *SPIE/IEEE Opticomm 2000*, pages 257–266, Dallas, TX, USA, October 2000.

[WMA00b] Xi Wang, Hiroyuki Morikawa, and Tomonori Aoyama. A deflection routing protoco for optical bursts in WDM networks. In *Fifth Optoelectronics and Communications Conference (OECC 2000)*, pages 94–95, Makuhari, Japan, July 2000.

[WPRT99] Jorn Y. Wei, Jorge L. Pastor, Ramu S. Ramamurthy, and Yukun Tsai. Just-in-time optical burst switching for multi-wavelength networks. In *Fifth International Conference on Broadband Communications (BC'99)*, pages 339–352, Hong Kong, November 1999.

[WRSK03] Ian M. White, Matthew S. Rogge, Kapil Shrikhande, and Leonid G. Kazovsky. A summary of the HORNET project: A next-generation metropolitan area network. *IEEE J. Select. Areas Commun.*, 21(9), November 2003.

[XPR01] Lisong Xu, Harry Perros, and George Rouskas. A survey of optical packet switching and optical burst switching. *IEEE Commun. Mag.*, 39(1):136–142, January 2001.

[XPR02] L. Xu, H. G. Perros, and G. N. Rouskas. A simulation study of protocols for optical burst-switched rings. In *Proceedings of Networking 2002*, pages 863–874, Pisa, Italy, May 2002. Springer.

[XPR03a] Lisong Xu, Harry G. Perros, and George N. Rouskas. Performance analysis of an edge optical burst switching node with a large number of wavelengths. In *18th International Teletraffic Congress (ITC)*, Berlin, Germany, August-September 2003.

[XPR03b] Lisong Xu, Harry G. Perros, and George N. Rouskas. A queueing network model of an edge optical burst switching node. In *Infocom 2003*, San Francisco, USA, March 2003.

[XPR03c] Lisong Xu, Harry G. Perros, and George N. Rouskas. A simulation study of optical burst switching and access protocols for WDM ring networks. *Comput. Networks*, 41(2):143–160, 2003.

[YMD00] Shun Yao, Biswanath Mukherjee, and Sudhir Dixit. Advances in photonic packet switching: An overview. *IEEE Commun. Mag.*, 38(2):84–94, February 2000.

[YQ97] M. Yoo and C. Qiao. Just-enough-time (jet): A high speed protocol for bursty traffic in optical networks. In *IEEE/LEOS Conference on Technologies for a Global Information Infrastructure*, pages 26–27, August 1997.

[YQ99] Myungsik Yoo and Chunming Qiao. Supporting multiple classes of services in IP over WDM networks. In *Global Telecommunications Conference - (GlobeCom'99)*, volume 1b, pages 1023–1027, 1999.

[ZA95] Zhensheng Zhang and Anthony S. Acampora. A heuristic wavelength assignment algorithm for multihop WDM networks with wavelength routing and wavelength re-use. *IEEE/ACM Trans. Networking*, 3(3):281–288, June 1995.

[ZAA+97] Lisa Zhang, Matthew Andrews, William Aieollo, Sandeep Bhatt, and K. R. Krishnan. A performance comparison of competive on-line routing and state-dependent routing. In *Global Telecommunications Conference - (GlobeCom'97)*, volume 3, pages 1813–1819, 1997.

[ZRP98a] Yohong Zhu, George N. Rouskas, and Harry G. Perros. Blocking in wavelength routing networks, part I: The single path case. Technical Report TR-98-02, Department of Computer Science, College of Engineering, North Carolina State University, February 1998. http://www.csc.ncsu.edu/research/tech-reports/README.html.

[ZRP98b] Yohong Zhu, George N. Rouskas, and Harry G. Perros. Blocking in wavelength routing networks, part II: Mesh topologies. Technical Report TR-98-10, Department of Computer Science, College of Engineering, North Carolina State University, October 1998. http://www.csc.ncsu.edu/research/tech-reports/README.html.

INDEX