A Reconfiguration Method for Energy-Efficient Operation of Multi-Layer Core Networks

Von der Fakultät für Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde
eines Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

vorgelegt von

Frank Feller

geb. in Kirchheim unter Teck

Hauptberichter:	Prof. em. Dr.-Ing. Dr. h. c. mult. Paul J. Kühn
1. Mitberichter:	Prof. Dr.-Ing. Thomas Bauschert (TU Chemnitz)
2. Mitberichter:	Prof. Dr.-Ing. Andreas Kirstädter

Tag der Einreichung:	11. Dezember 2014
Tag der mündlichen Prüfung:	18. April 2016

Institut für Kommunikationsnetze und Rechnersysteme
der Universität Stuttgart

2016

Summary

Environmentalist concerns, resulting regulatory measures, and increasing energy prices mandate efforts toward saving energy in many industries. This also applies to the information and communication technology sector, where communication networks face the particular challenge of limiting their energy consumption while serving exponentially increasing traffic volumes. Traditionally, the access and aggregation section has dominated the energy consumption of transport networks due to the large number of devices in this section. However, new efficient optical access technologies exhibit an energy consumption which is hardly dependent on access rate and traffic volume. In contrast, the energy consumption of the network's core is bound to increase tremendously with the projected traffic growth assuming current technology and operation paradigms. Hence, the core section is bound play a central role in reducing the energy consumption of transport networks. Accordingly, this thesis aims at energy savings in core networks.

While statistical fluctuations are generally small in highly aggregated core network traffic, the load in core networks exhibits significant diurnal variations following human activity. For instance, public Internet exchanges observe nightly periods of traffic as low as 25 % of the busy-hour peak. In contrast, transport networks have traditionally been operated in a static manner, which ensured the network's reliability in the face of challenging optical transmission technology. For the ease of network management, operators maintained the paradigm of static network operation despite technological advances. On the downside, this makes the energy consumption of the network all but independent of the actual traffic load, resulting in a significant waste of energy during low-load periods. To exacerbate energy inefficiencies, active core network resources are generally largely overdimensioned due to traffic uncertainties and for reasons of resilience.

While current core network equipment does not support load-dependent operation, state-of-the-art power-saving techniques could readily be integrated in the current network node architecture. However, scaling the capacity of network resources to the traffic load by such means will fall short of approaching the ideal of power proportionality, i. e. a power consumption proportional to the load. This is mainly due to limited possibilities of scaling the capacity and power consumption of optical resources and to the hierarchical architecture of nodes with chassis and line cards consuming power independently of the number of active interfaces. Accordingly, network reconfiguration aggregating traffic on fewer such resources and components in order to deactivate the remaining ones promises significantly higher energy savings than a pure scaling of resource capacity to load.

Generally, core networks have a multi-layer architecture, consisting at least of an optical circuit-switched lowest and an electrical packet-switched upmost layer. Such configurations offer several degrees of freedom for reconfiguration, including the adaptation of the virtual topology of the upper layer by modifying the set of circuits of the lower one. Maximizing energy savings therefore requires jointly optimizing the configuration in all network layers. In doing so, we need to consider constraints of the precedent network configuration. To the best of our knowledge, previous research on energy-efficient core network reconfiguration either neglects such constraints or applies simple heuristics instead of optimization methods. We contribute to filling this gap.

This thesis designs an optimization-based reconfiguration method for multi-layer core networks aiming at minimal energy consumption and systematically evaluates network configurations obtained by this method regarding energy consumption and potential side effects of the reconfiguration in realistic scenarios. Its first main contribution consists in establishing realistic technological and operational constraints and in formulating an according optimization problem. This includes specifying a dynamic resource operation and power model for core network nodes and establishing a one-step reconfiguration principle implied by reconfiguration time constraints. In addition to minimizing energy consumption, the optimization problem has the objective of avoiding traffic blocking and limiting configuration changes. To solve this problem, we adapt two standard methods of mathematical and heuristic optimization.

The conception and realization of a systematic evaluation of the network configurations obtained by these methods constitutes the second main contribution of this thesis. The evaluation criteria include the energy consumption as well as the impact on network operation and service quality in terms of configuration changes and traffic blocking. By means of a baseline resource scaling scheme, the evaluation isolates the effect of optimizing network configurations from the benefit of introducing load-dependently operated network devices. To realistically capture characteristics of reconfiguration situations, it makes use of traffic traces extracted from measurements, which are scaled to emulate different load points. Results show substantial energy savings at moderate operational impact, which would further increase with improved power proportionality of network nodes.

Zusammenfassung

Der Schutz der Umwelt, daraus resultierende regulatorische Maßnahmen sowie steigende Energiepreise erfordern Anstrengungen zur Reduktion des Energieverbrauchs in vielen Industriezweigen. Dies gilt auch für die Informations- und Kommunikationstechnik. Kommunikationsnetze stehen hierbei vor der besonderen Herausforderung, ihren Energieverbrauch trotz exponentiell steigender Verkehrsvolumina zu begrenzen. Aufgrund der großen Anzahl an Geräten waren die Zugangs- und Aggregationsnetze traditionell maßgeblich für den Energieverbrauch in Transportnetzen. Neue, effiziente optische Zugangstechnologien weisen jedoch einen kaum von Zugangsrate oder Verkehrsvolumen abhängigen Energieverbrauch auf. Im Gegensatz dazu wird der Energieverbrauch der Kernnetze erheblich ansteigen, sofern diese die erwarteten Verkehrsvolumina nach dem heutigen Stand der Technik und den gängigen Betriebsmustern bewältigen müssen. Folglich werden die Kernnetze eine zentrale Rolle bei der Reduktion des Energieverbrauchs in Transportnetzen spielen. Ziel dieser Dissertation sind daher Energieeinsparungen in Kernnetzen.

Während der hoch aggregierte Verkehr in Kernnetzen in der Regel nur geringe statistische Schwankungen erfährt, unterliegt die Last solcher Netze im Tagesverlauf deutlichen Veränderungen, welche sich auf menschliche Aktivitätsmuster zurückführen lassen. So sind an öffentlichen Internetknoten nachts regelmäßig Verkehrslasten von nur 25 % der Last zur Hauptverkehrsstunde zu beobachten. Hingegen wurden Transportnetze traditionell statisch betrieben, um ihre Zuverlässigkeit trotz anfälliger optischer Übertragungstechnik zu gewährleisten. Zwecks einfacheren Managements behielten Netzbetreiber diesen Modus trotz technischer Fortschritte bei. Dadurch ist jedoch der Energieverbrauch der Netze weitgehend unabhängig von der Verkehrslast, was eine erhebliche Energieverschwendung zu Niedriglastzeiten zur Folge hat. Erschwerend kommt hinzu, dass aktive Ressourcen in Kernnetzen üblicherweise großzügig überdimensioniert werden, um unerwartete Verkehrsspitzen aufnehmen sowie Ausfälle ausgleichen zu können.

Zwar unterstützen aktuelle Kernnetzelemente keinen lastabhängigen Betrieb, in anderen Bereichen etablierte Energiespartechniken können jedoch problemlos in die Architektur heutiger Kernnetzknoten integriert werden. Das Ideal einer zur Verkehrslast proportionalen Leistungsaufnahme ist durch die reine Anpassung der Kapazität von Netzressourcen an deren Auslastung aber nicht zu erreichen. Dies ist zum einen bedingt durch Einschränkungen bei der Skalierung der Kapazität und Leistungsaufnahme optischer Ressourcen und zum anderen durch die hierarchische Architektur von Kernnetzknoten, deren Chassis und Schnittstellenkarten eine von der tatsächlichen Anzahl aktiver Schnittstellen weitgehend unabhängige Leistungsaufnahme aufweisen. Daher verspricht eine Rekonfiguration des Netzes, welche Verkehr auf wenige solcher

Ressourcen beziehungsweise Komponenten konzentriert und die nicht mehr genutzten Ressourcen und Komponenten vollständig deaktiviert, wesentlich höhere Energieeinsparungen als das reine Skalieren der Ressourcenkapazität mit der Last.

Im Allgemeinen weisen Kernnetze eine mehrschichtige Architektur auf, welche mindestens aus einer unteren optischen leitungsvermittelten Schicht sowie einer oberen elektrischen paketvermittelten Schicht besteht. Eine solche Konfiguration bietet mehrere Freiheitsgrade für die Rekonfiguration. So kann neben der Verkehrslenkung in der oberen Schicht deren virtuelle Topologie angepasst werden, indem optische Verbindungen der unteren Schicht modifiziert werden. Um maximale Energieeinsparungen zu erzielen, muss folglich die Konfiguration aller Netzschichten gemeinsam optimiert werden. Dabei sind Randbedingungen zu berücksichtigen, die sich aus der vorigen Netzkonfiguration ergeben. Nach bestem Wissen des Autors vernachlässigen bisherige Forschungsarbeiten zur Rekonfiguration von mehrschichtigen Kernnetzen mit dem Ziel der Energieeinsparung entweder solche Randbedingungen, oder sie verwenden einfache heuristische Methoden anstelle von Optimierungsverfahren. Die vorliegende Arbeit trägt dazu bei, diese Forschungslücke zu schließen.

In dieser Dissertation wird ein auf Optimierung basierendes Rekonfigurationsverfahren für mehrschichtige Kernnetze entwickelt, welches auf die Minimierung des Energieverbrauchs abzielt, und es werden in realistischen Szenarien mit diesem Verfahren ermittelte Netzkonfigurationen systematisch hinsichtlich Energieverbrauch und Auswirkungen der Rekonfiguration bewertet. Der erste wesentliche Beitrag dieser Arbeit besteht im Herausarbeiten realistischer technologischer und betrieblicher Randbedingungen für die Rekonfiguration und der Formulierung eines entsprechenden Optimierungsproblems. Hierzu gehört auch die Definition eines dynamischen Modells für den Ressourcenbetrieb und die resultierende Leistungsaufnahme von Kernnetzknoten sowie die Ableitung des Prinzips einer einschrittigen Rekonfiguration aus zeitlichen Randbedingungen. Neben der Minimierung des Energieverbrauchs zielt das Optimierungsproblem darauf ab, Verkehrsblockierungen zu vermeiden und den Umfang der Konfigurationsänderungen zu begrenzen. Zur Lösung des Problems werden zwei etablierte Methoden der mathematischen beziehungsweise heuristischen Optimierung angepasst.

Die Konzeption und Durchführung einer systematischen Bewertung der mit diesen Methoden ermittelten Netzkonfigurationen stellt den zweiten wesentlichen Beitrag dieser Dissertation dar. Die Bewertungskriterien umfassen den Energieverbrauch sowie den Einfluss der Rekonfiguration auf Netzbetrieb und Dienstgüte bezüglich Konfigurationsänderungen und Verkehrsblockierung. Mithilfe eines Referenzschemas, welches lediglich die Ressourcenkapazität an die Last anpasst, wird bei der Bewertung zwischen Effekten der Optimierung der Netzkonfigurationen und dem Gewinn alleine durch die Einführung lastabhängig betriebener Geräte unterschieden. Um die Eigenschaften von Rekonfigurationsszenarien realistisch abzubilden, erfolgt die Bewertung auf Grundlage gemessener Verkehrstraces, welche zur Emulation unterschiedlicher Lastpunkte skaliert werden. Die Ergebnisse zeigen substanzielle Energieeinsparungen bei moderatem Einfluss auf den Netzbetrieb, welche durch eine besser mit der Anzahl optischer Verbindungen skalierenden Leistungsaufnahme von Netzknoten weiter gesteigert werden könnte.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations and Symbols

Abbreviations

3R	reamplification, reshaping, retiming	7
ADM	add-drop multiplexer	43
ALR	adaptive link rate	59
AON	all-optical networks	43
ASIC	application specific integrated circuit	9
ASON	automatically switched optical network	50
BILP	binary integer linear program	12
CAPEX	capital expenditure	44
CMOS	complementary metal-oxide-semiconductor	58
DSL	digital subscriber line	60
DSP	digital signal processing	9
DVFS	dynamic voltage and frequency scaling	59
DXC	digital cross-connect	43
ECMP	equal-cost multi-path routing	56
EDFA	erbium doped fiber amplifier	74
GMPLS	generalized multi-protocol label switching	50
GRASP	greedy randomized adaptive search procedure	23
ICT	information and communication technology	1
ILP	integer linear program	12
IP	Internet protocol	8

IPTV	Internet protocol television	61
LAN	local area network	59
LP	linear program	11
LR-PON	long-reach passive optical network	58
LSP	label switched path	45
MCF	multi-commodity flow (problem formulation)	12
MILP	mixed-integer linear program	12
MPLS	multi-protocol label switching	8
NP	network processor	9
OADM	optical add/drop multiplexer	42
OPEX	operational expenditures	57
OSPF	open shortest path first	60
OSPF-TE	open shortest-path first – traffic engineering	50
OTN	optical transport network	45
OXC	optical cross-connect	10
PBB-TE	provider backbone bridge traffic engineering	8
PON	passive optical network	1
QoS	quality of service	2
REN	research and educational network	112
ROADM	reconfigurable optical add/drop multiplexer	10
RS	resource scaling (baseline reconfiguration scheme)	119
RWA	routing and wavelength assignment	9
SA	simulated annealing	24
SDH	synchronous digital hierarchy	45
SDN	software-defined networking	51
SNMP	simple network management protocol	60
TDM	time division multiplex	45
TRX	transceiver	9

TXP	transponder	9
UPS	uninterruptable power supply	57
VPN	virtual private network	1
WDM	wavelength division multiplex	7
WLAN	wireless local area network	60
WOBAN	wireless-optical broadband access network	60
WON	wavelength-division optical network	42
WSS	wavelength selective switch	10

Symbols

A_{ij}	Set of active circuits in the new configuration from node i to node j	80
A	Set of active circuits in the new configuration	80
a_{ij}	Number of active circuits from node i to node j in the new configuration	90
$a_{\mathrm{LC}\,v}$	Number of active line cards at node v	84
$a_{\mathrm{LCC}\,v}$	Number of active line card chassis at node v	84
$a_{\mathrm{P}\,v}$	Number of active input and output ports at node v	83
$a_{\mathrm{PP}\,v}$	Number of active port pairs at node v	83
$a_{\mathrm{PP}\,ij}$	Number of port pairs occupied by circuits between nodes i and j in the new configuration	90
b_{ij}	Binary variable indicating blocking on the virtual link from node i to node j	81
B	Bit-rate capacity of one optical channel	79
b_{VT}	Number of demands blocked due to partitioning of the virtual topology	97
C	Tuple representing an optical circuit	80
c	Cost of a solution (i. e. objective function value; used for SA)	25
c_{B}	Cost of blocked traffic	85
c_{P}	Energetic cost (total power consumption of the network)	83
c_{R}	Reconfiguration cost	84
d_{sd}	Directed demand value between node s and node d	13
$\boldsymbol{D}$	Matrix of directed demand values	13
$d_v^{(k)}$	Net demand value for commodity k at node v	16
$\bar{d}_{\mathrm{peak}}$	Average peak demand	114
$d_{\mathrm{peak}\,sd}$	Peak demand value between node s and node d	113
$\boldsymbol{D}_{\mathrm{peak}}$	Peak demand matrix	113
D_{r}	Tuple representing a routed demand	81
E	Set of all network links	13
E_{f}	Set of virtual links with feasible circuit realizations	90
E_{fu}	Set of undirected virtual links with feasible circuit realizations	90

E_p	Set of all physical links	79
E_v	Set of active virtual links	80
$f_{ij}^{(k)}$	Flow of commodity k from node i to node j	16
F	Objective function (for mathematical programming)	11
G	Network graph	13
G_p	Graph of the physical network	79
$i_{\text{P}\,v}$	Index of a port pair of node v	80
K	Set of all commodities	13
L	Maximum transparent optical reach	82
l_{mn}	Geographical length of the physical link from node m to node n	79
$m_{\text{PP}\,ij}$	Number of port pairs occupied by circuits between nodes i and j (including transients)	90
N	Neighborhood function (in local search)	22
$n_{\text{F}\,mn}$	Number of installed fibers on the physical link from node m to node n	79
N_λ	Maximum number of optical channels per fiber	79
$n_{\text{LC}\,v}$	Number of installed line cards at node v	79
N_LC	Maximum number of line cards per line card chassis	74
$n_{\text{LCC}\,v}$	Number of installed line card chassis at node v	79
$n_{\text{PP}\,v}$	Number of installed port pairs at node v	79
N_PP	Maximum number of port pairs per line card	73
p_{ij}	Number of active circuits from node i to node j in the previous configuration	90
P_{ij}	Set of active circuits in the previous configuration from node i to node j	80
P	Set of active circuits in the previous configuration	80
P_LC	Power consumption of one active line card	74
P_LCC	Power consumption of one active line card chassis	74
p_LR	Probability of removing a virtual link (in a perturbation)	100
P_P	Power consumption of one active line card port	73
P_T	Power consumption per electrically switched traffic unit	74
R	Set of all routed demand sets	81

r_{ij}	Number of modified (i. e. established or torn down) circuits from node i to node j	90
R_{sd}	Set of routed demands between node s and node d	81
$\mathbb{R}$	Set of all real numbers	140
$\boldsymbol{R}_{\mathrm{C}}$	Route in the physical topology as a sequence of nodes	80
$\boldsymbol{R}_{\mathrm{D}}$	Route in the virtual topology as a sequence of nodes	81
$\mathbb{R}_0^+$	Set of all non-negative real numbers	140
$\mathbb{R}^+$	Set of all positive real numbers	140
S	Solution space (for mathematical programming)	11
t_{ij}	Traffic on virtual link from node i to node j	81
ΔT	Time interval between reconfiguration events	76
t	Traffic volume (of a routed demand)	81
$t_{\mathrm{B}\,ij}$	Volume of traffic blocked on the virtual link from node i to node j	91
$t_{\mathrm{B\,VT}}$	Volume of traffic blocked due to partitioning of the virtual topology	97
T_{Cm}	Circuit modification time (with full power consumption)	125
$t_{\mathrm{T}\,v}$	Volume of electronically switched transit traffic at node v	84
V	Set of all network nodes	13
X	(Candidate) solution of the SA based method	98
$\boldsymbol{x}$	Solution (for mathematical programming)	11
$\mathbb{Z}$	Set of all integer numbers	140
$\mathbb{Z}_0^+$	Set of all non-negative integer numbers	140
$\mathbb{Z}^+$	Set of all positive integer numbers (i. e. natural numbers)	140

α	Number of accepted solutions for the current temperature (in SA)	102
α_{max}	Maximum number of accepted solutions per temperature step (in SA)	27
β_{D}	Penalty for blocking a demand	97
β_{L}	Penalty for blocking on a virtual link	84
β_{T}	Penalty for blocking a traffic volume	85

δ	Reconfiguration penalty	84
η	Number of iterations since last improvement of accepted solution cost (in SA)	103
η_{max}	Maximum number of iterations since last improvement of accepted solution cost (in SA)	103
η_{r}	Iteration counter for relative accepted solution cost range evaluation (in SA)	103
Γ	A large (positive) constant	81
γ	Cooling factor (for SA)	26
ν	Number of iterations for the current temperature (in SA)	26
ν_{max}	Maximum number of iterations per temperature step (in SA)	25
ρ	Minimum realtive range of accepted solution cost (in SA)	103
σ_{dim}	Dimensioning factor	114
ϑ	Temperature (control parameter for SA)	25
ϑ_0	Initial temperature for SA	25

1 Introduction

Environmentalist concerns, the limitation of sources of fossil fuels, and the limited availability of alternative sources of renewable energy have made energy consumption a general concern, bringing forth regulatory measures and economic considerations. Accordingly, reduction of energy consumption has become a priority for many industries. Information and communication technology (ICT) is often named as an enabler for energy conservation in other sectors, e. g. by reducing human travel thanks to remote collaboration and videoconferencing. However, the ICT sector itself consumes a non-negligible amount of energy: ICT devices account for approximately two percent of the global electricity consumption, and their share is rising [1]. According to several survey papers [2, 3], it reaches up to ten percent in developed countries such as the United Kingdom. The network infrastructure of the Internet accounts for one percent of the electricity consumption in these countries [4]. This share is expected to increase as the projected traffic growth exceeds efficiency gains in silicon technology [2]. Hence, further approaches to increase the energy efficiency of the ICT sector in general and of the network infrastructure in particular are highly desirable.

We further discuss the energy savings potential in transport networks in section 1.1 and define the contribution of this thesis in section 1.2. Section 1.3 finally gives an outline of this monograph.

1.1 Energy Consumption and Operation of Transport Networks

Transport networks, i. e. the communication network infrastructure providing wide-area data transport for e. g. Internet, virtual private network (VPN), and voice telephony services, consist of several network sections differing in their power consumption characteristics. In the access and aggregation part, which provides connection to individual customers and aggregates their traffic towards metro and core networks, network devices have to deal with relatively low data rates and traffic volumes, generally resulting in moderate power consumption per device. In contrast, devices of the core network, which interconnects metropolitan areas on a national, continental, or global scale, exhibit high power consumption due to the large traffic volumes they process at high data rates. Nevertheless, the access and aggregation part dominates the power consumption of the network today, since these sections comprise a huge number of devices whereas core network locations are few. This is projected to change as new optical access technologies, namely passive optical networks (PON), support increasing data rates at constant power consumption. In contrast, the power consumption of core nodes is expected to grow with the capacity increase required to accommodate future traffic volumes [4, 5]. We therefore focus on energy savings in the core network.

Being highly aggregated, core network traffic generally exhibits only moderate statistical fluctuations. It is however characterized by significant diurnal variations following human activity. For instance, at public Internet exchanges, we observe nightly periods with traffic as low as 25 % of the busy-hour peak [6]. In contrast, core networks are operated in a static manner, which has traditionally ensured network stability in the face of challenging optical transmission technology. This results in almost traffic-independent, constant power consumption. In addition, network components are generally only moderately utilized even during busy-hour since overdimensioned due to traffic uncertainty. Both theoretical work and recent experimentation [7] give reason to believe that future optical core network technology will allow a more dynamic operation, which is already supported by standardized control plane protocols. This offers the potential for significant energy savings in core networks by adapting the configuration of their resources to the traffic load.

Many network resources do not allow a fine-grained scaling of their capacity and power consumption with the traffic load. This applies to both optical channels and network nodes featuring hierarchical structures. It may therefore be beneficial to modify the network configuration beyond scaling the capacity of resources to their current load, e. g. by rerouting in order to concentrate traffic on some of such resources and deactivate the others. The multi-layer architecture which most core networks adhere to, with electrical packet-switching in the topmost layer and optical circuit-switching in the lowest one, offers many degrees of freedom for such reconfiguration.

Since core networks carry large quantities of aggregated traffic, service outages would affect many customers. In addition, services for business customers (such as VPN) are governed by service level agreements, which may specify parameters like maximum packet delay in addition to availability metrics. For these reasons, reconfiguration must not degrade the quality of service (QoS) offered by core networks. In particular, we have to avoid service interruptions during the potentially time-consuming adaptation of resource settings. For optical resources, we therefore have to resort to a *make-before-break* approach.

A body of work studies reconfiguration of core networks aiming at energy savings. However, only a limited number of publications exploit the degrees of freedom offered by the multi-layer architecture. These contributions fall into two groups: either they optimize network configurations to determine maximum energy savings while disregarding reconfiguration-related QoS constraints, or they determine small incremental modifications of the network configuration by means of heuristics. To the best of our knowledge, no research has targeted energy-oriented multi-layer network reconfiguration under realistic constraints by means of optimization. We contribute to filling this gap.

1.2 Contribution of the Thesis

This thesis makes two central contributions: First, it derives and formalizes an optimization problem for finding a new network configuration from realistic technological and operational constraints. Second, it systematically evaluates network configurations obtained by solving this problem with respect to energy savings and operational impact in realistic scenarios. Means to implement the centrally determined configurations in the network, such as signaling protocols, are beyond the scope of this monograph.

1.2.1 Optimization Problem Formulation

Since the dynamic operation of core network resources is a novel concept unsupported by current equipment, we start by deriving a dynamic resource operation and power model for core network nodes from the hierarchical structure of such devices and state-of-the-art power-scaling techniques. Based on this model along with technological constraints governing the reconfiguration time of optical resources and requirements on network operation in terms of QoS, we formulate an optimization problem to determine a new configuration of minimal power consumption for a two-layer network. In accordance with the constraints, the reconfiguration fulfills the *hitless* property ensuring uninterrupted service.

We solve this optimization problem by adapting two standard methods of mathematical optimization and heuristic optimization, respectively. While this adaptation involves some design choices, the conception of the solution methods is not in the focus of this thesis and we do accordingly not discuss or evaluate design alternatives. Since the mathematical optimization method provides an indication of the quality of its solutions, these methods allow studying properties of the solutions of our reconfiguration problem independently of the design of the specific solution method.

1.2.2 Systematic Evaluation in Realistic Scenarios

Network reconfiguration scenarios are generally characterized by limited changes in the traffic load between subsequent configurations as well as constraints due to a previous configuration. In order to capture these properties in a realistic way for the evaluation, we simulate a sequence of reconfigurations based on traffic traces extracted from measurements. While our study is accordingly restricted to a set of network topologies for which such traces are available, we ensure a sufficient diversity in the scenarios by using several such topologies of varying size. In addition, we investigate the effect of different network resource dimensioning and scale the traffic load. This allows us to isolate and explain the influence of different factors on the evaluation metrics.

The evaluation centers on the energy savings achieved by the solutions of our reconfiguration problem as well as the impact of the reconfiguration on network operation with possible effects on QoS. Rather than referring to static network operation according to the current paradigm, which would yield energy savings strongly dependent on network dimensioning, we apply a reference resource scaling scheme exploiting the assumed possibilities of dynamic resource operation without otherwise changing the network configuration. In this way, we are able to isolate the effect of spending the effort of computing optimized configurations in terms of both energy savings and operational impact.

1.3 Outline

The remainder of this monograph is structured in two parts consisting of two chapters each. The first part covers the technical and methodological background along with a survey of the related literature. The second one details the main contributions of this thesis.

Chapter 2 prepares the ground for the contribution of this thesis by introducing the fundamental scenario of multi-layer core networks and presenting relevant aspects of the optimization methodology. Regarding the latter, it discusses a mathematical programming formulation commonly used for expressing networking problems after giving a general introduction to optimization. It then provides a classification of optimization methods and details one meta-heuristic method applied in this monograph.

Chapter 3 reviews the state of the art in two fields related to our reconfiguration problem: the design and reconfiguration of multi-layer networks, and energy-aware networking. It starts by providing an overview and classification of related publications of both fields. Then, it discusses the criteria characterizing research on network design and reconfiguration in depth, addressing assumptions on networking scenario, degrees of freedom, objectives, and methods. It finally surveys contributions on various approaches to energy-aware networking and details two aspects of particular importance to this monograph: power consumption modeling and load-dependent network operation.

The first central contribution of this thesis is provided in chapter 4, which derives the optimization problem along with appropriate solution methods from realistic constraints. To prepare for the problem definition, the chapter details our assumptions on technological and operational conditions regarding dynamic power model and resource modification times. These imply the one-step reconfiguration principle governing our approach. After stating further requirements, the chapter provides a mathematical formulation of the entire reconfiguration problem. It finally describes the solution methods, which decompose the problem into the realization of optical circuits addressed by heuristics and the optimization of virtual topology and traffic routing performed by a mathematical method and a heuristic method.

Chapter 5 details the second central contribution of this thesis: the systematic evaluation of our reconfiguration method under realistic conditions. It starts by discussing requirements on the evaluation scenario and defines according settings for our evaluation. In addition, it addresses the value assignment to parameters inherent to our reconfiguration method, and it defines the baseline resource scaling scheme. On this basis, it plots results for metrics covering energy consumption and operational impact and explains scenario-dependent effects shown in the plots. It concludes with a discussion of general observations yielding an overall appraisement of the reconfiguration method.

Chapter 6 finally provides a conclusion and an outlook on future work.

2 Optimization of Multi-Layer Core Networks

By delimiting the network scenario and describing the relevant methodology, this chapter lays the foundations for defining and solving the questions we deal with in this monograph. Section 2.1 situates the network segment we consider and introduces the fundamental properties of multi-layer core networks and their nodes. In section 2.2, we give a brief introduction to optimization problems and their mathematical formulation. We then describe different variants of a common formulation of network optimization problems in section 2.3. Section 2.4 introduces and classifies solution methods for such optimization problems. We finally give more details on one heuristic optimization method which we apply in this monograph in section 2.5.

2.1 Multi-Layer Core Networks

Transport networks enable a multitude of communication services by providing data transport between user equipment and servers. In addition to the data plane realizing the transport capabilities, they feature control and management planes serving for the configuration and monitoring of data plane functions. Since this monograph addresses methods to determine configurations of data plane entities, the presentation in this section likewise focuses on the data plane. While the control and management planes are essential in distributing and implementing the determined configuration in the concerned network elements, this realization aspect is out of scope of this monograph.

The communication network transporting data between remote end systems is composed of several segments differing in technology and structure. Figure 2.1 illustrates these segments. Customer devices may connect either directly to access networks or to local area networks, e. g. in office buildings or private homes, which in turn offer a connection to an access network. The access and aggregation networks, which may include wireless links, usually feature a tree topology and concentrate traffic towards metro nodes. For redundancy reasons, customers may also have several access connections. Metro networks transport the aggregated traffic within a certain region. Traditionally, they feature a ring topology, which is bi-directional for reasons of resilience. Alternatively, one finds meshed topologies in this segment. Finally, metro networks are interconnected by national, continental, and inter-continental core networks with generally meshed topologies. Since such topologies offer promising degrees of freedom for reconfiguration, the methods developed in this monograph target core networks. However, they are likewise applicable to metro networks with meshed topologies and similar architectures.

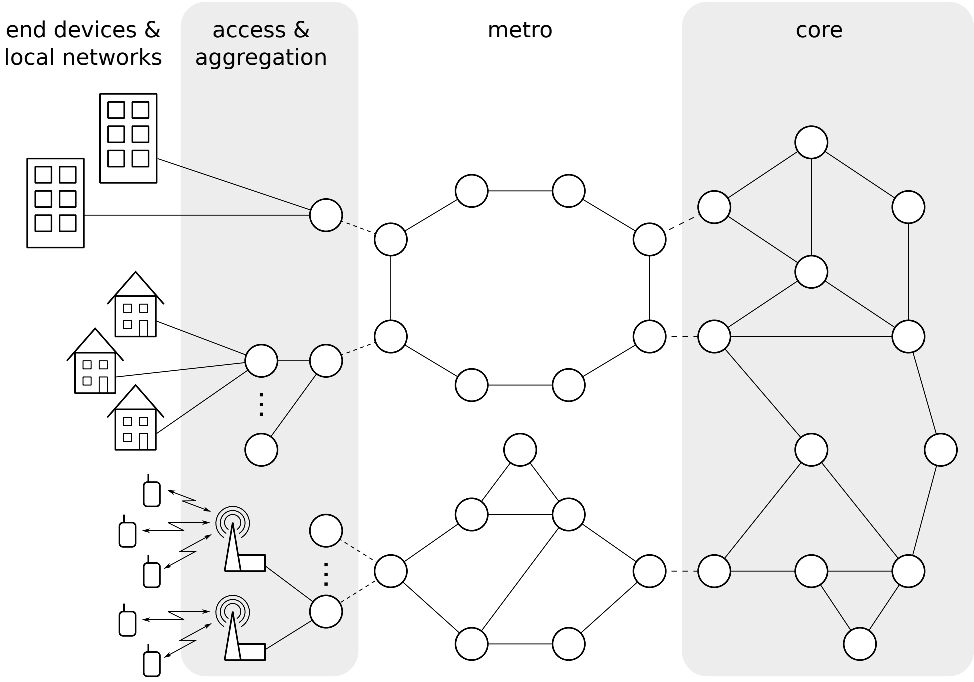


Figure 2.1: Network segments and their typical topologies

Core networks generally have multi-layer architectures. Section 2.1.1 discusses the characteristics of such network architectures. We then detail the variant we assume in the remainder of this monograph in section 2.1.2. Section 2.1.3 finally describes the relevant aspects of the node architecture for this variant.

2.1.1 Principle of Multi-Layer Networks

The ITU-T recommendation G.800 [G.800] specifies the functional architecture of transport networks in a technology-independent, generic way. A transport network is composed of several layer networks[1], which all offer transport functions, relying on the next lower layer for their implementation. Omitting most of the very specific terminology of the ITU recommendation, this section outlines the interplay of the layer networks and their major characteristics.

A layer network features a certain topology through which it transports data in a layer-specific format. It may apply either packet-switching or circuit-switching. In the former case, resources are only allocated while data is transmitted, while a constant amount of resources is reserved for a connection in the latter case. Packet-switched operation is either connectionless or connection-oriented, i. e. either a routing decision is taken in each network node for each packet based on its destination, or the route of the packets through the layer network is defined upon setup of a connection. By principle, circuit-switching is connection-oriented. The functional architecture of connection-oriented transport networks is further specified in the ITU-T recommendation G.805 [G.805].

[1]In the ITU-T terminology, *layer network* designates one layer of a multi-layer network.

Vertically adjacent layer networks have a client-server relation. In case of a connection-oriented lower layer, connections in this layer imply links in the topology of the upper layer. For increased capacity, one link of the upper layer may be supported by several lower-layer connections, which may be routed along different paths.

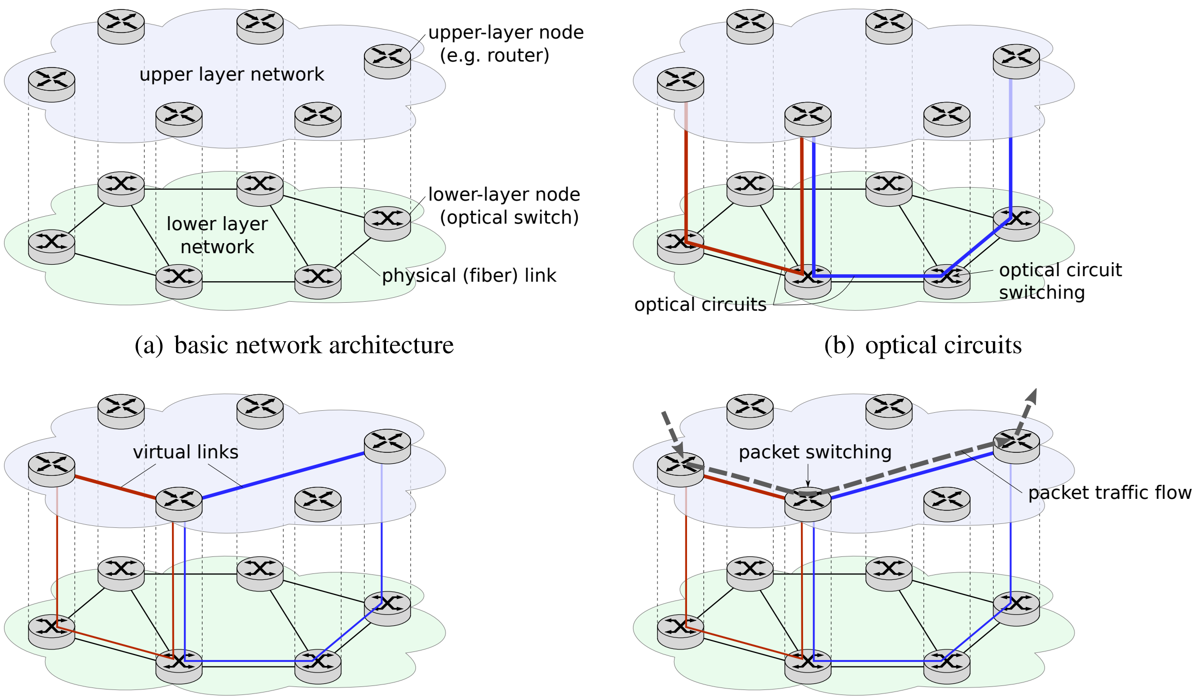
2.1.2 Two-Layer Configuration

In this monograph, we assume a configuration consisting of two layer networks with different characteristics. The upper layer applies packet-switching, whereas the lower layer is circuit-switched. This constellation corresponds to one set of practically relevant transport network architectures. Since we assume a circuit capacity significantly larger than typical elementary traffic flows, the setting implies a trade-off between transporting traffic in the two layers and thus brings about the essence of multi-layer network design problems. We assume connection orientation in both layers, which is in line with wide-spread transport network operation paradigms. By supporting explicit traffic routes in the upper layer, it additionally facilitates defining optimized configurations.

While the reconfiguration problem and the according solution methods we define in chapter 4 are in principle applicable to any such layer network constellation, we take physical constraints of a particular technological option for the lower layer into consideration. We here assume a reconfigurable optical network applying wavelength division multiplex (WDM). Mukherjee [8] gives an in-depth discussion of optical WDM networks. The WDM layer network is made up of optical fibers and the network nodes they interconnect, i. e. its topology is defined by the existence of physically installed resources. We accordingly refer to this layer network as *physical network* or, alternatively, *substrate network*, to its topology as *physical topology*, and to its links as *physical links* or *fiber links*. Figure 2.2(a) illustrates these physical resources in the context of the two-layer network architecture.

Conceptually, as depicted in Figure 2.2(b), a circuit of the lower layer interconnects two nodes of the upper layer. I. e. these nodes serve as termination points of the circuit. In case of WDM, a circuit occupies a certain share of the optical spectrum on each fiber it traverses. This spectrum is characterized by the *wavelength* corresponding to its center frequency, and we accordingly refer to it as *wavelength channel*. If optically switched from one fiber to another one, a circuit has to use channels of the same wavelength unless the switching nodes have wavelength conversion capability. It carries data at a constant rate, which is defined by modulation format, symbol rate, and coding. An optical circuit is also called a *lightpath* [9].

Due to attenuation, the optical signal requires amplification after certain distance. While this is achieved efficiently by fiber amplifiers in the optical domain, it introduces noise, which in turn limits the distance a signal can travel before the data becomes undetectable. Hence, conversion to the electrical domain is necessary when attaining the *maximum optical reach*. At this point, the optical circuit is either terminated, handing the data to the packed-switched upper layer, or electrical signal regeneration is performed before converting the signal back to the optical domain. The latter is called 3R regeneration (reamplification, reshaping, retiming) and does not require extracting the structure of the upper-layer data from the transmitted bit-level data stream. We discuss time constraints imposed on reconfiguration by optical amplifiers in section 4.1.3.



(a) basic network architecture (b) optical circuits

(c) resulting virtual topology (d) packet traffic routing

Figure 2.2: Illustration of two-layer networking concepts

The upper layer network may implement any connection-oriented packet-switching technology. One option commonly assumed in network design and reconfiguration studies (cf. section 3.2) is Internet protocol (IP) with multi-protocol label switching (MPLS) [G.8110/Y.1370, RFC 3031]. The latter extension adds switching labels for connection-oriented operation to the otherwise connectionless operation of IP routers. In addition, the IP over WDM architecture is treated as a promising option for energy-efficient transport networking [10]. However, the upper layer could likewise feature an Ethernet transport technology like provider backbone bridge traffic engineering (PBB-TE) [802.1ay, G.8010/Y.1306].

The topology of the upper layer consists of the nodes and so-called virtual links realized by circuits of the lower layer. As illustrated in Figure 2.2(c), it is independent of the physical topology. We accordingly refer to it as *virtual topology*, or alternatively as *logical topology* or *lightpath topology* [11]. Figure 2.2(d) depicts the routing of a packet traffic flow from the leftmost to the rightmost network node in the virtual topology.

Summing up the degrees of freedom in the two layer networks, we obtain four interrelated dimensions in the configuration of such a two-layer network[2]:

- defining (the links of) the virtual topology of the upper layer,
- routing traffic in this virtual topology,
- routing of optical circuits to realize the virtual topology,

[2] Cf. chapters 7 and 8 of [8].

- wavelength assignment to the optical circuits (possibly under continuity constraints).

The latter two aspects are subsumed as the *routing and wavelength assignment* (RWA) problem. The first two form the *virtual topology design problem*. Finding an optimal network configuration generally requires addressing all four dimensions simultaneously and therefore constitutes a highly complex problem.

2.1.3 Network Node Architecture

Network nodes of multi-layer core networks generally consist of one or several modular devices for each layer network. This section describes the node architecture of a typical representative of the two-layer configuration according to section 2.1.2: IP/MPLS over WDM networks. We particularly focus on the most relevant components in terms of power consumption and resource adaptation.

The architecture of core network nodes of different vendors is similar. In the following, we present this architecture along with the abstraction researchers derived as a basis for modeling equipment cost [12, 13] and power consumption [14, 15].

2.1.3.1 IP/MPLS Router

Network nodes forwarding packets in the IP/MPLS layer are generally referred to as routers. Figure 2.3 gives their typical architecture. They consist of one or several line card chassis featuring power supply, cooling, chassis control, and routing engines, as well as a passive backplane interconnecting slots for line cards and switch fabric cards. Switch fabric cards are essential for exchanging packets between the line cards of one chassis. If a router is composed of several line card chassis to achieve the required capacity, these chassis are interconnected by means of a fabric card chassis featuring further switching fabric cards. In abstract models, all these components (excluding line cards) are subsumed in the *base node*.

Line cards add the actual packet processing capability and interfaces towards the optical layer and other networks to the base node. One can functionally subdivide them into *slot cards*, or forwarding engine cards, and *port cards*. The former comprise network processors (NP) for packet processing, memory serving as packet buffer and for storing forwarding information, and further application specific integrated circuits (ASIC). The models assume only one type of slot card, which has a processing capacity equaling the bandwidth of the backplane, and leave the differentiation of line cards to different port card configurations.

Port cards form the interface between a slot card and one or several transceivers (TRX). When interfacing to the optical layer network, the TRXs bi-directionally convert the signal between the electrical and the optical domains. There are two classes of optical TRXs: So-called colorless short-reach TRXs are connected by dedicated fibers to transponders (TXP) translating the signal to an appropriate WDM wavelength for long-haul transmission. For this, the TXP first converts the signal back to the electrical domain and applies appropriate coding and modulation, which it effectuates by digital signal processing (DSP). The advantage of this configuration is a better

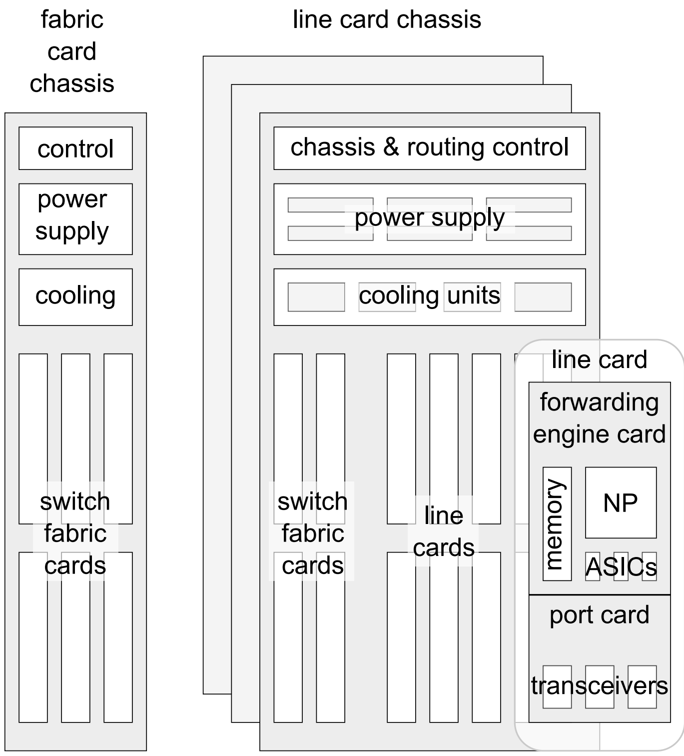


Figure 2.3: IP/MPLS router architecture

separation of the layer networks, which facilitates using equipment from different vendors. Alternatively, port cards may feature colored long-reach TRXs providing the functionality of the WDM side of a TXP, thereby avoiding the conversion of the signal to the optical domain for the transport between line card port and TXP. Due to space constraints, this comes at the price of fewer TRXs per line card compared to short-reach interfaces.

2.1.3.2 WDM Node

The architecture of WDM nodes depends on their switching capabilities and their nodal degree, i. e. the number of fiber links to neighbor nodes. For a nodal degree of two, reconfigurable optical add/drop multiplexers (ROADM) are sufficient, whereas higher nodal degrees require more general optical cross-connects (OXC). Such devices may be composed of optical splitters and combiners, WDM multiplexers and demultiplexers, wavelength selective switches (WSS), and filters. In addition, they comprise diverse amplifiers and they may include wavelength converters. Those components may be discrete devices installed in optics racks providing power supply, cooling, and control. Unlike routers of the upper layer, the optical nodes do not exhibit a hierarchical structure beyond the physical housing in racks. Figure 2.4 shows an exemplary architecture of an OXC with full wavelength conversion capability. For further details and alternative architectures, the reader is referred to [8, 16].

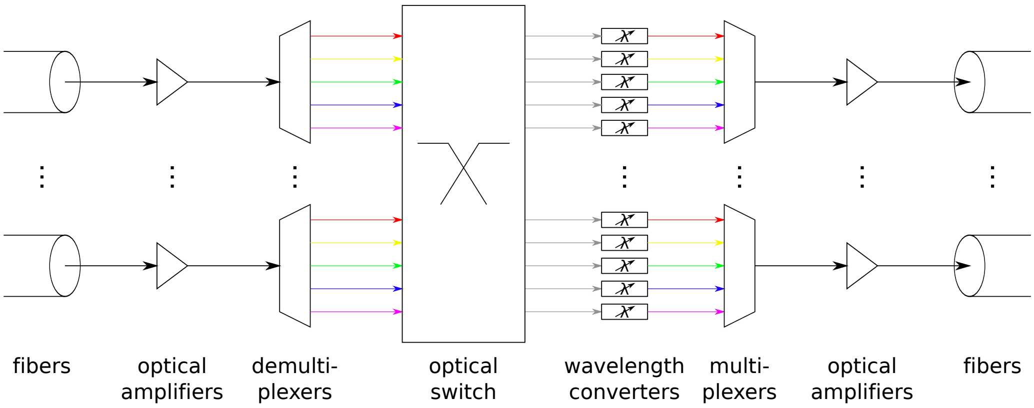


Figure 2.4: Exemplary architecture of an OXC with wavelength conversion capability (derived from [16])

2.2 Introduction to Optimization

Optimization means solving the decision problem of finding a solution out of a set of candidates which is optimal in regard to a certain objective [17]. In many fields of application, solutions to optimization problems are intuitively chosen by human operators based on experience or following predefined procedures. However, this approach tends to fail to identify good solutions for practically relevant problems with a large set of possible solutions. This gave rise to the field of operations research and formal methods to solve optimization problems.

One theory to formalize optimization problems, which underlies the problems and methods discussed in this monograph, is *mathematical programming*[3]. It describes a solution by a vector $\boldsymbol{x}$ of n variables belonging to the *(feasible) solution set* or *optimization space* $S \subseteq \mathbb{R}^n$ [4]. The objective of the optimization is expressed by a real-valued objective function $F : S \to \mathbb{R}$. Thus, the basic form of a mathematical programming problem is[5]

$$\textbf{minimize} \quad F(\boldsymbol{x}) \tag{2.1}$$
$$\textbf{subject to} \quad \boldsymbol{x} \in S \tag{2.2}$$

Linear programming problems constitute an important subclass of mathematical programming problems. Many practically relevant problems are expressible as *linear programs* (LP). In an LP, the objective function is linear in the solution variables and the solution space is delimited by a set of linear inequations. Efficient solution methods exist for such problems if the variables assume continuous values, i. e. $\boldsymbol{x} \in S \subseteq \mathbb{R}^n$. Using a vector $\mathbf{c}$ of n cost coefficients and a $m \times n$ coefficient matrix $\mathbf{A}$ along with a right-hand side vector $\mathbf{b}$ of size m to constrain S, we can express the general LP in vector notation (cf. chapter 5 of [18]):

$$\textbf{minimize} \quad F(\boldsymbol{x}) = \mathbf{c}\boldsymbol{x} \tag{2.3}$$
$$\textbf{subject to} \quad \mathbf{A}\boldsymbol{x} \leq \mathbf{b} \tag{2.4}$$

[3] The following presentation is inspired by annex A of [18].

[4] We describe the fundamental mathematical notations used in this monograph in annex A.

[5] Without loss of generality, we only discuss minimization problems. Inverting the sign of the objective function allows expressing maximization problems in the same form.

Some optimization problems are only expressible in linear form when using discrete variables. If all solution variables have to assume integer values, i. e. $S \subseteq \mathbb{Z}^n$, we speak of *integer linear programs* (ILP). As a special case, the problem may only contain decision variables assuming the values zero or one ($S \subseteq \{0,1\}^n$) and constitute a *binary integer linear program* (BILP). If finally a subset of variables has to be integer and the rest is real-valued, we are facing a *mixed-integer linear program* (MILP). All those problems are more difficult to solve than LPs with only continuous variables [19].

A finite number of feasible solutions characterizes the so-called *combinatorial optimization problems*. BILP and ILP with bounded value ranges fall into this category. Accordingly, a subset of combinatorial problems is expressible as an ILP. Hence, optimization methods conceived for either problem type are applicable to this set of problems.

The theory of complexity aims at quantifying the effort required to solve an optimization problem [17, 20]. It generally does so by classifying the scaling behavior of the (abstract) execution time of the best known algorithm with the problem size. Different problem size metrics are conceivable, including the numbers of variables n and constraints m for integer and non-integer LPs and that of feasible solutions for combinatorial problems. In practice, however, the structure of the solution space is decisive for the time needed to solve a specific problem instance (cf. section 2.4.1 and [18]).

For non-integer LP problems, algorithms with polynomial time complexity exist [18]. Such problems are considered as *easy problems* [17], since instances of considerable size are solvable in relatively short time. In contrast, problems involving discrete variables are often NP-complete, i. e. they can only be solved by a polynomial algorithm on a non-deterministic machine [20]. The best known deterministic algorithms for these problems have an exponential time complexity, which results in prohibitively long computation times for large problem instances. Section 2.4 discusses several approaches to cope with this issue.

2.3 Multi-Commodity Flow Problems

The *multi-commodity flow* (MCF) or *multi-commodity network flow* [18] formulation is the standard way to express network optimization problems for mathematical programming. Its fundamental equations are linear, allowing us to represent many network optimization problems as LPs, ILPs, or MILPs. All linear programming formulations in the related literature reviewed in chapter 3 are MCF problem variants.

MCF problems target scenarios where several flows of distinguishable type need to be transported over one network, sharing its resources. They are thus e. g. applicable to road transportation networks serving flows of vehicles between specific origin and destination points (such as towns). Likewise, we can apply MCF formulations to communication networks catering to different (data) traffic flows. A flow in the problem formulation may represent network traffic of different granularity, reaching from transmissions triggered by individual customers to large traffic aggregates. To cope with complexity, it is a common approach in core network optimization to consider aggregate traffic flows between the network nodes, which generally represent metropolitan areas [8].

In the following sub-section, we introduce the fundamentals of different variants of the MCF formulation by means of simple example problems in single-layer network scenarios. Therein, our choice of symbols has partly been inspired by [8, 21]. Section 2.3.1.3 then outlines the extension of the formulations for multi-layer network optimization. The literature surveyed in chapter 3 contains many extensions of the basic problem formulations for different purposes. The problem we specify in section 4.4.2.2 is a further such example.

2.3.1 Fundamental Formulations

The input parameters of any MCF problem include a description of the network topology and the traffic which has to be accommodated. We can describe the topology by the network graph[6]

$$G = \langle V, E \rangle \tag{2.5}$$

Herein, V is the set of network nodes (vertices) and E is the set of all directed links (edges) between pairs of different nodes[7]:

$$E \subseteq \{(i, j) \in V \times V | i \neq j\} \tag{2.6}$$

In the context of network optimization, the traffic to be transported is generally referred to as *(traffic) demand*. We designate the demand volume, expressible e. g. as a bit rate or packet rate, to be carried from source node s to destination node d ($s, d \in V$) by $d_{sd} \in \mathbb{R}_0^+$. As for the links, we disregard traffic local to one node, i. e. $d_{vv} = 0 \ \forall v \in V$. The *demand matrix* consists of all node-to-node demand values:

$$\boldsymbol{D} = (d_{sd})_{s,d \in V} \tag{2.7}$$

The definition of further elements, in particular that of flow variables, differs between the formulation variants. In the following, we discuss the *link-node* formulation, its aggregated form, and the *link-path* formulation. We borrowed these terms from Pióro and Medhi [18], who provide an in-depth introduction to different formulation variants and specific formulations for a large set of network design problems. Slightly different denominations for these formulation types are found elsewhere in the literature.

2.3.1.1 Disaggregated Link-Node Formulation

The link-node MCF formulation allows an intuitive description of network optimization problems with few constraints on the routing of demands. In its disaggregated form, we consider the traffic flowing from one source node s to one destination node d as one *commodity* and accordingly define the set of commodities

$$K = \{(s, d) \in V \times V | d_{sd} > 0\} \tag{2.8}$$

[6] See annex A for the mathematical notation conventions adhered to in this monograph.

[7] As long as the internal structure of a node is not modeled, we exclude self-loops which are not reasonably used by inter-node traffic flows.

MCF formulations express the routing of traffic by so-called *flow variables*. The link-node formulation devises one such variable for each link and each demand, i. e.

$$f_{ij}^{(sd)} \in \mathbb{R}_0^+ \qquad \forall\,(s,d) \in K, (i,j) \in E \tag{2.9}$$

For ease of notation, we assume flow variables to be defined for all node pairs and not only on links, i. e. $f_{ij}^{(sd)} \in \mathbb{R}_0^+ \ \forall\,(s,d) \in K, (i,j) \in V \times V$. By additional constraints, we ensure $f_{ij}^{(sd)} = 0 \ \forall\,(s,d) \in K, (i,j) \in (V \times V) \setminus E$.

Flow conservation is the central constraint on the flow variables. It stipulates that traffic of one commodity may only enter and leave the network at the respective source and destination nodes. In all other nodes, the sums of traffic of this commodity on incoming and outgoing edges, respectively, need to be balanced. Formally, the constraint reads

$$\sum_{j \in V \setminus \{v\}} f_{vj}^{(sd)} - \sum_{i \in V \setminus \{v\}} f_{iv}^{(sd)} = \begin{cases} d_{sd} & \text{if } v = s \\ -d_{sd} & \text{if } v = d \\ 0 & \text{otherwise} \end{cases} \qquad \forall\,(s,d) \in K, v \in V \tag{2.10}$$

Based on these elements, which are generic to all MCF problems in disaggregated link-node formulation, we exemplarily define a number of elementary optimization problems. Let us first consider a *capacitated* design problem, i. e. a problem where the quantity of available resources is given. Such problems occur e. g. for infrequent reconfigurations allowing to completely change the network setup except for the installed hardware. In the simplest case, we have a capacity bound $B_{ij} \in \mathbb{R}_0^+ \ \forall (i,j) \in V \times V$ for each link, where $B_{ij} = 0$ if $(i,j) \notin E$. In order to enforce these bounds, we have to accordingly limit the sum of the flows of all commodities on the respective link, i. e.

$$\sum_{(s,d) \in K} f_{ij}^{(sd)} \leq B_{ij} \qquad \forall\,(i,j) \in V \times V \tag{2.11}$$

To obtain an optimization problem, we need to complement the constraints by an objective. For instance, we might want to minimize the resource utilization over all links in absolute terms, which translates into minimizing the total flow in the network:

$$\textbf{minimize} \qquad \sum_{(i,j) \in V \times V} \sum_{(s,d) \in K} f_{ij}^{(sd)} \tag{2.12}$$

With its real-valued variables, the problem defined by Equation (2.10) to (2.12) constitutes an LP. It is applicable e. g. to packet-switched networks with packet flows of arbitrary bit rates where packets of one flow are routable on multiple alternative paths in arbitrary ratios. This is generally given in the upper layer of IP/MPLS over WDM core networks where the high level of traffic aggregation allows us to approximate any traffic splitting ratio without bifurcating elementary traffic flows (e. g. transmission control protocol (TCP) streams). The latter may be undesirable due to packet reordering issues [22]. For the lower layer of such networks, in contrast, demands correspond to integer numbers of optical circuits. While each circuit may follow a different route, we cannot split one circuit up into flows of finer granularity. Still, we can apply almost the same formulation to this WDM layer problem. Assuming that $d_{sd} \in \mathbb{Z}_0^+$ gives the number of optical circuits, we only have to require the flow variables to be integer, i. e. $f_{ij}^{(sd)} \in \mathbb{Z}_0^+$, which prevents the solution from routing fractions of circuits on different paths. The problem therewith turns into an ILP.

Other operational assumptions may disallow the bifurcation of traffic flows in packet networks (with arbitrary demand granularity, i. e. $d_{sd} \in \mathbb{R}_0^+$). We can cover this case by redefining the flow variables as binary indicators whether the respective demand uses a link or not: $f_{ij}^{(sd)} \in \{0,1\}$. We therewith rewrite the capacitated total flow minimization problem, which becomes a BILP, as follows:

$$\textbf{minimize} \quad \sum_{(i,j)\in V\times V} \sum_{(s,d)\in K} f_{ij}^{(sd)} d_{sd} \tag{2.13}$$

$$\textbf{subject to} \quad \sum_{j\in V\setminus\{v\}} f_{vj}^{(sd)} - \sum_{i\in V\setminus\{v\}} f_{iv}^{(sd)} = \begin{cases} 1 & \text{if } v = s \\ -1 & \text{if } v = d \\ 0 & \text{otherwise} \end{cases} \quad \forall\,(s,d)\in K, v\in V \tag{2.14}$$

$$\sum_{(s,d)\in K} f_{ij}^{(sd)} d_{sd} \le B_{ij} \quad \forall\,(i,j)\in V\times V \tag{2.15}$$

Let us now come back to the initial assumption of arbitrarily splittable flows of real-valued volume and consider an *uncapacitated* design problem. Such problems, which determine how many resources (possibly of different types) shall be installed, typically serve for the initial dimensioning of a network. Extending our simple scenario, we assume that we can define link capacities as integer multiples of a basic capacity unit B. The concept of discrete link capacity steps is generally referred to as *link modularity* [18]. For our problem formulation, we need to introduce integer variables $a_{ij} \in \mathbb{Z}_0^+$ counting the number of capacity units per link $(i,j) \in E$. For ease of notation, we again define the variables for all $(i,j) \in V\times V$ and force the undesired ones to zero by Equation (2.19). As an exemplary objective, we want to minimize the gross number of installed capacity units, which corresponds to minimizing cost if we assume a flat price per capacity unit. This yields the following MILP formulation:

$$\textbf{minimize} \quad \sum_{(i,j)\in V\times V} a_{ij} \tag{2.16}$$

$$\textbf{subject to} \quad \sum_{j\in V\setminus\{v\}} f_{vj}^{(sd)} - \sum_{i\in V\setminus\{v\}} f_{iv}^{(sd)} = \begin{cases} d_{sd} & \text{if } v = s \\ -d_{sd} & \text{if } v = d \\ 0 & \text{otherwise} \end{cases} \quad \forall\,(s,d)\in K, v\in V \tag{2.17}$$

$$Ba_{ij} - \sum_{(s,d)\in K} f_{ij}^{(sd)} \ge 0 \quad \forall\,(i,j)\in V\times V \tag{2.18}$$

$$a_{ij} = 0 \quad \forall\,(i,j)\in V\times V\setminus E \tag{2.19}$$

For the definition of our reconfiguration problem in section 4.4.2, we use a more elaborate MCF formulation eliminating the variables for inexistent links. However, the size of the problem formulation is still growing fast with the size of the network scenario. For an approximate upper bound on the ratio of variables and constraints to the problem dimensions, let us assume a fully-meshed topology and a positive demand value between any node pair. This allows us to describe the problem size by the number of nodes. Neglecting terms of lower degree, we find the number of flow variables $f_{ij}^{(sd)}$ to scale[8] with $|V|^4$. The set of constraint equations is dominated by those for flow conservation, whose number increases with $|V|^3$. In order to keep problems computationally tractable for larger network scenarios, it is desirable to reduce these ratios. The aggregated link-node formulation presented in the following section does so by one order for some of the previously discussed problems without recurring to approximation.

2.3.1.2 Aggregated Link-Node Formulation

The benefit of the aggregated link-node formulation results from a reduced number of commodities. Instead of treating each node-to-node demand as a commodity, we only introduce one commodity for each node sourcing at least one non-zero demand[9] and subsume all traffic sourced at the node by this commodity. While we therewith loose information on the routing of individual node-to-node demands, we can be sure that such a routing exists. A relatively simple algorithm [23] allows re-establishing the individual routes by decomposing the aggregated commodity flows.

We thus re-define the set of commodities:

$$K = \{v \in V | \exists\, d \in V : d_{vd} > 0\} \tag{2.20}$$

The according flow variables per commodity and link $f_{ij}^{(k)} \in \mathbb{R}_0^+$ are conceptually related to those of the disaggregated formulation by

$$f_{ij}^{(k)} = \sum_{d \in V \setminus \{k\}} f_{ij}^{(kd)} \qquad \forall k \in K, (i,j) \in E \tag{2.21}$$

To simplify the formulation of the flow conservation constraints, we define the net demand value of commodity k *entering* the network at node v. I. e., these values are negative for all destination nodes where some of the commodity flow is terminated. For node k, the value corresponds to the positive sum of all demands sourced there:

$$d_v^{(k)} = \begin{cases} \sum_{d \in V} d_{vd} & \text{if } v = k \\ -d_{kv} & \text{if } v \neq k \end{cases} \qquad \forall\, k \in K, v \in V \tag{2.22}$$

[8] We derive the scaling behavior from the number of value combinations of the four index parameters.
[9] Analogously, we could base the formulation on the destination nodes, cf. [23].

Based on these definition, we rewrite the LP for the capacitated design problem according to Equation (2.10) to (2.12) as follows. The flow conservation constraint in Equation (2.24) now simply features the net demand value on the right-hand side.

$$\textbf{minimize} \qquad \sum_{(i,j)\in V\times V}\ \sum_{k\in K} f_{ij}^{(k)} \tag{2.23}$$

$$\textbf{subject to} \qquad \sum_{j\in V\setminus\{v\}} f_{vj}^{(k)} - \sum_{i\in V\setminus\{v\}} f_{iv}^{(k)} = d_v^{(k)} \qquad \forall\, v\in V, k\in K \tag{2.24}$$

$$\sum_{k\in K} f_{ij}^{(k)} \leq B_{ij} \qquad \forall\,(i,j)\in V\times V \tag{2.25}$$

Under the same worst-case scenario assumptions of a fully-meshed topology and positive demands between all nodes, we see that now the number of flow variables $f_{ij}^{(k)}$ scales with $|V|^3$ and that of flow conservation constraints with $|V|^2$. By accordingly rewriting the problem formulation, we can equally reduce these metrics for the capacitated ILP for the WDM layer and the MILP of the uncapacitated problem of Equation (2.16) to (2.19). In contrast, the non-bifurcated routing of demands of arbitrary volume does by principle not have an equivalent aggregated MCF formulation. Likewise, neither of the link-node formulations allows fine-grained control of the routing of demands, e. g. in terms of restrictions on the path length. If such are required, the link-path formulation outlined in the following section may be more suitable.

2.3.1.3 Link-Path Formulation

The link-path MCF formulation explicitly defines admissible paths for each commodity, which generally represents a node-to-node demand. We thus define the set of commodities as for the disaggregated link-node formulation in Equation (2.8):

$$K = \{(s,d)\in V\times V | d_{sd} > 0\} \tag{2.26}$$

For each commodity $(s,d)\in K$, we predefine the set of admissible paths P_{sd}[10]. Each candidate path $p\in P_{sd}$ represents an ordered list of links forming a path from node s to node d. Mimicking the formulation of [18], we can encode the assignment of links to paths by the *link-path incidence relation*

$$\delta_{ij,p}^{(sd)} = \begin{cases} 1 & \text{if link } (i,j) \text{ belongs to path } p\in P_{sd} \text{ for commodity } (s,d)\in K \\ 0 & \text{otherwise} \end{cases}$$
$$\forall\,(i,j)\in E, (s,d)\in K, p\in P_{sd} \tag{2.27}$$

The sets of predefined paths reduce the degrees of freedom in routing demands (or commodities) to deciding on the distribution of their volume among those paths. Accordingly, we only need one flow variable per demand and candidate path, i. e. in case of arbitrarily splittable demands:

$$f_p^{(sd)} \in \mathbb{R}_0^+ \qquad \forall\,(s,d)\in K, p\in P_{sd} \tag{2.28}$$

[10] To conform to common notations in literature, we define some symbols for the link-path MCF formulation in this section differently from the rest of this monograph. The respective symbols may not be given at all or they are given with a different description in the list of symbols (page xviii *et sqq.*).

Therewith, we can re-formulate the initial capacitated design problem as follows. While expressing link flows – and the total flow – in the objective appears somewhat more complicated, the flow conservation constraints reduce to ensuring that all demands are correctly distributed onto their candidate paths.

$$\textbf{minimize} \quad \sum_{(i,j)\in V\times V} \sum_{(s,d)\in K} \sum_{p\in P_{sd}} \delta_{ij,p}^{(sd)} f_p^{(sd)} \tag{2.29}$$

$$\textbf{subject to} \quad \sum_{p\in P_{sd}} f_p^{(sd)} = d_{sd} \qquad \forall\,(s,d)\in K \tag{2.30}$$

$$\sum_{(s,d)\in K} \sum_{p\in P_{sd}} \delta_{ij,p}^{(sd)} f_p^{(sd)} \le B_{ij} \qquad \forall\,(i,j)\in V\times V \tag{2.31}$$

All other problems discussed in section 2.3.1.1 are analogously expressible in link-path formulation. As previously argued, this is indicated if we need to convey particular constraints on the routing of demands. Regarding the size of the formulation in terms of the numbers of variables and constraints, the link-path variant is only beneficial if we significantly restrict the number of candidate paths, e. g. to a fixed small number per commodity [18, 24]. Otherwise, e. g. if generous path length bounds allow routing along the majority of source-to-destination paths in a fully meshed network, the number of flow variables easily exceeds that of the disaggregated link-node formulation.

2.3.2 Multi-Layer Networking Problems

Mathematical programming formulations of multi-layer network optimization problems are commonly composed of interrelated MCF formulations for the individual layer networks. Essentially, the capacity required on the elements (e. g. links) of an upper-layer network defines the demands for the next-lower layer network. In a single problem, we may apply different MCF formulation variants to different layers, cf. [21].

In this monograph, we adhere to an indexing convention for two-layer (e. g. IP/MPLS over WDM) networks introduced by Mukherjee (*et al.*) [8, 25], which is also applied in [26]:

- $(s,d) \in V\times V$ denotes the source and destination nodes of a packet-traffic demand of the upper layer network,
- $(i,j) \in V\times V$ stands for the start and end nodes of a link in the virtual topology of the upper layer, or equivalently for the respective termination points of optical circuits, and
- $(m,n) \in V\times V$ denotes node pairs connected by fiber links in the physical topology.

Using these definitions, we construct a simple IP/MPLS over WDM network design problem. It applies the uncapacitated single-layer MCF formulation of Equation (2.16) to (2.19) for the upper layer and the capacitated one of Equation (2.10) to (2.12) with integer variables (denoted by $\tilde{f}_{mn}^{(ij)} \in \mathbb{Z}_0^+$ for distinction) for the lower. We assume no restrictions on admissible virtual links and full wavelength conversion capability in the lower layer. The variables $a_{ij} \in \mathbb{Z}_0^+$ now denote the number of circuits on the virtual link (i, j). The set of commodities K shall comprise all non-zero node-to-node demands for the upper layer according to Equation (2.8). The network graph $G = \langle V, E \rangle$ shall represent the physical network with one fiber per directed physical link. Hence, a binary number of fibers encodes the physical topology:

$$n_{\mathrm{F}mn} = \begin{cases} 1 & \text{if } (m,n) \in E \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, (m,n) \in V \times V \tag{2.32}$$

Each fiber supports up to N_λ optical channels of capacity B each. Therewith, we get the following problem minimizing the number of optical circuits. This is e. g. desirable to minimize the cost of installing TXPs and line cards.

$$\textbf{minimize} \qquad \sum_{(i,j) \in V \times V} a_{ij} \tag{2.33}$$

$$\textbf{subject to} \quad \sum_{j \in V \setminus \{v\}} f_{vj}^{(sd)} - \sum_{i \in V \setminus \{v\}} f_{iv}^{(sd)} = \begin{cases} d_{sd} & \text{if } v = s \\ -d_{sd} & \text{if } v = d \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, (s,d) \in K, v \in V \tag{2.34}$$

$$B a_{ij} - \sum_{(s,d) \in K} f_{ij}^{(sd)} \geq 0 \qquad \forall\, (i,j) \in V \times V \tag{2.35}$$

$$\sum_{n \in V \setminus \{v\}} \tilde{f}_{vn}^{(ij)} - \sum_{m \in V \setminus \{v\}} \tilde{f}_{mv}^{(ij)} = \begin{cases} a_{ij} & \text{if } v = i \\ -a_{ij} & \text{if } v = j \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, (i,j) \in V \times V, v \in V \tag{2.36}$$

$$\sum_{(i,j) \in V \times V} \tilde{f}_{mn}^{(ij)} \leq N_\lambda n_{\mathrm{F}mn} \qquad \forall\, (m,n) \in V \times V \tag{2.37}$$

Equation (2.34) stipulates flow conservation and the accommodation of all demands in the potentially fully-meshed virtual topology. Equation (2.35) postulates a sufficient number of optical circuits on each virtual link to transport the resulting flow. The flow conservation constraint in Equation (2.36) governs the routing of the postulated circuits, while Equation (2.37) enforces fiber capacity constraints. Further examples of multi-layer MCF formulations are e. g. found in [8, 18].

2.4 Overview of Solution Methods

For certain classes of optimization problems, researchers have developed algorithms which determine optimal numerical solutions. Such mathematical optimization methods, which section 2.4.1 introduces, are able to guarantee the optimality of their solutions, or they may specify a lower bound for the objective function value along with a feasible sub-optimal solution if terminated preliminarily. On the downside, some of these methods are computationally too complex to be applied to problems of practically relevant size. This is particularly true for problems involving discrete variables.

For this reason, researchers and practitioners resorted to developing *heuristic methods* (or, in short, *heuristics*), which aim at providing good solutions in limited time. Such methods are generally not able to provide any guarantees on the quality of the solutions they return. On the upside, many of these methods do not require a stringent mathematical formulation of the problem, but they can operate on a more intuitive representation which may facilitate the implementation of the solution [17]. A first class of heuristics exploits specific properties of the problem in order to design one good feasible solution. That is, these methods only identify and evaluate one point in the solution space. Although the conception and the procedure to define this point may be highly sophisticated, we refer to the methods not contemplating alternative solutions as *simple heuristics*. We outline this approach in section 2.4.2.

Optimization meta-heuristics constitute a different class of heuristic methods. Instead of defining one solution, they center on guiding a non-exhaustive search process of the solution space. As the term *meta*-heuristic implies, these methods express the decision rules controlling the search process in a problem-agnostic way. To apply them to a specific problem, one therefore needs to define additional operations based on the properties and representation of the problem. Section 2.4.3 further classifies heuristic optimization methods.

2.4.1 Mathematical Optimization

Mathematical solution methods are able to provide optimal numerical solutions to instances of mathematical programming problems, i. e. they assure that the provided solution actually yields the minimal objective function value. However, they always only return one optimal solution, whereas many problem instances have several – or infinitely many – optimal solutions. Some of these methods, which often take considerable time to solve problem instances of practically relevant size, are likewise able to provide a feasible solution along with a lower bound for the minimally achievable objective function value if interrupted before the optimal solution is found. Researchers facing optimization problems in many application fields usually make use of the implementation of such methods in commercial or open-source solver software like CPLEX [27] or SCIP [28].

For LP problems, i. e. linear programs with continuous-valued variables, the *simplex* method initially developed by Dantzig [29] solves many practically relevant instances in linear time. While the worst-case time complexity of this algorithm is exponential, it is considered one of the fastest approaches and often preferred to alternative algorithms of polynomial worst-case complexity [18]. Simplex exploits the following properties of LPs: First, the linear constraints according to Equation (2.4) reduce the feasible solution space to a polytope[11]. Second, if this polytope has *extreme points*, or *vertices*[12], then at least one vertex belongs to the set of optimal solutions. Essentially, the algorithm starts at one vertex and iteratively moves to neighbor vertices with lower cost function values until the optimum is reached. A more comprehensive overview of the simplex method is found in [18]. Littger [17] describes the algorithm in more detail.

For problems involving discrete variables, no comparably efficient mathematical solution method is known. For combinatorial problems (i. e. problems having a finite number of feasible solutions), one could in principle resort to exhaustive search, i. e. completely enumerate the solu-

[11] A *(hyper)polyhedron* of a dimension higher then three is called a polytope.

[12] Colloquially speaking, one may think of the corner points of a polyhedron.

tions, compute their objective function values, and select the minimal one. However, the size of the solution set prohibits this approach for the majority of practical problems. Partial enumeration alleviates this problem by suitably restricting the search space.

The *branch and bound* method, which is also applicable to MILP problems, is the most prominent and relevant partial enumeration method. Its basic idea goes back to the work by Land and Doig [30]. The method is based on initial relaxation of all integer constraints, yielding an LP solvable e. g. by the simplex algorithm. It then selects one originally integer variable with non-integer assignment in the LP solution and creates two new LP instances which bound the considered variable either above by the next-lower integer to its assignment in the LP solution, or below by the next-higher integer. This is referred to as the *branch* operation of the algorithm. By proceeding in the same way with the new LP instances, the method creates a binary search tree with each node being an LP instance. The interesting property of this tree is that, due to the relaxation of (integer) constraints, the objective function value of each node represents a lower bound for the objective function values achievable in the subtree rooted at this node. The method exploits this property for partial enumeration: if a feasible solution with an objective function value lower than this bound is known, it is futile to investigate the respective subtree, which is thus excluded by the *bound* operation. A feasible solution is obtained at leaf nodes, where the LP solution assigns integer values to all integer variables of the non-relaxed problem. More details on the branch and bound algorithm are again provided in [17, 18].

The ability of the branch and bound method to exclude parts of the solution space and speed up the solution process decisively depends on the quality, i. e. the tightness, of the lower bounds computed for the search tree nodes. The *branch and cut* method therefore improves branch and bound in this regard: It complements the LP relaxations of the (M)ILP by so-called *valid inequalities*, which constrain the feasible (continuous) solution space more closely than the constraints of the original problem while including all integer solutions. Means to define such inequalities include Lagrangian relaxation, Bender's decomposition, and problem-specific methods. More details are discussed in [18] and references therein.

Some of the described methods and software tools make use of so-called *dual problems* to prove the optimality of a solution and to provide lower, so-called *dual* bounds for the objective function value. The dual problem to a *primal* – as the original problem is called in this context – minimization problem is a maximization problem, of which the optimal objective function value equals the minimal one of the primal problem. Accordingly, a feasible solution to the dual problem produces an objective function value which bounds the one of the primal solutions below. The transformation of the problems yields a variable in the dual problem for each constraint inequation of the primal problem. Duality is further discussed e. g. in [18].

2.4.2 Simple Heuristics

By *simple heuristics*, we designate methods which create one – presumably good – feasible solution by exploiting properties of the problem at hand. Such solution methods are thus necessarily specific for a certain problem or problem class. They may either be based on a solid theoretical foundation or on the experience or intuition of practitioners. Littger [17] names examples for the latter case, which is legitimate as a practical approach for small problems.

Unlike optimization methods, which explicitly make use of the objective function while analyzing the solution space, simple heuristics do not necessarily evaluate the objective function. For instance, one strategy to minimize the overall resource utilization in a single-layer network consists in routing all traffic flows on their shortest paths – regardless of the utilization along the chosen path. In contrast, a heuristic aiming at balanced link utilization may apply link weights proportional to the residual capacity, thereby locally evaluating terms of the objective function. The methodological survey in section 3.2.5 names further simple heuristics in the context of multi-layer networking. Domschke and Drexel [19] discuss simple heuristics of different degrees of sophistication in regard to creating feasible initial solutions for heuristic optimization methods.

2.4.3 Optimization Meta-Heuristics

Heuristic optimization methods have been conceived to guide a non-exhaustive search of the large but finite solution space of combinatorial optimization problems [18]. Some extensions also allow the application of such methods to problems involving continuous variables, e. g. in case of *evolutionary programming* [31]. Unlike the branch and bound method, optimization heuristics do not make use of objective function value bounds when restricting the visited part of the solution space. Accordingly, they may miss the optimal solution and do not provide any hard guarantee[13] on the quality of the solutions they find.

Many relevant optimization heuristics belong to the class of *local search* methods. After creating one or several feasible initial solutions, these methods search the solution space by iteratively creating new candidate solutions which bear similarity to the previously evaluated solutions. We refer to these similar solutions as *neighbor solutions*. Using mathematical programming notations, we formally describe the set of all neighbor solutions $N(\boldsymbol{x}) \subset S$ to solution $\boldsymbol{x} \in S$ by the *neighborhood function* $N : S \to 2^S$ [32, 33].

The way they select candidate solutions from the neighborhood $N(\boldsymbol{x})$ characterizes different heuristics. In particular, we can differentiate deterministic and stochastic methods. A further classification criterion, which also serves to structure the following presentation, is the number of candidate solutions evaluated during each iteration. In section 2.4.3.1, we describe methods dealing with a single solution at a time, whereas section 2.4.3.2 addresses algorithms using sets of candidate solutions.

2.4.3.1 Trajectory Methods

The class of trajectory methods subsumes a number of different meta-heuristics which search the solution space by iteratively modifying one candidate solution [33]. The term *trajectory* refers to the resulting search path in the solution space.

Iterative improvement methods [32, 34] take an intuitive, deterministic approach to identify solutions better than the feasible initial solution: They evaluate the neighborhood of this solution,

[13] Asymptotic convergence can be proven for some methods, e. g. simulated annealing, however assuming impractically many search steps.

move on to one neighbor solution showing a better objective function value, and continue the search in the neighborhood of this solution. Depending on the way this neighborhood is evaluated, we differentiate between *iterative first improvement* and *iterative best improvement*. The former method, which is also referred to as *iterative descend* or *hill climbing* for minimization or maximization problems, respectively, chooses the first neighbor solution it encounters with better objective function value. The latter, in contrast, performs an exhaustive search of the neighborhood and moves to one solution yielding the maximum improvement of the objective. Both methods terminate if the neighborhood contains no solution with a better objective function value.

By principle, iterative improvement terminates after reaching a local optimum contained in the successively evaluated neighborhoods. While the structure of some problems may ensure that this is likewise the global optimum, no statement on the quality of the found solution is possible in the general case. One can mitigate this issue by enlarging the considered neighborhood. However, this comes at the price of the computational effort to evaluate more solutions in each iteration, in particular for the iterative best improvement scheme. Still, the methods will not search beyond a local optimum.

Greedy randomized adaptive search procedures (GRASP) constitute one approach to improve the coverage of the solution space. Such methods perform iterative (best) improvement for several different initial solutions, which are constructed by randomly choosing components from a set of candidates. This increases the chance of exploring several local optima, which are in turn more likely to include a globally optimal solution or at least a solution of an objective function value close to the optimum. Further details are found in [34] and references therein.

Tabu search is a deterministic method augmenting iterative best improvement by means to escape a local optimum. In its basic form [18], it makes use of a memory structure called *tabu list* which stores characteristics of the modifications of the solution during recent iterations. When evaluating the next neighborhood, it excludes candidates differing from the current solution in a way contained in the tabu list. In addition, the method does not terminate if not finding a better solution, but selects the one incurring the least deterioration of the objective function value in this case. By these means, it enables the search process to move away from a local optimum, since the tabu list prevents cycling in its neighborhood. Glover *et al.* [35] discuss this method and its extensions in more detail.

Simulated annealing [18, 36] performs a stochastic search of the solution space similar to the iterative first improvement method. In each iteration, it randomly selects one neighbor solution and moves to it deterministically if it improves the objective function value. Otherwise, the new solution is accepted with a probability that decreases with the amplitude of the deterioration of the objective. If the solution is rejected, another candidate is randomly chosen from the neighborhood of the previously accepted solution. Controlled by a parameter called *temperature*, the probability of accepting worse solutions decreases over time. Hence, the method initially facilitates the exploration of different parts of the solution space and finally focuses on refining the solution towards a (local) optimum. Section 2.5 introduces the simulated annealing method in more detail.

2.4.3.2 Evolutionary Algorithms

Evolutionary algorithms subsume a number of methods inspired by the Darwinian principles of natural selection [31]. Mimicking the evolution of biological species, these methods operate on a *population* of *individuals*, i. e. a set whose elements directly or indirectly represent solutions to the problem under consideration. They iteratively apply *reproduction operators* to a subset of individuals determined by *selection* depending on their *fitness*, which is a quality metric e. g. based on objective function values. Depending on the number of individuals they are applied to, the reproduction operators are referred to as *recombination* or *crossover* if N-ary and *mutation* or *modification* if unary. Finally, a *replacement* step performs another selection to determine which ones of the old and the newly created individuals remain in the population to form the next generation.

The field of evolutionary algorithms comprises three traditional variants, which have originally been developed independently, and a number of combinations and enhancements thereof. The traditional types are *evolutionary programming*, *evolutionary strategies*, and *genetic algorithms*. The former two are mainly applied to problems with continuous-valued variables and they primarily rely on unary mutation operators. While the first one directly operates on any problem-specifically modeled individuals, the second variant assumes arrays of floating-point numbers and uses adaptive mutation strategies.

In contrast, genetic algorithms, the most widely-used variant of evolutionary algorithms, solves combinatorial optimization problems. It represents solutions by so-called *chromosomes*, which are strings of *genes* which assume one of several values referred to as *alleles*. Traditionally, alleles are binary values, though other options are possible. The set of all possible chromosomes forms a *genotype space*, whose elements are mapped to (problem-specifically described) individuals in the *phenotype space*. The phenotypes generally serve for fitness evaluation. Typically, the predominant reproduction operator is crossover, for which a number of strategies to exchange parts of the genome of two parents exist. A detailed introduction to the operators and application of genetic algorithms is given in [37].

Crossover and mutation operations create one or several new individuals based on the genome of their input in a stochastic way. Conceptually, the unary mutation operator randomly selects one individual from a neighborhood of the considered (input) individual. This resembles the permutation in simulated annealing. Regarding recombination, the genome of the parent individuals and the rules controlling the applied crossover operator likewise define a neighborhood, i. e. a set of all possible offspring individuals from which the actually generated individuals are stochastically chosen. The size of this neighborhood depends on the similarity of the genome of the parents.

2.5 Simulated Annealing

The optimization meta-heuristic of simulated annealing (SA) has been invented independently by Kirkpatrick *et al.* [38] and Černý [39]. It is inspired by the slow annealing process in condensed-matter physics, which aims at producing solids of particles forming a highly-structured lattice corresponding to a low-energy ground state. Therefore, the method is related to the

Metropolis algorithm, which simulates the evolution of a solid to thermal equilibrium at a constant temperature [40]. To address combinatorial optimization problems, the meta-heuristic builds on a correspondence of problem solutions to states of the physical system and equates the objective function value with the energy of the state[14]. Aarts *et al.* [36] further detail the physical analogon.

As introduced in section 2.4.3, SA is a stochastic local search method iteratively modifying one solution. In order to apply it to a specific problem, one needs to decide upon the representation of the solutions, devise a procedure to create a feasible initial solution, design the *perturbation* operation which randomly provides one neighbor solution, and define the objective function, or *cost function*, which evaluates the solutions. Based on these elements, the meta-heuristic guides the search process by deciding probabilistically based on cost values whether to accept or reject a solution obtained by perturbation. It adjusts this probabilistic decision-making by means of a control parameter referred to as *temperature*, which is decreased over time. In addition, the meta-heuristic determines the number of iterations in-between these decrements as well as the stopping criterion.

In the following, we address these aspects in more detail. Section 2.5.1 describes the meta-heuristic search procedure, and section 2.5.2 further discusses the parameters guiding this search. We finally comment on the problem-specific elements in section 2.5.3.

2.5.1 Meta-Heuristic Procedure

Algorithm 2.1 describes the principle of the meta-heuristic optimization procedure of SA. It uses the symbols $\boldsymbol{x}$ and $F(\boldsymbol{x})$ introduced in section 2.2 for a feasible solution and the objective function, respectively. We start by initializing the *accepted* solution $\boldsymbol{x}$ and compute its cost (i. e. objective function value) c. Before entering the iterative search process, we likewise set the temperature parameter ϑ controlling the stochastic search to its initial (maximum) value of ϑ_0.

In its basic form, the algorithm performs a fixed number of $\nu_{\max}$ iterations for each value of the temperature parameter ϑ. The iteration starts with deriving a *candidate* solution $\boldsymbol{x}'$ from the accepted solution $\boldsymbol{x}$ by perturbation and the computation of the cost c' of the candidate solution. We deterministically accept the candidate solution, i. e. we set it as the accepted solution, if its cost c' is not higher than that of the accepted solution. Candidates increasing the cost are stochastically accepted with a probability p_{acc} depending on the cost difference:

$$p_{\text{acc}} = \mathrm{e}^{-\frac{c'-c}{\vartheta}} \tag{2.38}$$

This acceptance rule is a generalization of the so-called *Metropolis criterion* [36] for state transitions in solids. For large values of the temperature parameter, which are used at the beginning of the optimization, worse solutions are accepted with a high probability. This probability decreases with the temperature in the course of the procedure. In the extreme case of $\vartheta = 0$, the

[14]This stipulation assumes a minimization problem.

Algorithm 2.1 Principle simulated annealing algorithm

$\boldsymbol{x} \leftarrow$ CREATEINITIALSOLUTION()
$c \leftarrow F(\boldsymbol{x})$
$\vartheta \leftarrow \vartheta_0$ {initialization of temperature parameter}
repeat
  for $\nu \leftarrow 1$ **to** $\nu_{\max}$ **do** {*temperature-length* iterations}
    $\boldsymbol{x}' \leftarrow$ PERTURBATE($\boldsymbol{x}$)
    $c' \leftarrow F(\boldsymbol{x}')$
    $r \leftarrow$ random number in $[0,1)$
    if $c' \leq c$ **or** $r < \exp\left(-\frac{c'-c}{\vartheta}\right)$ **then** {accept candidate solution}
      $\boldsymbol{x} \leftarrow \boldsymbol{x}'$
      $c \leftarrow c'$
    end if
  end for
  $\vartheta \leftarrow \vartheta \cdot \gamma$ {annealing}
until $\vartheta < \vartheta_{\min}$ {stopping criterion}
return $\boldsymbol{x}$

method turns into pure iterative improvement. For a given temperature, the acceptance probability rapidly decreases with increasing cost difference, but it remains positive even for significantly worse solutions. These features allow the search to explore promising neighborhoods of solutions while being able to escape from local optima.

After $\nu_{\max}$ iterations, a value also referred to as *temperature length*, we perform a cooling step and reduce the temperature ϑ. This is often done according to a geometric decrement function, i. e. multiplying the temperature by a positive cooling factor $\gamma < 1$, cf. e. g. [18, 36]. In principle, alternative decrement functions are applicable, e. g. reducing the temperature by a constant value.

The search procedure terminates upon meeting a stopping criterion which, in the simple case, is a lower limit for the temperature parameter. It then returns the solution accepted at this moment. Since this finally accepted solution is not necessarily the best one encountered during the search process, many implementations keep track of the best encountered solution and finally return this solution instead.

2.5.2 Cooling Schedule

The values and laws for the parameters guiding the stochastic search process define the so-called *cooling schedule*. A static cooling schedule as outlined in section 2.5.1 includes [36]:

- the initial value of the temperature ϑ_0,
- the decrement function reducing the temperature,
- the stopping criterion in terms of the minimum temperature value $\vartheta_{\min}$, and

- the number of iterations $\nu_{\max}$ for each temperature value.

Some rules of thumb for the calibration of this cooling schedule exist [36]: In order to assure a sufficient initial acceptance ratio, the initial temperature should range in the order of the maximum difference in cost between any two neighbor solutions. Similarly, the minimum temperature for the stopping criterion may be based on the minimum cost difference between such solutions. Typical cooling factors are $\gamma \in [0.80, 0.99]$. The number of iterations $\nu_{\max}$ per temperature level, generally a large value, may be related to the size of the neighborhood.

In practice, it may be difficult to estimate the properties of a specific problem instance required to define these parameters of the static cooling schedule. Therefore, a number of different extensions have been proposed which result in dynamic cooling schedules. Aarts *et al.* [36] outline some of these extensions. For instance, one auto-tuning approach initially increases the temperature until a desired proportion, typically 90 % to 99 %, of candidate solutions is accepted. The normal annealing process is then started from this temperature. Another proposal introduces an adaptive temperature length by decreasing the temperature not only after a fixed number of iterations but already after accepting $\alpha_{\max}$ candidate solutions. The decision to terminate the computation may likewise be taken dynamically after a certain number of iterations without any improvement of the cost of the best encountered solution. In scenarios where SA is applied for on-line optimization and a solution is required at a given moment in time, the computation can alternatively be terminated simply upon expiry of the available time budget.

The literature discusses a number of more sophisticated dynamic cooling schedules which are generally based on the notion of reaching a statistical equilibrium at each temperature level [36]. One such example is the contribution by Huang *et al.* [41], which aims at auto-tuning all elements of the cooling schedule based on statistics of the cost variations produced by perturbations. As a result, the scheme shall be applicable for any combinatorial optimization problem. However, the study reported in [42] hints that for a specific problem, a simpler dynamic cooling schedule may perform at least as good as the elaborate auto-tuning scheme.

2.5.3 Problem-Specific Elements

We conclude the discussion of the SA method with a few comments on requirements and properties of the problem-specific elements used by the meta-heuristic procedure. Unlike mathematical programming or genetic algorithms, SA does not impose any constraints on the representation of a solution. We may thus use any structure deemed suitable for the respective problem, provided that it supports the definition of a permutation operator and allows cost computation. Due to the large number of invocations of these two operations during the iterative optimization procedure, a computationally efficient implementation is highly beneficial for the execution time of the heuristic.

In the physical analogon, a permutation corresponds to a small distortion like the displacement of a single particle [36]. Accordingly, it is advisable to restrict the permutation operation to simple modifications with limited scope. This principle proved to yield good performance in the work cumulating in [43].

As for any optimization heuristic, a feasible initial solution may be created randomly or by a simple heuristic already aiming at a good solution. The latter is particularly beneficial if limited computation time restricts the subset of the solution space the heuristic search can cover. In this case, we can focus the stochastic search to the promising neighborhood of the initial solution by setting a relatively low initial temperature. In contrast, the initial solution should not significantly impact the result for a sufficiently long cooling schedule dimensioned according to the stipulations of section 2.5.2.

3 Multi-Layer and Energy-Aware Network Design and Reconfiguration

This chapter reviews the state of the art in two research fields relevant to the problem and solution methods investigated in this monograph. The first one is the reconfiguration of multi-layer networks. Since reconfiguration essentially means finding a new network configuration for changed conditions, it is closely related to the initial problem of network design. We accordingly include publications on multi-layer network design. The second field concerns energy savings in communication networks, commonly referred to as *energy-aware networking* or *green networking*.

The first section provides an overview of publications from both areas which cover studies on or methods for network design or reconfiguration. It classifies these contributions in regard to several criteria, ranging from the assumed networking scenario over the exploited degrees of freedom and the objective to the applied solution method. In section 3.2, we systematically discuss different aspects of single-layer and multi-layer network design and reconfiguration, which underlie the aforementioned classification. Like the first overview section, this section refers to publications on both conventional and green networking. In section 3.3, we finally introduce energy-aware networking research in a broader context and detail work in two areas specifically relevant to this monograph: the design of power consumption models for network elements and the investigation of schemes for load-dependent resource operation.

3.1 Overview of Related Research

The conception and investigation of network design and reconfiguration problems and methods lends itself to a number of classification criteria which are either independent or only loosely related. Before discussing these criteria individually in section 3.2, we provide a tabular overview of the referenced publications in this section, which highlights the diversity of combinations of criterion values found in the literature.

Table 3.1 summarizes publications on network design and reconfiguration targeting conventional objectives like minimizing installation cost or resource utilization. We present respective contributions from the green networking field, i. e. where the objective is minimizing energy consumption, in Table 3.2. In the following, we shortly describe the classification criteria presented in the columns of the tables, which are further detailed in section 3.2. We base our classification on the actual properties of the investigated scenarios, problems, and methods, which may diverge from the claims by the respective authors. For publications presenting several options, e. g. for the objective or the solution method, we enumerate the alternatives in the respective columns.

The column *problem* distinguishes five major increasingly restrictive *problem types*:

- *Uncapacitated design* problems are characterized by the absence of resource constraints. Usually, such problems aim at determining the amount and type of resources to be installed in order to satisfy all demands while minimizing the monetary or energetic cost. By *(un)capacitated design*, we designate publications which first describe an uncapacitated problem used to determine a dimensioning of resources, which they in turn apply for the evaluation of a capacitated design method.

- *Capacitated design* problems aim at finding a network configuration satisfying a certain set of demands under given resource constraints. Traffic blocking may not be avoidable in all cases, but most proposals devise some implicit or explicit means to minimize its extent. The problems additionally target explicit objectives, e. g. balancing resource utilization or minimizing the number of active resources. This category also comprises proposals for network reconfiguration which determine an independent configuration for each period, since the problem providing each configuration essentially is a capacitated design problem.

- *Reconfiguration* problems are similar to capacitated design problems, but additionally feature some dependency on a previous configuration. This covers a wide range of mechanisms, from constraints or objective terms limiting the extent of configuration changes to approaches only proposing a single modification.

- *Dynamic grooming* problems constitute a particularly constrained form of reconfiguration problems. They aim at accommodating a single additional demand at a time without any disruptive changes of the configuration like rerouting of traffic of preexisting demands or modification of non-idle resources.

- *Capacitated* or *uncapacitated multi-hour design* problems fall in-between design and reconfiguration problems. They assume complete knowledge of future traffic demands and design one or several sequential configurations satisfying these demands while optimizing some additional objective. Usually, such methods exploit their complete knowledge of demands to restrict the (overall) extent of required configuration changes.

The *scenario* column distinguishes single-layer and multi-layer (mostly two-layer) networks, gives the assumed technologies in parentheses if applicable, and it may provide some additional information. The indicated number of layers reflects the degrees of freedom in the problem: we designate a method targeting only one layer network in a multi-layer scenario as *single-layer*.

The *traffic model* indicates the format in which demands are specified. We mostly find traffic matrices with continuous aggregate demand values or matrices indicating numbers of demands of specific granularities. Sequences of such matrices enable reconfiguration studies. By *set*s of demands, we designate more detailed models giving several demands of varying granularity for each pair of nodes. For publications on dynamic grooming, we indicate the assumed statistical traffic model.

The column *split* specifies whether the respective problem allows the bifurcation, or splitting, of individual demands. Thereby, one demand is defined as indicated in the traffic model column, i. e. it may be the aggregated traffic between a pair of nodes, one of several demands of discrete volume between such a pair, or the statistically arriving traffic flow in case of dynamic grooming.

In the *objective* column, we give the quantifiable objective the authors claim to pursue. If they determine the complete network configuration by exact or heuristic optimization methods, this coincides with the objective function. Heuristic methods may use different decision criteria, which are however supposed to yield a good solution in terms of the objective.

In the second case, the *approach* column outlines the primary decision criterion or the heuristically pursued properties. By *direct optimization*, we denote the application of optimization methods explicitly targeting the *objective* given in the correspondent column. Incremental reconfiguration methods execute the procedures sketched in this column upon each invocation.

The column *degrees of freedom* categorizes the elements of a network configuration which the problems determine or modify, respectively. While the networking *scenario* may restrict the available options, this column gives the degrees of freedom actually exploited by the problem. We consider the following aspects:

- For the *routing of demands*, i. e. the routing of (packet) traffic in the upper layer in case of a multi-layer scenario, we distinguish whether *all* demands may be freely routed or rerouted, whether routing is only done for *new*ly arriving demands in case of dynamic grooming, or whether demands being *affected* by a different configuration change are rerouted.

- In a multi-layer scenario, *virtual topology adaptation* refers to the possibility to add and remove optical circuits independently of an initial configuration. *Virtual topology design* denotes the analogous degree of freedom in defining one static configuration. In both cases, we specify whether the realization of new circuits by RWA or only their routing (e. g. when assuming full wavelength conversion capability) is considered.

- The pure *deactivation* of elements, e. g. links, circuits, or network node components, in an otherwise pre-defined network configuration is frequently applied in green networking. While this likewise results in a modification of the virtual topology in case of multi-layer networks, we reserve the term *virtual topology adaptation* for the above-mentioned more general scheme.

- Green networking may additionally feature means for more fine-grained adaptation of power consumption to load. This includes *power scaling* in nodes and *link rate adaptation*.

- Finally, uncapacitated design problems generally allow for *resource dimensioning*.

The column *centr./dist.* specifies whether the proposed solution method works in a centralized or distributed way.

The *method* column indicates the class of the applied solution method. In case of simple problem-specific heuristics and particular combinations of methods, we give some additional details on the ideas governing the respective approach.

For publications targeting some form of adaptation of the network configuration over time, the column *trigger* gives the type of event upon which the respective methods are invoked. For time-driven schemes, we additionally indicate (the order of magnitude of) the inter-reconfiguration times.

Table 3.1: Publications on conventional network (re)configuration

ref.	problem	scenario	traffic model	split	objective	approach	degrees of freedom	centr./dist.	method	trigger
[24]	uncapacitated design	single-layer network (WDM)	one static matrix of lightpath demands	no	min total fiber length or number of fibers	direct optimization	resource dimensioning (regarding fibers on physical links); RWA	centralized	ILP	n/a
[44]	uncapacitated design	single-layer network (opaque SDH over WDM, with optional SDH links w/o WDM)	one set of demands of granularity equaling the WDM channel capacity	no	min CAPEX regarding per-connection equipment (line cards etc.), OXCs, and WDM line systems	direct optimization, initial shortest path routing with rerouting on least-cost path	resource dimensioning (on nodes and links); routing of all demands	centralized	MILP, simple heuristic, GRASP	n/a
[45]	uncapacitated design	two-layer network (IP/MPLS WDM)	one static matrix of (aggregate) demands	no	min. CAPEX for fiber installation, WDM line systems, OXCs, routers and line cards with TXPs	cost-based addition and removal of resources while routing demands by LP	resource dimensioning; virtual topology design; routing of all demands	centralized	iterative solution refinement by applying forward synthesis and design tightening heuristics to both layers	n/a
[46]	capacitated design	two-layer network (IP/MPLS over WDM)	one set of demands of continuous granularity	no	min. congestion regarding physical links and router ports	direct optimization; demand routing leaving least spare capacity, circuit routing on least congested path, and rerouting from lowly-loaded circuits	virtual topology design w/ routing; routing of all demands	centralized	ILP, simple multi-layer routing heuristic with local solution improvement	n/a
[25]	capacitated design	two-layer network (IP/MPLS or OTH (s-TDM) over WDM)	one set of demands of several discrete granularities	no	max. total throughput	direct optimization; max. single-hop traffic or max. resource utilization	virtual topology design w/ RWA; routing of all demands	centralized	ILP, simple multi-layer routing heuristics	n/a
[47]	capacitated design	two-layer network (*TDM over WDM) w/ different grooming capabilities	one static matrix of (aggregate) demands	no	min. electronic layer resources and processing and wavelength conversion	shortest-path routing in auxiliary graph with higher link weights for electrical processing and wavelength conversion	virtual topology design w/ RWA; routing of all demands	centralized	ILP formulation, simple multi-layer routing heuristics, randomized gradual path definition heuristic, optionally with local improvement heuristic	n/a

ref.	problem	scenario	traffic model	split	objective	approach	degrees of freedom	centr./dist.	method	trigger
[48]	capacitated design	two-layer network (packet over WDM)	one static matrix of (aggregate) demands	yes	min. congestion on virtual links	direct optimization; virtual topology definition based on physical node distance, (largest) traffic demands, or solution of LP relaxation	virtual topology design w/ RWA; routing of all demands	centralized	MILP, simple virtual topology design heuristics with demand routing by LP	n/a
[49]	capacitated design and reconfiguration	two-layer network (packet over WDM)	a single matrix resp. a sequence of matrices of (aggregate) demands	yes	min. packet hops and circuit changes for reconfiguration	direct optimization (sequentially for the objectives in case of reconfiguration), max. single-hop traffic or multi-hop traffic	virtual topology design/adaptation w/ routing; (re)routing of all demands	centralized	MILP (sequential for reconfiguration), simple multi-layer routing heuristic	n/a resp. not specified
[50]	uncapacitated multi-hour design	two-layer network (ATM over WDM)	implicitly: a sequence of matrices of (aggregate) demands	not discussed	hitless transition between a given sequence of virtual topologies	resource dimensioning for given configurations, heuristic wavelength assignment	resource dimensioning; wavelength assignment (disregarding routing)	centralized	simple dimensioning rule, simple graph-coloring based wavelength assignment heuristic	not discussed
[51]	uncapacitated multi-hour design	two-layer network (? over WDM)	a periodic series of matrices of (aggregate) demands	yes	min. (cost of) TXPs	direct optimization; elimination of temporally rarely used TXPs by tabu search	virtual topology design (static or per period); routing of all demands in each period	centralized	MILP, tabu search applying MILPs for single time periods affected by resource modification	design for several periods
[11]	capacitated multi-hour design	two-layer network (IP/MPLS over WDM)	a sequence of matrices of (aggregate) demands	yes	min. electronic processing, congestion on links, circuits, or linear combination thereof	direct optimization	design of static virtual topology; routing of all demands (static or per period)	centralized	MILP	design for several periods
[52]	capacitated multi-hour design	two-layer network (IP/MPLS over WDM)	sets/matrices of demands of continuous granularity for several periods per day (in evaluation, derived from population and time zone)	no	max. total carried traffic	direct optimization	design of static virtual topology w/ RWA; routing of all demands in each period	centralized	MILP, tabu search virtual topology optimization with simple demand routing heuristic	design for several periods (of 1h or 4h)

ref.	problem	scenario	traffic model	split	objective	approach	degrees of freedom	centr./dist.	method	trigger
[53]	reconfiguration	two-layer network (connection-oriented packet over broadcast WDM)	single on-off connections between node pairs with negative-exponentially distributed holding and return times	no	min. weighted sum of ratio of packets lost due to reconfiguration and packet hops	selection of virtual topology template based on current configuration and connection state	selection of virtual topology template; (re)routing of all demands	centralized	design of template selection strategy by complete enumeration and heuristics to restrict the considered state space	connection state change
[54]	reconfiguration	two-layer networks (IP over WDM)	continuous evolution of aggregate demand values	no	min. circuits and congestion on them	identification of one circuit to add or remove, by optimizing congestion or heuristically based on demands, connectivity, and link load	virtual topology adaptation w/ routing; (re)routing of all demands	centralized	MILP, simple circuit modification and RWA heuristics with shortest-path demand routing	link-load thresholds
[55]	reconfiguration	two-layer network (packet over WDM)	two consecutive matrices of (aggregate) demands	optional	min. circuits, fiber link usage, and packet hops in different linear combinations	direct optimization	virtual topology adaptation w/ routing; (re)routing of all demands	centralized	MILP	not discussed
[26]	reconfiguration	two-layer network (IP over WDM)	a sequence of matrices of (aggregate) demands (based on measurements)	yes	min. weighted sum of active circuits, new circuits, and total flow	direct optimization	virtual topology adaptation w/ RWA; (re)routing of all demands	centralized	MILP, optionally with limited set of modifiable circuits and precomputed RWA	link-load thresholds
[56]	reconfiguration	two-layer network (IP over WDM)	a sequence of matrices of (aggregate) demands (based on measurements)	no	min. weighted sum of active circuits and new circuits	heuristic direct optimization	virtual topology adaptation w/ RWA; (re)routing of all demands	distributed	distributed optimization of virtual topology by Lagrangian relaxation of MILP with shortest-path demand routing and simple RWA heuristic	link-load thresholds

ref.	problem	scenario	traffic model	split	objective	approach	degrees of freedom	centr./dist.	method	trigger
[57]	hybrid dynamic grooming concept	two-layer network (IP/MPLS over WDM)	arrival, departure, and bandwidth modification of demands of different priority; prediction of expected demands	?	min. resource usage by min. circuits and logical hops while filling circuits and limiting fiber utilization	constraint-based multi-layer routing of new demands and bandwidth modification with rerouting/preemption of lower-priority demands; occasional complete reconfiguration	virtual topology adaptation; routing of new demands; rerouting and preemption of demands; occasionally rerouting of all demands	both possible	none	demand arrival or bandwidth modification
[58]	dynamic grooming	optical two-layer network; wavelength drop&continue capability in nodes	Poisson demand arrivals and departures	no	min. demand blocking prob.	min. hops or length of logical path or length of physical path	virtual topology adaptation w/ RWA; routing of new demands	centralized	auxiliary graph heuristics	demand arrival
[59]	dynamic grooming	two-layer network (OTN over WDM) w/ different grooming capabilities	Poisson arrival and departure of demands of four different granularities	no	min. traffic blocking prob.	min. hops of logical or physical path, or min. circuits or wavelength-links, or adaptive utilization-dependent selection thereof	virtual topology adaptation w/ RWA; routing of new demands	centralized	auxiliary graph heuristics	demand arrival
[60]	dynamic grooming	two-layer network (SDH or IP over WDM) where OXCs can add/drop complete wavelengths	Poisson arrival and departure of demands of four different granularities	no	min. traffic blocking prob.	shortest-path routing in virtual and (if failed) in physical topology, or in auxiliary (reachability) graph	virtual topology adaptation w/ RWA; routing of new demands	supposed to be distributed	simple routing algorithm, auxiliary graph heuristic	demand arrival
[61]	dynamic grooming	two-layer network (SONET over WDM)	Poisson arrival and departure of demands of three different granularities	no	min. demand blocking prob.	selection of routing option based on route length metric covering both layers	virtual topology adaptation w/ RWA; routing of new demands	centralized	exhaustive evaluation of routing options with up to a given number of logical hops	demand arrival
[62]	dynamic grooming	two-layer network (IP/MPLS over WDM)	sequential arrival of demands of continuous random granularity; method also supports demand departure	no	min. demand blocking prob.	max. residual capacity (in current virtual topology) between all other node pairs	virtual topology adaptation w/ RWA; routing of new demands	centralized	auxiliary graph algorithm	demand arrival

Table 3.2: Publications on green network (re)configuration

ref.	problem	scenario	traffic model	split	objective	approach	degrees of freedom	centr./dist.	method	trigger
[63]	uncapacitated design	two-layer network (? over WDM)	one static matrix of (aggregate) demands, granularity being multiple of a base data rate	yes	min. circuits, electronically switched transit traffic, power consumption as linear combination thereof	direct optimization	virtual topology design w/ RWA; routing of all demands	centralized	ILP	n/a
[64]	uncapacitated design (but with upper bounds)	two-layer network (opaque IP over WDM)	one static matrix of (aggregate) demands	yes	min. power consumption by IP router ports, TXPs, and EDFAs	direct optimization, max. single-hop traffic or max. circuit utilization	virtual topology design w/ routing; routing of all demands	centralized	MILP, simple multi-layer routing heuristics	n/a
[65]	(un)capacitated design	single-layer network (IP)	single matrices of (aggregate) demands (randomly generated)	yes	min. power consumption by router chassis and line cards	direct optimization	resource dimensioning (design); deactivation of router chassis and line cards; (re)routing of all demands	centralized	MILP	not discussed
[10]	(un)capacitated design	two-layer network (IP over WDM) w/ optional flex-rate TXPs (25,50,75,100 Gbps)	static sets of bi-directional demands (randomly generated w/ or w/o geographic correlation), scaled by day profile for reconfiguration	yes	min. power consumption by all installed/active components of both layers	direct optimization	resource dimensioning (design); deactivation/rate adaptation of circuits along with terminating equipment and regenerators; (re)routing of all demands	centralized	MILP	traffic variation by 5 % of peak
[21]	(un)capacitated design	two-layer network (IP over WDM)	sequences of matrices of (aggregate) demands (based on measurements)	yes	min. power consumption by installed/active line cards	direct optimization	resource dimensioning (design); virtual topology design w/ routing, optionally adaptation or link deactivation; (re)routing of all demands (optionally)	centralized	MILP	periodic (15 min)

ref.	problem	scenario	traffic model	split	objective	approach	degrees of freedom	centr./dist.	method	trigger
[66]	capac-itated design	single-layer net-work (WDM)	one static matrix of lightpath demands	no	min power con-sumption by fiber amplifiers and OXCs (incurred as soon as one used by one circuit)	direct optimiza-tion; routing on least-cost/most-used path, de-mands optionally in sequence of least incremental cost	RWA w/ fiber selection	central-ized	ILP, simple rout-ing heuristics with simple fiber and wavelength selection strategy	n/a
[67]	capac-itated design	single-layer prob-lem (topmost layer of IP over SDH over WDM)	not discussed	yes	min. (energetic) cost of active router ports	demand routing with (sequential) activation of required / least expensive links or link deactivation with rerouting of affected traffic	deactivation of router ports; routing of all demands	central-ized	MILP formulation, simple single-layer routing resp. link deactivation heuristics	n/a
[68]	capac-itated design	single-layer net-work	one static matrix of (aggregate) demands	yes	min. active cables in bundled links	iterative removal of cables while feasible routing exists, selecting next candidate by utilization or resulting increase of total flow	deactivation of cables; (re)routing of all demands	central-ized	MILP formulation; simple and local exhaustive search heuristics for cable deactivation with demand routing by LP	e. g. periodic
[69]	capac-itated design	single-layer net-work (IP/MPLS)	sequences of matrices of (ag-gregate) demands (partly based on measurements)	yes	min. power con-sumption by ports and line cards, min. congestion for minimal power consumption	direct optimization (sequentially for the objectives)	deactivation of ports and line cards; (re)routing of all demands	central-ized	MILP and LP (sequentially)	periodic (5-15 min)
[70]	capac-itated design	single-layer net-work (IP)	single resp. a se-quence of matrices of (aggregate) demands (the lat-ter obtained by sinusoidal scaling)	yes (MI-LP) / no	min. power con-sumption by active links and nodes	direct optimiza-tion; sequential deactivation of elements, iteration sequence random or depending on topology or load	deactivation of nodes and links; (re)routing of all demands	central-ized	MILP, simple heuristics trying to deactivate nodes and then links if feasible with shortest-path rerouting under given constraints	not dis-cussed

ref.	problem	scenario	traffic model	split	objective	approach	degrees of freedom	centr./dist.	method	trigger
[71]	capacitated design	single-layer network (IP)	single resp. a sequence of matrices of (aggregate) demands (the latter obtained by scaling one matrix)	yes (MILP) / no	min. power consumption by active links and nodes	direct optimization; sequential deactivation of elements, iteration sequence random or depending on topology, power consumption, or load	deactivation of nodes and links; (re)routing of all demands	centralized	MILP, simple heuristics trying to deactivate nodes and then links if feasible with capacity-weighted shortest-path rerouting under given constraints	not discussed
[72]	capacitated design	single-layer network (IP)	a sequence of matrices of (aggregate) demands (based on measurements)	yes	min. power consumption by active links and nodes obeying energy profiles	direct optimization	deactivation of nodes and links; (re)routing of all demands	centralized	MILP	periodic (1h)
[73]	capacitated design	single-layer network (IP)	single matrices of (aggregate) demands	yes	min. power consumption by active line cards (with rate adaptation) and routers	per iteration, reroute least amount of traffic to move one network element to a lower-power state	deactivation of nodes and links; link rate adaptation; (re)routing of all demands	distributed	distributed algorithm making ingress nodes move traffic to alternative paths to enable power reduction in one other node or link	Adaptively periodic (depending on RTT)
[74]	capacitated design	single-layer network	single matrices of (aggregate) demands	yes	min. power consumption by links and nodes obeying energy profiles (~10 % load-dependent consumption)	direct optimization using transformed network graph modeling bridged operation modes of nodes	routing of all demands; deactivation of links; node operation mode selection (off/bridged/on)	centralized	MILP (on transformed network graph)	n/a
[75]	capacitated design	single-layer network (e. g. IP/MPLS)	one static matrix of (aggregate) demands	yes	min. power consumption by elements (e. g. nodes) obeying energy profiles	direct optimization	deactivation of / power scaling in nodes; routing of all demands	centralized	MILP	n/a
[76]	capacitated design	two-layer network (IP/MPLS over WDM)	one static matrix of (aggregate) demands	no	min. power consumption by routers and interfaces (with rate adaptation)	sequential routing of demands in full-mesh virtual topology with load-dependent link weights (discouraging routing on unused links)	virtual topology design; routing of all demands	centralized	simple single-layer routing heuristic and elimination of unused virtual links	n/a

ref.	problem	scenario	traffic model	split	objective	approach	degrees of freedom	centr./dist.	method	trigger
[77]	capacitated design (finding link weights)	single-layer network (IP OSPF routing)	single ones and a sequence of matrices of (aggregate) demands	yes (ECMP)	min. power consumption by nodes and links	integrated link weight optimization; sequentially determining residual topology and link weight assignment for feasible routing	link weight assignment for OSPF routing with deactivation of idle nodes and links	centralized	MILP formulation, iterative procedure of node and link deactivation by simple heuristic and relaxed MILP w/ tabu-search based link weight assignment heuristic (IGP-WO)	periodic (several hours)
[78]	uncapacitated multi-hour design	two-layer network (IP over WDM)	a sequence of matrices of (aggregate) demands (obtained by proportionally scaling one matrix)	yes	min. power consumption by active router chassis and line cards and for lightpath switching in OXCs	direct optimization: first for peak hour, then iterative definition of subset of topology (and optionally demand routes) for next-lower traffic period	virtual topology design w/ RWA (for peak); deactivation of line cards and chassis; (re)routing of / selection of route subset for all demands	centralized	MILP for each planning period with varying (reconfiguration) constraints	design for periodic (2h)
[79]	capacitated multi-hour design	single-layer network (IP/MPLS)	a sequence of matrices of (aggregate) demands	no	min. power consumption by active chassis and line cards over all periods and chassis during (re)activation	direct optimization with constraint on line card switching; sequential accommodation of demands with minimized power consumption and congestion	LSP routing and link/node deactivation	centralized	ILP, GRASP heuristic sequentially solving two ILP instances minimizing power consumption and congestion for this power value	periodic (variable, ~several hours)
[80]	reconfiguration	single-layer network (IP)	continuously evolving aggregate demand values; evaluation uses a sequence of matrices of (aggregate) demands	not discussed	min. power consumption by links	node-local decision on link (de)activation based on required power, network connectivity, and past experience regarding decisions causing overload	deactivation of links (with traffic rerouting in resulting topology)	distributed	learning-based local link (de)activation heuristic making use of OSPF link state advertisements (LSA)	LSA reception at any node
[81]	reconfiguration	single-layer network (IP/MPLS)	one static matrix of (aggregate) demands, scaled by sinusoidal profile for reconfiguration	no	min. power consumption by links	shortest-path routing of LSPs with link weights depending on utilization, resource state, and requested capacity	rerouting of demands upon modification with deactivation of idle links	distributed	simple routing algorithm	LSP capacity adaptation upon threshold violation

ref.	problem	scenario	traffic model	split	objective	approach	degrees of freedom	centr./dist.	method	trigger
[82]	reconfiguration	two-layer network (IP/MPLS over WDM)	Poisson demand arrivals and departures, intensity modulated by sinusoidal day profile	no	min. power consumption by active circuits and electronic traffic processing	addition/removal of virtual links based on thresholds on link load or aggregate traffic for one destination	virtual topology adaptation (in terms of links); rerouting of affected demands	distributed	threshold-based link addition/removal heuristic with simple shortest-path demand rerouting	link-load thresholds
[83]	reconfiguration	two-layer network (IP over WDM)	sequences of matrices of (aggregate) demands (based on measurements)	no	min. power consumption by active line cards, line card chassis, and fabric card chassis	identification of circuits to add or remove based on connectivity and link load	virtual topology adaptation; rerouting of affected demands	centralized	simple circuit addition and deletion heuristics with simple demand rerouting	periodic (15 min) with link-load thresholds
[84]	reconfiguration	two-layer network (IP over WDM)	sequences of matrices of (aggregate) demands (based on measurements)	no	min. weighted sum of power consumption by active line cards and traffic newly routed onto links	virtual link deactivation if traffic reroutable on shortest paths; heuristic optimization of virtual topology (regarding circuits); connectivity and link load based addition/removal of circuits from [83]	link deactivation or virtual topology adaptation; rerouting of affected demands	centralized	MILP formulation, simple link deactivation heuristic, genetic algorithm optimizing virtual topology, simple circuit addition / deletion heuristics; all with simple demand (re)routing	periodic (15 min)

3.2 Multi-Layer Network Design and Reconfiguration

This section systematically discusses research work on network design and reconfiguration from different points of view. We start by addressing different networking scenarios partly defined by varying device capabilities assumed in the literature in section 3.2.1. On this basis, section 3.2.2 classifies uncapacitated and capacitated design problems from literature according to their constraints and the degrees of freedom they exploit. In section 3.2.3, we complement the classification of problems by detailing specific properties of network reconfiguration. Section 3.2.4 surveys objectives targeted by network design and reconfiguration. We conclude with an overview of the applied methods in section 3.2.5.

As stated in chapter 4, by this monograph we aim at filling a gap in the research on reconfiguration methods for the green operation of IP/MPLS-over-WDM or similar two-layer networks. Like the relevant related publications, we assume a conventional WDM layer network with a fixed, uniform line rate and a uniform spectral width per wavelength channel as provided by the ITU 50 GHz grid [G.694.1]. Accordingly, we do not detail recent research contributions on flexible rate and flexible grid or gridless optical networks, respectively, in the following discussion, since such work focuses on specific properties of these extended scenarios which do not apply under our assumptions. We likewise excluded such publications from the overview in section 3.1.

3.2.1 Network Scenarios and Node Capabilities

The problem of designing and configuring multi-layer networks became relevant with the emergence of WDM, which did not only increase the transmission capacity exploitable in an optical fiber compared to using a single wavelength[1], but also sparked interest in optically interconnecting nodes which are not direct neighbors in the physical topology. This creation of a virtual topology which is different from the physical one was initially applied to physical fiber rings traditionally predominant in metro networks, cf. [85] and references therein. First publications addressing the embedding of virtual topologies on top of physical mesh topologies assumed so-called *wavelength-division optical networks* (WON), which broadcast optical signals throughout the network and receive them wavelength-selectively in nodes [53, 86, 87]. These studies consider the transmission capacity of a wavelength as abundant while nodes only have a small number of (expensive) ports.

Later research assuming wavelength-routed networks as the lower layer network, i. e. nodes with wavelength-selective optical switching capability, is more relevant to this monograph. Such networks allow the reuse of the same wavelength for different optical circuits if they are not routed over the same fiber. The four problem dimensions stated in section 2.1.2 apply to this scenario. A body of earlier work in this area addresses the design and configuration of networks under a set of constraints implied by limited capabilities of the network nodes, as e. g. surveyed by Huang *et al.* [88]: Optical equipment like transmitters, receivers, and optical add/drop multiplexers (OADM) may only be able to operate on one fixed wavelength, or they

[1]While a single optical carrier using more bandwidth would likewise support an increased data rate, limitations of electrical processing at the sender and receiver side eliminate this option.

may be tunable to a subset of wavelengths. Likewise, OXCs may be limited to jointly switching all wavelengths arriving in one fiber, or to switching wavebands, i. e. contiguous sets of wavelengths. Besides, multi-layer nodes basing on OADMs may feature dedicated (electronic) add-drop multiplexers (ADM) for each dropped wavelength in the upper layer. In this case, traffic processed in the electronic domain for forwarding to other nodes cannot be switched to another wavelength. This is different if one digital cross-connect (DXC) terminates all dropped wavelengths[2]. While research mostly targets networks exclusively composed of multi-layer nodes, some publications assume that a subset of nodes is only equipped with OXCs. These nodes are accordingly restricted to switching entire optical circuits [46, 62].

Most of the more recent studies assume full flexibility in switching traffic electronically between lightpaths. Respecting wavelength continuity, lightpaths are also considered switchable from any incoming to any outgoing fiber. Assumptions on wavelength conversion capabilities differ. Dutta and Rouskas [89] distinguish the cases of *none*, *fixed* where one predefined wavelength is converted into another predefined one, *limited* where a wavelength from one set may be converted to a wavelength from another set, and *full*, i. e. converting any wavelength to any other. The network design and reconfiguration studies we survey either require wavelength continuity or provide full conversion capability in all nodes.

The transmission of signals purely in the optical domain is referred to as *transparent*, since it does not require decoding and understanding the modulated signal between source and sink. Transparent WDM layer networks are also referred to as all-optical networks (AON) [47, 48]. If done optically, wavelength conversion[3] preserves the *transparent* property. In contrast, wavelength conversion by intermediately converting the signal to the electronic domain is an *opaque* operation. The same applies to 3R regeneration. Networks using both transparent and opaque switching of optical circuits are called *translucent*. While most publications on multi-layer network design and reconfiguration consider transparent or translucent networks, some research focuses on opaque networks [44, 64]. In such networks, lightpaths are terminated in every node and the resulting bit stream is switched electronically. This opaque switching is more expensive in terms of hardware complexity and power consumption than (transparent) optical switching. However, it is still more efficient than terminating the optical circuit, which requires delimiting the packets contained in the bit stream, and switching these packets individually in the upper layer. In addition, opaque nodes inherently provide wavelength conversion.

One set of publications restricts the number of fibers per direction on a physical link to one, e. g. [10, 11, 25, 63, 78], limiting the number of available wavelength channels and making wavelength assignment more challenging in case conversion is not possible. Multi-fiber networks do not have the restrictions of such single-fiber networks. They are assumed in a number of investigations, e. g. [24, 26, 49, 55, 56, 64, 66], as well as in this monograph.

Equipment terminating optical circuits like TRXs and TXPs generally feature one input port and one output port intended for bi-directional circuits. Accordingly, some research requires lightpaths to be bi-directional, e. g. [10, 11, 21]. Other publications do not obey this restriction of current hardware, e. g. [25, 26, 45, 49, 54–56, 78].

---

[2]IP routers in the IP/MPLS-over-WDM architecture likewise exhibit this property.

[3]See [8], section 2.7.1 for technological options for wavelength conversion.

The multi-layer network design problem outlined in section 2.1.2 defines an optical circuit as a point-to-point connection implying a single link in the virtual topology. While most researchers adhere to this assumption, some publications propose optical multicast capabilities. They are either exploited in *drop-and-continue nodes* [58] locally receiving part of the information from an optical circuit which is simultaneously forwarded to another node. Alternatively, signals are optically split and forwarded to several other nodes, generalizing the concept of lightpaths to *lighttrees* [90] in order to efficiently support multicast services and to reduce required hardware resources. For this monograph, optical multicast capabilities are out of scope.

3.2.2 Types of Design Problems

The degrees of freedom in designing a multi-layer transport network, in particular for a green-field installation, bring about a number of different optimization problems which have been discussed by researchers. This survey covers publications assuming that both the multi-layer network architecture and the location of nodes are given. The latter is often the case in practice when nodes are placed in proximity to traffic sources and sinks, i. e. human agglomerations or data centers. In addition, we exclude studies on the installation and routing of fiber links, i. e. the design of the physical topology, since unrelated to our reconfiguration problem.

In section 3.2.2.1, we survey reconfiguration problems with regard to the dimensions they cover and the degrees of freedom they exploit. Section 3.2.2.2 then discusses the effect of different properties of traffic and demands. We finally address some extensions and variants of network design problems which are not in the main focus of this monograph in section 3.2.2.3.

3.2.2.1 Problem Dimensions

The remaining design problems target accommodating a certain traffic load in a network with a given physical topology and architecture. As shortly introduced in section 3.1, they comprise uncapacitated and capacitated problems [18]. Solutions to the former specify the quantity of resources to be installed on network elements (e. g. fibers per physical link, ports and lightpath switching capacity per node) in addition to the network configuration (i. e. existence and routing of lightpaths along with traffic routing in the virtual topology). Minimizing the capital expenditure (CAPEX) for these resources is a common optimization criterion in such problems. The latter problem class targets accommodating the traffic using predefined resources in a suitable configuration. The amount of installed resources may result from an uncapacitated problem with a different traffic load, e. g. representing an imprecise prediction. Some earlier studies simply assume a uniform number of resources in all elements, e. g. the same number of ports in [48]. As detailed in section 3.1, some of the reviewed publications discuss uncapacitated problems [24, 44, 45, 51, 65], others make use of such problems to obtain resource limits for further studies [21, 83]. The majority, however, describes capacitated problems, which bear more similarity with reconfiguration problems.

Literature discusses the related network design problems under several terms referring to the subproblems outlined in section 2.1.2. The first one is *traffic grooming*, which, in a strict sense, describes the efficient multiplexing of smaller (so-called *sub-wavelength*) traffic demands onto optical circuits [91]. This term was widely used for synchronous digital hierarchy (SDH) ring networks and later generalized to mesh networks [85]. A traffic grooming problem is most efficiently solved if the decision where to establish optical circuits is made at the same time. For the second term, *virtual topology design*, the inverse argument applies. In a strict sense, it describes the problem where to establish optical circuits including their realization in terms of RWA [89]. However, traffic routing in the virtual topology needs to be considered during virtual topology design. Consequently, *traffic grooming* and *virtual topology design* are often used interchangeably when referring to some or all of the four sub-problems of multi-layer network design. Mukherjee [8] gives a detailed discussion of the problems covered by both terms in WDM networks.

While some research targets high-quality solutions by addressing all network design subproblems simultaneously, other work takes a sequential approach, first defining virtual topology and demand routing and then trying to perform RWA to implement the virtual topology. Regarding the interworking of the layer networks, the former is referred to as peer model and the latter as client-server model [52]. The approach of De Maesschalck *et al.* [45] falls in-between: they iteratively refine the solution of single-layer heuristics by contemplating costs resulting in the respective other layer. A number of further publications [21, 46, 49, 55, 64] avoids addressing the wavelength assignment problem by assuming full wavelength conversion capability. Implying sufficient wavelength channels on all links, others [11, 51, 82–84] completely disregard the RWA problem of the lower layer.

3.2.2.2 Demand and Traffic Properties

The traffic load to be accommodated is central to a network design problem. Depending on the architectural choice for the upper layer, traffic demands are either requests for time division multiplex (TDM) channels of a certain bit rate in case of SDH or optical transport networks (OTN), or they specify a packet flow of an arbitrary (average) bit rate between two nodes in a packet-switched layer network. While requests for TDM channels, which feature one out of a discrete set of bit rates, are generally accommodated individually, packet traffic demands between a node pair may or may not be aggregated into a total traffic rate between the nodes. Aggregation simplifies the design problem without restricting the design space if traffic demands may be bifurcated, or split, in order to route their parts along different paths from the source to the destination nodes. Otherwise, separately routing the traffic demands (which may technically be differentiated by using label switched paths (LSP)) is advisable to preserve more degrees of freedom in network design. Aggregated packet traffic demands are commonly presented in a square demand matrix with a zero diagonal (since traffic local to one node does not appear in the network design problem). Some studies simplify the design problem by assuming symmetric demands [10, 21], i. e. the same volume of traffic flowing in both directions between a node pair, although generally demands are asymmetric. Section 3.1 details the demand models of the related publications.

In principle, aggregating different packet traffic flows on optical circuits can bring about statistical multiplexing gains. This underlies the application of effective bandwidth theory in IP-over-WDM networks [92]. In core networks, traffic demands are generally highly aggregated, which limits multiplexing gains. Most publications – including this monograph – therefore assume the bit rates of packet flows to simply add up upon aggregation.

3.2.2.3 Further Variants and Aspects of Network Design

In multi-domain scenarios which comprise networks belonging to competing operators, network design has to cope with a limited availability of topology and load information. For the design of the problem and methods proposed in this monograph, we exclude this additional degree of complexity and assume a single-domain setting with all information available. Accordingly, the related publications listed in section 3.1 only target single-domain scenarios.

Multi-hour and multi-period design extends static network design problems towards dynamic operation. Multi-period design addresses the step-wise upgrade of network resources – along with required changes to the network configuration – for traffic demands evolving over planning periods of months or years [18]. Multi-hour design likewise makes use of several demand matrices, albeit applying to shorter time periods in the order of hours. It essentially determines network reconfiguration cycles with perfect knowledge of future (often periodic) traffic demands and may simultaneously dimension required resources. Among others listed in section 3.1, Skorin-Kapov *et al.* [51] and Zhang *et al.* [78] discuss such problems.

Early work on virtual topology design, e. g. surveyed in [89], addressed the embedding of regular virtual topologies on arbitrary physical topologies. The advantage of such topologies is well-controlled properties like the maximum number of virtual hops on any path. On the downside, pre-defined topologies restrict the possibilities to take traffic demands into consideration, which may turn out particularly problematic for strongly unbalanced traffic. The work by Ramaswami and Sivarajan [48] is one of the first to allow arbitrary virtual topologies.

Some more recent publications investigate multi-layer network design with a lower layer supporting circuits of several discrete bit rates, where the cost of TXPs and the optical reach may depend on the rate. For dynamic operation, studies consider elastic line rates (i. e. TXPs switchable to different bit rates) in addition to the case of statically defined multiple line rates. As argued above, we do not survey such publications since they are not related to our contribution. Impairment-aware multi-layer design [10] extends this concept by taking the interference of parallel optical channels into consideration in addition to the line rate to determine the optical reach.

High reliability requirements generally imposed on transport networks necessitate appropriate resilience mechanisms. Traditionally, protection has been the mechanism of choice in core networks. I. e., dedicated backup resources are allocated and preconfigured to absorb the traffic carried by a failing link or path. The allocation of (node or link) disjoint backup paths is part of network design and has to be performed simultaneously to the definition of the primary working paths for optimal configurations. Section 11.4.3 of [8] formulates the design problem with protection in the lower layer. However, protection (generally realized by resources in hot standby) is not the only means to achieve the required degree of resilience. If deactivated idle resources are activatable sufficiently fast as assumed in [71], they may likewise be configured in real time to carry traffic interrupted due to a failure. This gives way to restoration mechanisms, which put fewer constraints on network design. Pickavet *et al.* [93] give an overview of approaches to achieve resilience in multi-layer networks and their potential interplay.

3.2.3 Network Reconfiguration

Network reconfiguration means modifying the configuration of a network to adapt to changed conditions. While such a change may concern the available resources, e. g. in case of an outage, we focus on research responding to changes of the traffic load while assuming unchanged quantities of installed resources. Basically, deciding on the reconfiguration corresponds to solving a capacitated network design problem for the new traffic demands. In addition, a way to transition from the old to the new configuration has to be devised. Since changing the configuration of an operational network may bring about QoS degradations (e. g. short service interruptions), limiting the extent of configuration changes is often an additional goal.

In the following, we first discuss different dynamic traffic models and the way they influence the time and scope of reconfiguration in section 3.2.3.1. Section 3.2.3.2 then outlines the effect of varying assumptions on the temporal availability of demand information. We then survey approaches limiting the extent of reconfiguration and means to move from one configuration to the next in section 3.2.3.3 and section 3.2.3.4, respectively. Section 3.2.3.5 finally comments on signaling to enable reconfiguration.

3.2.3.1 Traffic Models and Trigger Conditions

The type of the assumed dynamic traffic model generally influences both the class of methods to obtain the new configuration and the condition triggering the reconfiguration. The first type models the arrival and departure of requests for TDM channels or reserved packet transmission capacity [59–61]. For simulation studies, researchers often assume stochastic models with negative-exponentially distributed request inter-arrival and holding times, where the source and destination nodes of a request as well as the required bandwidth are likewise randomly chosen. For such traffic models, the reconfiguration problem consists in accommodating newly arriving requests in existing or newly established optical circuits, and in deactivating unneeded resources upon departure of a request. A number of studies (e. g. [59, 60]) on this *dynamic traffic grooming* problem [59] disallows rerouting ongoing traffic flows of other requests in order to prevent service interruption, considerably limiting the degrees of freedom of the reconfiguration. Although basing on a different traffic model, the proposal by Vasić and Kostić [73] is a counter-example rerouting existing LSPs. Scharf [82] likewise reroutes ongoing traffic flows. This publication uses a request arrival and departure based model to generate traffic for the investigation of a reconfiguration scheme triggered by link load thresholds.

The second type of traffic model reflects continuous changes of the data rate in packet-switched networks. While some researchers actually formulate continuous-time functions for the demand values and make use of rate sampling to take reconfiguration decisions at discrete instants of time [54, 81], most publications make use of a series of demand matrices applying to consecutive time intervals or discrete instants of time. Such matrices may be derived from measurements [21, 26, 56, 69, 72, 80, 83, 84] or created according to different models with [52, 69] or without [49, 77, 79] a temporal component. A new configuration is either determined periodically for each of these matrices [21, 52, 69, 72, 84], or load thresholds are evaluated first in order to determine whether or not to trigger a reconfiguration [26, 54, 56, 83]. Some authors also evaluate distributed reconfiguration schemes conceived with asynchronous triggers using sequences of demand matrices [80]. Assuming continuous variations of traffic rates, this monograph develops periodic reconfiguration methods and evaluates them using sequences of demand matrices derived from measurements.

Zhu and Mukherjee [85] call reconfiguration for a request arrival and departure model *dynamic* and refer to reconfiguration based on consecutive demand matrices by *static* due to the resemblance with static design problems. Iovanna *et al.* [57] propose a hybrid scheme combining both types: while the arrival, departure and bandwidth modification of LSPs is handled by limited, incremental modifications, the complete network configuration is re-optimized upon significant changes of the traffic pattern. An algorithm for the latter task is discussed in [46].

3.2.3.2 Knowledge of Future Traffic

The scope of assumed knowledge of future traffic situations is a further differentiating factor in addition to the traffic model. The majority of dynamic grooming schemes react upon the arrival or departure of demands. They thus do not require any traffic forecasts, but need to determine and implement a new configuration in real time to accommodate arriving demands.

A number of demand matrix based reconfiguration schemes content themselves with the assumption that traffic evolves smoothly, i. e. without significant changes in the traffic volume in short time. They only partially load resources in the new configuration in order to preserve a capacity buffer for traffic increases [26, 54, 56]. In many of these cases, reconfiguration is triggered when resource utilization (e. g. link load) violates thresholds.

Other authors assume perfect knowledge of the peak traffic demands in the time interval during which the newly determined configuration is valid [21, 49, 72, 84]. Accordingly, it is sufficient to provide precisely as much capacity as required according to these forecasts in the new configuration. In most of these cases, reconfiguration is performed periodically without considering whether changes in the traffic load necessitate an adaptation. The network re-design problems solved for the predicted demands correspond to those only partially loading resources if the forecasts are obtained as the current traffic demands scaled up by a factor of one over the maximum resource utilization.

Finally, the assumption of complete knowledge of the future evolution of (generally periodic) traffic demands gives way to the pre-computation and scheduling of consecutive network configurations. This is referred to as multi-hour network planning as e. g. proposed in [11, 51, 78, 79].

3.2.3.3 Limiting Reconfiguration Effort

Some studies focusing on maximum benefits achievable by network reconfiguration disregard the amount of modification between subsequent configurations, i. e. they compute independent configurations optimized for each demand matrix [21]. Other research limits reconfiguration either by only contemplating adaptations of restricted scope or by a cost penalty for or constraints on configuration changes. By only performing changes required to accommodate individual demands, the dynamic grooming methods of [59, 60] fall into the first category. This applies in particular if rerouting of ongoing traffic flows is not allowed. Reconfiguration methods based on a sequence of demand matrices can likewise by principle restrict the extent of modifications. For instance, Gençata and Mukherjee [54] use optimization to determine a single optical circuit to be established or torn down per reconfiguration event. Only rerouting traffic without changing the virtual topology [52] is a different way to restrict the scope of reconfiguration.

As mentioned above, another approach consists in applying a penalty or constraints on the extent of reconfiguration. For this, the extent is quantified in terms of either the volume of rerouted traffic or the number of modified optical circuits. The former quantity appears as a term in the cost function in [83, 84]. Tran and Killat [26, 56] minimize a cost function being a weighted sum including the number of added optical circuits. Ramamurthy and Ramakrishnan [55] take a different approach. Instead of co-minimizing circuit changes, they restrict the number of added or removed circuits as well as the number of affected fiber links by a constraint. Their proposal is derived from the work by Banerjee and Mukherjee [49], which first determines a minimum cost value for the primary objective without considering the reconfiguration effort and then minimizes the reconfiguration term while setting the minimal primary objective as a constraint (i. e. they seek the configuration requiring the least amount of circuit modifications among those minimizing the primary objective). A drawback of this procedure is that it cannot trade the value of the primary objective off against the reconfiguration effort, thus missing slightly worse solutions (in terms of the primary objective) which require significantly less modifications. In multi-hour design studies, the complete knowledge of the future traffic allows limiting the number of configuration changes over a day [79].

In addition to the extent of configuration changes incurred at each reconfiguration event, the frequency of such events, i. e. the *reconfiguration policy* [94] determines the effort of reconfiguration and potential QoS impairments. For periodic schemes, one can directly set the time between reconfiguration events. For schemes evaluating sampled traffic rates against thresholds, it is likewise possible to adapt the sampling interval. Moreover, many such schemes apply a hysteresis in terms of different thresholds for circuit establishment and teardown in order to prevent frequent reconfiguration if the load fluctuates around the threshold [26, 54, 56, 83]. Leaving headroom between the maximum link utilization allowed in a new configuration and the high watermark triggering the next adaptation [26] likewise delays the next reconfiguration event.

3.2.3.4 Transition Between Configurations

Besides finding a new configuration, reconfiguration requires devising a way to move the network from the previous to the new configuration. This is particularly important for the modification of optical circuits, since this may require a non-negligible amount of time and circuits occupy limited hardware resources that may be used differently in subsequent configurations. While in principle the latter likewise applies to capacity reservations in the upper layer, rerouting of traffic can be effectuated faster and is thus less critical. If addressing the transition between configurations, most publications therefore only consider the optical layer.

Since reconfiguration methods yielding single changes to the network configuration usually respect resource limitations, implementing their solutions trivially means activating or deactivating the optical circuits they indicate. This applies to both dynamic grooming schemes [59, 60] and demand matrix based reconfiguration of single circuits [54]. A number of reconfiguration schemes aiming at energy-efficient operation limit the reconfiguration of the virtual topology to the deactivation and reactivation of optical circuits with pre-allocated resources, e. g. [10, 69–72, 77–79, 81] and FUFL (fixed upper, fixed lower) and DUFL (dynamic upper, fixed lower) in [21]. In these cases, the transition between configurations is also simple since resources cannot be occupied in a conflicting manner. The concept of *hitless reconfiguration* [50] enforces this property for arbitrary reconfiguration: only currently idle resources may be allocated for new circuits in the next configuration; resources freed by discontinued circuits cannot be reused immediately. While Bala *et al.* [50] discuss resource dimensioning and RWA to enable hitless transition between a series of given configurations, the methods proposed in this monograph as well as those presented in [26] achieve hitless reconfiguration under predefined resource constraints.

If a new configuration is computed without regard of the previous one [21, 49], called *direct approach* in [94], finding a *transition path* minimizing service interruption is a non-trivial task. Ideally, traffic is consecutively rerouted from circuits, which are then deactivated in order to reuse their resources for new circuits. The survey on reconfiguration in both optical broadcast and wavelength-routed networks by Golab and Boutaba [94] details approaches to transit between virtual topologies, which may bring about transient loss of traffic. Traffic loss may also result from pure rerouting of demands, e. g. due to packet reordering or the convergence time of routing protocols. However, the latter issue does not occur with centrally defined demand routes specified by many reconfiguration schemes.

Although mostly disregarded in publications on multi-layer network reconfiguration, moving from one demand routing configuration to the next involves the complex problem of finding a sequence of route modifications such that all intermediate settings respect capacity constraints. This becomes particularly challenging if each demand shall directly be rerouted to its definitive path in order to limit route modifications. Fortunately, simple heuristics proved to efficiently solve many realistic instances of such reroute sequence problems [95]. In the context of multi-layer network reconfiguration, the capacity of additional circuits active during traffic rerouting in hitless virtual topology reconfiguration tends to further simplify these problems.

3.2.3.5 Signaling for Configuration Changes

Control planes following the automatically switched optical network (ASON) [G.8080/Y.1304] architecture, generally implemented by generalized multi-protocol label switching (GMPLS) [96, RFC 3945], can signal centrally made reconfiguration decisions to network elements, as implicitly or explicitly [61, 62, 76] assumed in the related literature. Some researchers also use or extend GMPLS protocols, in particular open shortest-path first with traffic engineering extensions (OSPF-TE) [RFC 3630], to realize the information exchange required for distributed reconfiguration schemes [57, 80, 81]. This monograph focuses on centralized methods to determine new network configurations. This centralized decision-making is in line with the current trend towards software-defined networking (SDN) [97]. While acknowledging that GMPLS provides a viable solution to signal the new network configuration, we consider ways to realize network reconfiguration out of scope and accordingly do not review literature related to this aspect.

3.2.4 Design and Reconfiguration Objectives

Multi-layer network design and reconfiguration target a variety of objectives, which range from minimizing equipment cost over preparing networks for traffic increases and assuring QoS to minimizing energy consumption. Optimization problem formulations express these objectives explicitly in their cost function. While heuristics may likewise evaluate the cost of local choices regarding these objectives, such algorithms generally take decisions which are only implicitly favorable for the objective. For instance, traffic routing which exploits the spare capacity of already well-loaded optical circuits helps minimizing the number of circuits [81].

Uncapacitated design problems often aim at satisfying given traffic demands while minimizing CAPEX for installing the required equipment. Thereby, researchers apply different cost models. They may purely focus on the optical layer, applying cost values for installed optical fibers or their length, respectively [24]. Alternatively, only TRX are counted [51], or cost is assigned to elements of both layers: IP routers, their line cards, OXCs, and WDM line systems [4] [45]. Höller and Voß [44] only count line card ports for the upper layer, but include the option of using cheaper non-WDM systems on fiber links instead. In the context of green networking, a similar optimization problem arises for minimizing the static power consumption of installed equipment, e. g. considering different types of router chassis and line cards [65].

One objective in capacitated design and reconfiguration problems is avoiding hot spots of scarce spare capacity, which limit the ability of the network to absorb increases of traffic demands. For this purpose, *congestion*[5] is defined as the maximum of the utilization over all relevant network elements (generally links or nodes, respectively). Complete reconfiguration of the network by optimization allows minimizing this congestion, e. g. on virtual links in [11]. Heuristics for the addition of single optical circuits [54] or traffic demand and lightpath routing during network design [46] likewise aim at minimal congestion. It is also possible to compute a minimally congested network configuration without absolute demand values, i. e. if demands are only indicated relatively to each other [48]. In this case, minimizing congestion corresponds to maximizing the average demand value the network can support under the given demand distribution.

A second objective related to preparing the network for increasing traffic is minimizing the overall resource usage. The respective problems bear resemblance with network dimensioning problems minimizing CAPEX. Minimizing the number of optical circuits, pursued along with other objectives in [11, 26, 55], falls into this category. This objective is likewise found in green network configuration studies attributing energy consumption to active circuits, cf. section 3.3.

[4] A WDM line system comprises multiplexers and demultiplexers required at both ends of a WDM fiber as well as optical amplifiers.

[5] The usage of this term in the context of network design differs from the common definition in traffic engineering.

A traditional goal of network design and reconfiguration is ensuring QoS. One common criterion in this regard is the average packet delay, which is composed of contributions from queuing and propagation. Provided that resources are not excessively loaded, queuing delays are small compared to propagation delays in WANs. Some studies therefore approximate packet delays by the latter, e. g. [48] (which bounds packet delay by a constraint instead of minimizing it in the cost function). By assuming uniform propagation delays on all links, minimizing the average packet delay reduces to minimizing the total flow, i. e. the sum of the traffic carried by all links [49, 55]. The same objective is used in further publications to favor resource-efficient traffic routing along short paths [26] and to limit the need for packet processing requiring complex and thus expensive hardware [11]. When accounting for the power consumption of active packet processing resources, green networking methods likewise aim at minimizing the total flow. Combining this objective with minimizing the number of optical circuits targets efficient resource usage in both network layers [26, 55].

A further aim related to QoS is minimizing traffic blocking. The capacitated multi-hour network design problem in [52] explicitly minimizes blocking across all traffic matrices. The idea of reducing the risk of blocking for unknown future traffic underlies the objectives of minimizing congestion and resource usage detailed above. Heuristics for dynamic grooming likewise try to limit the blocking probability of future requests by maintaining spare capacity on all relevant resources [62] or by minimizing overall resource usage [58]. Fairness is another objective considered in the scope of QoS. For instance, Zhu and Mukherjee [25] target fairness in terms of equal blocking probability for requests of different bit rates.

The characteristic of green network design and reconfiguration is the objective of minimizing energy consumption. However, the assumed power models and consequently the elements of network design which impact the total power consumption vary between publications. We detail these aspects in section 3.3. Some of the work on energy-efficient multi-layer networking also respects traditional network design objectives. One way to do so is to restrict the respective objective function values by constraints. For instance, Bianzino *et al.* [72] apply an upper bound on network congestion. This approach is also used to combine traditional design objective, e. g. to minimize congestion while constraining packet delay and electronic processing [48] or to minimize the number of circuits and the total flow while constraining congestion [26].

3.2.5 Design and Configuration Methods

The related publications propose a variety of methods to solve network design and reconfiguration problems. Basing on the general description of solution methods in section 2.4, we discuss the specifics of their application to the respective problems in this section. We first address simple heuristics for limited modifications and the complete (re)definition of the network configuration, respectively, in section 3.2.5.1 and section 3.2.5.2. Section 3.2.5.3 and section 3.2.5.4 then cover applications of heuristic and mathematical optimization methods, respectively. We describe combinations of several methods in section 3.2.5.5 and finally comment on distributed schemes in section 3.2.5.6.

3.2.5.1 Simple Heuristics for Limited Modifications

To route arriving requests, many dynamic grooming schemes use simple first-fit heuristics, i. e. they use the first route they contemplate which is able to accommodate the additional traffic. For instance, requests may be routed over a series of existing circuits if feasible, and otherwise a new circuit is established from source to destination node of the request [60]. Some authors map the whole decision process of how to accommodate a demand onto the computation of a shortest path in a so-called *auxiliary graph*. Such a graph may feature one layer mimicking the physical topology per available wavelength to model wavelength assignment, with nodes accordingly decomposed into several vertices whose interconnection represents wavelength conversion and grooming capabilities [58]. Edges interconnecting network nodes do not only represent existing (physical or virtual) links; they may likewise stand for potential links in the virtual topology. The auxiliary graph is therefore also referred to by *reachability graph* [60]. Auxiliary-graph based methods can implement different grooming strategies by according assignment of edge weights. Such weights may be static (e. g. imposing a high cost for wavelength conversion) or a function of the network state, in particular dependent on link loads [59, 62]. Constraint based routing constitutes a similar approach to respond to single requests [57]. The same set of methods is applied for rerouting LSPs when changing the reserved capacity [57, 81].

Some reconfiguration schemes computing a single modification to the network at a time likewise make use of first-fit algorithms. Upon each reconfiguration event, Gençata and Mukherjee [54] add one optical circuit for the largest demand not routable due to network partitioning if such a demand exists, otherwise they add a direct circuit offloading the largest demand from the most highly loaded link if the load on some link exceeds a high watermark, or otherwise they remove the least loaded circuit (of those with a utilization below a low watermark) which either does not carry any traffic or whose elimination does not partition the network. Idzikowski *et al.* [83] apply a similar strategy. Based on thresholds for the transit traffic volume between two neighbors of a node, Scharf [82] establishes and eliminates bypass circuits. Alternatively, bypasses are established directly to a destination node if the respective traffic volume exceeds a threshold.

3.2.5.2 Simple Network Design and Reconfiguration Heuristics

For the design or the complete reconfiguration of networks, a number of researchers apply simple heuristics suitable for dynamic grooming (as discussed in section 3.2.5.1) sequentially for all traffic demands. The sequence of iteration is thereby an additional degree of freedom. Possible criteria include the demand volume [25, 49, 64, 67], the minimum number of physical hops a demand has to travel [46], the ratio [25], or the product [49] of both (the latter corresponding to the contribution to the total flow). Common routing strategies include setting up a direct circuit for each considered demand until resources run out and routing the remaining demands in the resulting topology [25, 49]. Optionally, circuits are initially established on all physical links (e. g. in [49] which minimizes the total flow). Alternatively, demands are preferentially routed on existing circuits, and a direct circuit from the source to the destination of the demand is established if this fails [46, 64]. Such routing is also applied in an auxiliary graph (called *wavelength graph*) in [47]. Optionally, the result of a routing heuristic is subjected to a *local improvement* algorithm, which is another first-fit heuristic. It iterates over lightly loaded links and tries to reroute the traffic they carry and to deactivate them in order to free their resources [46].

Publications targeting energy savings by rerouting traffic and deactivating circuits of a fixed virtual topology, which is essentially a single-layer problem, likewise route demands sequentially. In their traffic engineering approach, Puype *et al.* [76] thereby apply load-dependent link weights which increase for both low and high link utilization. This removes traffic from links that would otherwise be lightly used, allowing their deactivation, while preventing overloading other links. Huang *et al.* [67] take a different approach to energy-efficient traffic grooming. They start with all links deactivated, and activate them in increasing order of the implied power consumption until they are able to route the demand under consideration. For energy-efficient RWA, a related strategy sequentially routes lightpath requests depending on the load on fiber links [66].

Regarding energy savings by deactivating network elements, one also finds a different type of first-fit heuristic. Taking a single-layer view, Chiaraviglio *et al.* [70, 71] iterate over each node and link, respectively, and deactivate it if they can reroute the affected traffic in the residual network under maximum link load constraints. Thereby, they apply link weights inversely proportional to link capacity. Investigated iteration orders include random, least-flow, and most-power for both nodes and links, as well as least-link (i. e. minimum nodal degree) for nodes. A different strategy explicitly deactivates redundant edge nodes.

A number of publications listed in section 3.1 develop rather sophisticated algorithms for the design and reconfiguration of the upper layer, i. e. demand routing and virtual topology design, while applying a simple heuristic for the RWA problem: shortest-path routing of lightpaths with first-fit wavelength assignment (e. g. [25, 54], or without considering wavelength assignment due to assumed conversion capability, [21, 64]). Lightpath routing may be *fixed*, *fixed-alternate*, or *adaptive* [98] (as referenced in [25]). In the first case, one path is precomputed and lightpath setup fails if resources are depleted along this path. In the second case, several (e. g. the k shortest) paths are precomputed and tested for free resources. In the third case, the routing is performed upon lightpath setup considering the network state, in particular resource availability on links.

3.2.5.3 Meta-Heuristic Optimization

A number of authors perform network design by different heuristic optimization methods. For instance, Sengezer and Karasan [52] apply tabu search to optimize the virtual topology for multi-hour traffic. They obtain neighbor solutions by adding or removing individual optical circuits. In order to evaluate the cost of the candidate solutions in terms of blocked traffic, they route the demands of each hourly traffic matrix by a first-fit heuristic using load-dependent link weights.

Some publications make use of GRASP-inspired heuristics. Commonly, they generate several randomized iteration orders over demands or network elements. They then use these orders in sequential first-fit design heuristics as presented in the previous section 3.2.5.2. Höller and Voß [44] apply this approach for least-cost network design: Starting from shortest-path routing of all demands, they iterate over all nodes in randomized order and try to reroute the traffic sourced by the considered node. This rerouting uses load-dependent link weights which favor the concentration of traffic. For each source node, they keep the new routing if it reduces the total equipment cost. Cinkler *et al.* [47] likewise propose a GRASP-like *sequential shortest-path* (S-SP) heuristic for the routing of demands in an auxiliary graph. In a second step, they use a demand-wise rerouting scheme resembling the method of [44], however without randomized iteration order.

The authors of [47] additionally design a problem-specific algorithm making use of randomization. Instead of routing demands sequentially, they do so in parallel by extending candidate routes of randomly chosen demands by one hop of the shortest path feasible under resource constraints. In order to avoid getting stuck in an impasse, the strategy includes shortening or resetting candidate routes with a certain probability.

3.2.5.4 Mathematical Programming

Mathematical programming serves both to precisely specify a network design or reconfiguration problem and to obtain exact solutions by using solver software. All network design and reconfiguration problem formulations in the surveyed literature are MCF problem variants. In line with the introduction in section 2.3.1, most of these problems allow a linear formulation, but resources of discrete capacity entail integer-valued variables. We hence obtain ILPs or MILPs if bifurcated routing of demands introduces continuous-valued variables. Since these problems are generally NP complete, i. e. hard or impossible to solve exactly for medium-size to large problem instances, some publications directly proceed to defining heuristics after formulating a (M)ILP [47, 66–68, 70, 84], while others first apply an exact solution method to small problem instances [25, 44, 46, 49, 54, 64, 71] or relaxations thereof for lower bounds [48, 64]. A number of contributions also purely rely on solving (M)ILP formulations [10, 11, 21, 24, 26, 55, 63, 65, 69, 72, 74, 78], often citing the design of heuristics for future work.

Due to the similarity of many MCF-based problem formulations, we only address some particularities here. They primarily concern simplifications to make the problems computationally more tractable. For their reconfiguration problem, Tran and Killat [26] restrict the set of candidate lightpaths for modification based on demand fluctuations. Banerjee and Mukherjee [49] restrict the routing of demands to virtual links adjacent to the nodes on their k shortest paths in the physical topology, and they apply a propagation delay (i. e. physical length) bound for lightpaths.

3.2.5.5 Hybrid Methods

As outlined in section 2.4.1, algorithms exist which efficiently solve LPs without integer variables. On the other hand, heuristic methods conceived for combinatorial optimization are most suitable for problems with purely integer-valued variables. Likewise, simple heuristics for traffic routing rarely support bifurcation of demands[6]. For these reasons, a number of researchers combine heuristics for virtual topology design with an LP for bifurcated demand routing in this topology. Ramaswami and Sivarajan [48] do so in a strictly sequential way, first defining the virtual topology based on the volume of traffic demands and the distance of nodes in the physical topology. Referring to deactivating cables in bundled links, Fisher *et al.* [68] reduce the capacity of links in discrete steps as long as the routing LP remains solvable. De Maesschalck *et al.* [45] propose a more sophisticated heuristic for multi-layer network design involving the use of routing LPs: In alternation, they apply the two-step *forward synthesis and design tightening* heuristic [99] for the minimum capacity installation problem to the two layer networks while updating circuit demand and circuit cost, respectively. In the first phase, the two-step heuristic adds capacity on elements facing maximum excess traffic until the routing LP becomes solvable. In the second phase, it proceeds by removing the most expensive lowly loaded resources while solutions to the LP problem persist. Zhang *et al.* [69] likewise make use of a routing LP. However, they apply it to minimize congestion after finding a virtual topology minimizing the power consumption by means of a MILP.

Other publications embed MILPs describing smaller and thus less complex subproblems within heuristic solution procedures. Instead of leaving network design for the complete set of demands to the MILP solver, Addis *et al.* [79] device two MILPs to accommodate one demand at a time under the constraints of already routed traffic. The first MILP minimizes power consumption, and the second one minimizes congestion with the minimal power consumption as a constraint. A GRASP-like heuristic defines the sequence in which demands are routed by the MILPs. Skorin-Kapov *et al.* [51] likewise control the use of MILPs for subproblems by an optimization heuristic: Addressing a multi-hour planning problem, they apply tabu search to eliminate temporally lowly-used TXPs while solving the network configuration problem for each time period by a MILP with resource constraints.

Most of the publications listed in section 3.1 as well as the problem and methods investigated in this monograph assume a connection-oriented upper layer network where explicit routes are defined for individual demands. Another body of research targets networks with shortest-path routing in the upper layer without discrimination of demands. In this case, network design has to be extended by finding a link weight assignment resulting in an appropriate distribution of traffic on the virtual links. This non-trivial problem, which is out of scope for this monograph, is addressed by a mixture of simple heuristics and relaxed MILP formulations in a recent publication [77].

3.2.5.6 Distributed Schemes

Another criterion for the classification of network reconfiguration methods is whether they are applicable in a distributed manner. The majority of reviewed schemes as well as the methods addressed in this monograph are not. Some dynamic grooming algorithms using shortest-path routing with locally determinable link weights inherently allow a distributed implementation.

[6] Exceptions include equal-cost multi-path routing (ECMP) [RFC 2991], which serves for local load balancing rather than network-wide optimization, and the rerouting scheme of [73], which improves an already existing configuration.

Other authors explicitly develop distributed schemes. Vasić and Kostić [73] design an energy-aware traffic engineering scheme which determines the minimum traffic volume along each traffic path that needs to be removed to reduce the power consumption of one network element. The source node of a concerned demand then shifts an according share of traffic to alternative paths. Bianzino *et al.* [80] let nodes decide locally on link deactivation based on topology information gathered from link-state routing protocols (augmented with link load information) and historical information on the effect of particular actions. Unlike these contributions directly conceiving distributed schemes, Tran and Killat [56] derive a distributed algorithm from a dual problem to a reduced centralized ILP formulated in [26] by Lagrangian relaxation. This scheme iteratively computes a near-optimal solution to the problem in a distributed manner based on information exchanged with link state advertisements.

3.3 *Green* Networking

Since the seminal work by Gupta and Singh [100] suggesting energy savings by means of sleep modes in network devices, a lot of research has targeted energy efficiency in network design and operation. Motivations for such energy efficiency range from economic considerations, i. e. reducing the electricity bill, over environmental concerns, i. e. reducing green-house gas emissions, to fostering the deployment of the Internet in the developing world where energy resources are scarce [3, 100]. In addition, lower power consumption extends the time the telecommunication infrastructure can operate on uninterruptable power supply (UPS) batteries in case of a disaster [100]. The objectives expressed by these motivations may suggest contradictory approaches and go beyond the scope of network engineering: Reducing operational expenditures (OPEX) in terms of the electricity bill may favor shifting power consumption to times or places where energy is less expensive rather than reducing the overall consumption, whereas green networking in the strict sense of reducing green-house gas emissions may center on using renewable energy sources and performing networking functions close to energy sources to reduce electrical transmission losses [3]. We therefore follow Bianzino *et al.* in taking an *engineering point of view*, which defines green networking as “a way to reduce energy required to carry out a given task while maintaining the same level of performance” [3].

In this section, we survey the literature on green networking in this engineering point of view. After giving a general overview of this field in section 3.3.1, we discuss two aspects of particular relevance for this monograph in more detail: In section 3.3.2, we address power models required to quantify potential energy savings by resource adaptation, and we classify proposals for energy savings by load-dependent network operation in section 3.3.3.

3.3.1 Overview of Energy-Aware Networking Research

In this section, we give an overview of the different fields and aspects subsumed under the labels of *green networking* or *energy-aware networking* research with regard to fixed networks. Based on three survey papers on this field [2, 3, 101], we primarily focus on research targeting networks of telecommunication operators and Internet service providers ranging from access to core networks and from the physical to the IP layer, i. e. excluding end systems and the transport and application layers.

In section 3.3.1.1, we outline approaches to reduce the static, load-independent power consumption of networks. Next, we address approaches scaling power consumption with load in order to achieve savings in low-load periods. In section 3.3.1.2, we describe such approaches with a scope limited to node components and links, and proceed to network reconfiguration schemes in section 3.3.1.3. Section 3.3.1.4 shortly addresses protocol aspects. Finally, we describe different levels of power modeling in section 3.3.1.5.

3.3.1.1 Reduction of Static Power Consumption

Several technological and architectural approaches aim at reducing the static power consumption of networks. On the *component level* (classification according to [101]), progress of CMOS (complementary metal-oxide-semiconductor) technology in terms of smaller feature sizes allows operation at lower voltages and higher frequencies, resulting in lower energy dissipation per operation. However, these improvements lag behind the expected traffic growth, requiring additional effort to at least stabilize the energy consumption of networks. Also on the component level, the integration of more functions in single chips may enable further energy savings. On the *transmission level*, which concerns single network links or optical circuits, progress of optical technology like energy-efficient lasers and low-attenuation fiber reducing the need for amplification may likewise increase energy-efficiency without fundamental changes to the network.

Regarding the *network level*, a first approach without architectural changes is minimizing the total power consumption of installed devices instead of e. g. their cost during network design. Such green design has been proposed for both single-layer [65] and multi-layer [10, 63, 64] networks. The benefits of green over cost-oriented network design are however limited since the complexity of devices likewise defines their cost and power consumption [64, 102].

More promising proposals target the network architecture: Reducing the extent and flexibility of packet processing in network nodes allows reducing the hardware complexity in terms of the number of gates, which in turn reduces power dissipation. Increasing the share of traffic transported in the optical domain likewise helps reducing energy consumption thanks to highly energy-efficient optical switching. Long-reach passive optical networks (LR-PON) in the access and optical bypassing of electronic nodes in the core are such approaches. In addition, the use of mixed line rates helps to scale the power consumption of optical links with the load.

Further architectural approaches aim at simplifying packet switching at the IP layer. They include flow routing, i. e. identifying and switching packets belonging to one flow instead of taking a routing decision based on longest prefix match for each packet [103], and pipeline forwarding. This concept uses a parallel network structure offering synchronous TDM channels dubbed *super-highways* for IP traffic flows of substantial volume and thus offloads routers from handling the packets of these flows [104]. Finally, the redesign of certain protocols may yield direct energy savings. Examples include simplifying the encoding within the Ethernet protocol in order to reduce the related computation effort [105].

3.3.1.2 Dynamic Reduction of Power Consumption

Power proportionality, or *proportional computing*, describes the idea that the power consumption of a system should scale linearly with the workload. Diverse dynamic approaches to green networking aim at this goal. Much work focuses on the *component* and *transmission levels*, where researchers investigate schemes of similar nature, which have already been put forth in the seminal paper [100]. According to the taxonomy of [3], they are *local* and *on-line*, i. e. working on information locally available at a node or link, respectively, and performing adaptations on time scales below one second.

A first type of schemes reduces the operation rate of electronic node components or links under low load. According mechanisms are dynamic voltage and frequency scaling (DVFS) in processors and switching between different link speeds e. g. for Ethernet, dubbed adaptive link rate (ALR). A second type of schemes saves energy by shortly deactivating idle resources in-between the processing or transmission of two packets, referred to by idle logic or low-power idle. Both types of schemes may result in degradation of QoS, since packets may experience additional delay or loss. While these schemes have primarily been designed for local area networks (LAN) and access networks, the resource and power model we develop in section 4.1.2 applies some of the concepts to components of core network nodes. A third type of schemes, which is also amendable to core network components, deactivates resources for time periods beyond the scale of inter-packet gaps. Such sleep modes generally require explicit or implicit mechanisms on the *network level* to divert traffic from the concerned resources.

The publications name a number of adaptations reaching up to the *application level* required to enable the use of these local dynamic power-saving schemes. Routing protocols have to be aware of the deliberate temporary deactivation of links, since the potentially frequent link state change would cause routing instabilities. Likewise, the spanning tree protocol deployed in Ethernet networks shall not try to replace sleeping links. In addition, limiting the frequency of signaling message exchanges helps saving energy by preventing resources from being reactivated unnecessarily.

3.3.1.3 Network Reconfiguration

For a number of network components, technological limitations prevent strict power proportionality. For instance, chassis control, switching fabric, and power supply impose a base power consumption in network nodes which cannot scale linearly with the load. Likewise, DSPs for modulation and demodulation as well as lasers in TXPs sustaining optical channels of a constant bit rate need to work independently of the circuit utilization by packet traffic in order to maintain synchronization. Consequently, dynamic adaptation of such components is limited to deactivating them, i. e. putting them into sleep modes for substantial periods of time. To do so without severe QoS impact, it is necessary to divert the traffic from the respective resources unless they are already idle. In the general case, this cannot be achieved locally by a node. Hence, mechanisms enabling the sleeping of links and node components act on the *network level*. In the taxonomy of [3], they are *global* since they require more information than available at one link or node and *off-line*, i. e. effective on time scales above one second, due to the necessary coordination between nodes. Still, most of the schemes do not pertain to the network design phase, but target the reconfiguration of networks while in operation.

A straight-forward realization of this idea consists in traffic rerouting in order to free resources and prepare their deactivation. Gupta and Singh [100] introduced this concept under the term of *coordinated sleeping*. It is likewise referred to as *green routing* [101], in particular if realized by adaptations of the open shortest path first (OSPF) routing protocol, or as *resource consolidation* [3] if it preserves a certain overprovisioning factor. In section 3.3.3, we survey publications proposing explicit rerouting schemes for core networks to enable resource deactivation.

Alternatively, the rerouting of traffic in response to resource deactivation could be left to higher network layers. However, the response time (e. g. for routing protocol convergence) of such reactive schemes tends to entail considerable QoS degradation. We therefore do not consider this option for the reconfiguration schemes developed in this monograph and accordingly do not detail related work in this field.

In multi-layer networks, reconfiguration schemes can improve energy savings by exploiting a further degree of freedom in addition to adapting traffic grooming: the modification of the virtual topology. We detail green multi-layer network reconfiguration schemes reported in literature in section 3.3.3.3. Like dynamic power-saving schemes on the component and transmission levels, rerouting and network reconfiguration schemes may entail QoS degradation in terms of additional packet delay and packet loss due to longer traffic routes, congestion resulting from the concentration of traffic, and temporarily invalid routes.

Similar energy-saving approaches are found in access networks. They include dynamic spectrum management for digital subscriber line (DSL) technologies, which trades throughput to reduce crosstalk along with the required transmit power, and dynamic wavelength assignment in LR-PONs in order to deactivate transmitters in low-traffic periods. In wireless networks, different mechanisms target the deactivation of access devices solely installed for capacity reasons, i. e. serving areas already covered by other devices. Respective devices include UMTS base stations or parts thereof, and wireless local area network (WLAN) or wireless-optical broadband access network (WOBAN) access points. In metro networks, which traditionally are founded on optical rings, one proposal for energy-efficient operation is dynamic optical bypassing.

3.3.1.4 Protocol Aspects

As stated in section 3.2.3.5, an ASON/GMPLS based control plane may effectuate multi-layer network reconfiguration according to a centralized decision. Obviously, this also applies to reconfiguration aiming at energy savings. In addition, some protocols may require extensions in order to support power-saving features. For instance, one proposal augments the simple network management protocol (SNMP) [RFC 3410] with device power state information [101]. Further requirements on control protocols include line rate negotiation and coordination of sleeping [65]. Besides, routing protocols may support convergence in advance for scheduled topology modifications, thereby possibly respecting green traffic engineering objectives [71].

3.3.1.5 Modeling of Power Consumption

For designing efficient means to reduce the energy consumption of communication networks, researchers need to know which parts of and components in the network significantly contribute to this consumption. With regard to dynamic power saving schemes, further information on the dependency of the power consumption on load or operation state is required. Therefore, the creation of power models is an integral part of green networking research.

Some publications report on measurements of the power consumption of commercially available network devices. They may consider varying load and different software and hardware configurations, e. g. in terms of router chassis with a varying number of installed line cards [65]. These existing devices generally exhibit a power consumption almost insensitive to the load and do not allow dynamically activating or deactivating installed components.

Therefore, researchers proceed to establishing models for the power consumption of hypothetical future green network devices. This ranges from the simple assumption that larger components like line cards allow deactivation to sophisticated models based on a detailed analysis of power saving techniques applicable to architectural components of network nodes. In addition, simple theoretical prototypes of energy profiles (indicating power consumption over load) find application in studies evaluating the benefit of power saving schemes under varying scenario assumptions, e. g. [75]. We give a more detailed overview of energy models proposed for core network nodes in section 3.3.2.

In order to compare the energy efficiency of different technologies and to identify the network sections allowing for the largest energy savings, more coarse-grained power models have been proposed. They primarily characterize access technologies by their energy consumption per transferred bit or by the power consumption over the peak bit rate. One of the surveys [101] cites a model augmenting the access network by a content delivery network for Internet protocol television (IPTV).

The group of Professor Tucker [4, 5] took a more global approach in developing a comprehensive power consumption model for the Internet. For this purpose, they dimensioned access, metro and core networks for different access technologies based on assumed peak access rates per customer and simplified network design rules taking over-subscription ratios and the capacity of typical commercially available devices into account. Using power consumption data for these devices, they derive the energy efficiency of the whole network in terms of the power consumption over the overall access bit rate. They extrapolate this figure for increasing access bit rates and technological advances in terms of annual gross efficiency gains of electronic components. They find that, due to access-rate independent consumption of PONs, the energy consumption of routers in the core network becomes predominant with increasing access rate and a presumed proportional core capacity increase. As a remedy, they propose optical bypassing of core IP routers.

Van Heddeghem *et al.* [14] consider network-level power modeling from a different angle. Their model derives the power consumption of one multi-layer network domain from network characteristics like the number and average volume of traffic demands and the network diameter in terms of the average hop count in different layers. Further input values are factors representing external overhead (e. g. for facility cooling) and resource overdimensioning for protection.

Finally, they assume characteristic power consumption values per traffic rate for different technologies and device categories. The purpose of this model is to estimate the power consumption of large networks without dimensioning network resources, which is the prerequisite for deriving the total consumption from equipment-level power models like the one proposed in the same publication.

3.3.2 Device Power Consumption Measurements and Models

Researchers have taken different approaches to analyze and model the power consumption of core network devices. These approaches coarsely subdivide into the estimation of the consumption of state-of-the-art devices, performed by measurement or structural analysis, and projections for future energy-aware network components. The latter range from the assumption of abstract energy profiles to models applying different dynamic power-saving methods to the architectural components of state-of-the-art network nodes.

After describing a measurement approach in section 3.3.2.1, we survey abstract power models in section 3.3.2.2. Section 3.3.2.3 then presents device-structure based static power models, and section 3.3.2.4 finally outlines a related dynamic one.

3.3.2.1 Power Consumption Measurement

Chabarek *et al.* [65] performed measurements of the gross power intake of two Cisco router models with varying configurations concerning installed line cards and activated software functions under varying packet traffic load. Their findings include that the base nodes[7] account for more than half of the maximum power consumption, the traffic load influences power consumption by up to 10 %, and different software configurations (regarding monitoring, access control features, and routing table size) do not have any effect.

3.3.2.2 Abstract Power Models

The observation that the power consumption of current network nodes is largely insensitive to the traffic load led researchers to hypothesize about future improvements in this regard. To investigate a green traffic engineering scheme, Puype *et al.* [76] differentiate architectural and technological improvements. Assuming the power consumption to increase linearly with the load from an idle power value to the maximum consumption, they project to reduce both the idle and the maximum power by the same offset thanks to architectural improvements. Alternatively, they scale both power values proportionally for increasing efficiency of CMOS technology.

For a green routing study, Cardona Restrepo *et al.* [75] additionally consider non-linear energy profiles inspired by different device types and power-saving techniques. These profiles include on-off behavior, i. e. constant power consumption for non-zero load and zero consumption if the component can be deactivated, applicable e. g. for TXPs. Logarithmic profiles result from

[7] Term frequently used in the context of resource modeling, comprises chassis with control, cooling and power supply as well as the switching fabric, see also section 2.1.3.1.

hibernation modes: the authors differentiate a log10 profile for pure hibernation and a log100 profile resulting from the combination with on-off behavior. DVFS yields cubic energy profiles, and certain switch architectures finally exhibit a linear behavior. The authors consider the application of these profiles to whole subnetworks, nodes, or components of nodes.

Huang *et al.* [67] derive their flow-based power model from a slightly more detailed model of a multi-layer network node. Inspired by metrics of the Energy Consumption Rating (ECR) Initiative, they assume the power consumption of input, processing, and output of data at each layer to be proportional to the traffic load. By separating input, processing, and output, they can express the power consumption for switching a traffic flow at a given layer by the sum of processing at this layer and the input and output values of this and all lower layers (which have to terminate the connections serving the considered traffic flow).

In the same publication [67], the authors propose an interface-based power model, which more realistically reflects the behavior of state-of-the-art devices[8]. For each layer, this model gives static power consumptions for the base node and the active line cards and includes a linear traffic-dependent term per line card, which is stated to be negligibly small compared to the static consumption. In addition, the model accounts for a static layer-independent power consumption of a network node location.

A number of publications on energy-efficient network operation by deactivation of links and nodes assume similar models. On-off profiles for links (or pairs of ports or line cards, respectively) and nodes are used e. g. in the single-layer network studies of [65, 66, 70, 71]. Accounting for link rate adaptation in discrete steps, Vasić and Kostić [73] extend this type of models by rate-dependent consumption figures for links. In a multi-layer network scenario, Yetginer and Rouskas [63] apply a power model similar to the profiles in [76] to optical circuits. Shen and Tucker [64] consider the consumption of optical amplifiers on active fiber links in addition to components in nodes.

Discussing a single-layer networking scenario, Kist and Aldraho [74] enhance the energy-savings potential of network reconfiguration by introducing *bridged* operation modes for nodes. While nodes receiving traffic generally need to remain active and consume considerable quantities of power, a node in bridged mode can forward traffic between pairs of electronic interfaces without processing, driving its power consumption down to 10 % of normal operation. This is particularly beneficial to transfer add/drop traffic to/from an active neighbor node, but may also be used for transit traffic within the network.

3.3.2.3 Device Structure Based Power Models

To provide input to their green IP-over-WDM network design study, Rizzelli *et al.* [10] construct a power model based on the structure and power consumption of several commercially available network devices. In the IP layer, they assume router chassis (corresponding to base nodes) housing up to 32 line cards featuring a 10 Gbps short-reach interface each. For increased capacity, up to nine chassis are interconnectable by a fabric card chassis, and the use of several such chassis

[8] This holds true if the model is applied for the consumption of the installed components – in contrast to dynamic deactivation of resources not supported by current equipment.

further enhances the capacity. Power consumption figures for these components are taken from product specifications and technical reports. The consumption of an empty router chassis is computed by subtracting the consumption of the line cards from the figure for a fully equipped router. A node in the optical layer consists of a master rack housing OXC, multiplexers, demultiplexers, and optical amplifiers, and multiple slave racks housing TXPs in several shelves. The model differentiates TXPs for several line rates (10 Gbps, 40 Gbps, 100 Gbps) and modulation formats with varying power and space requirements. In addition, it comprises a flexible-rate TXP supporting 25 Gbps, 50 Gbps, 75 Gbps and 100 Gbps derived from the 100 Gbps fixed-rate TXP. It further derives properties of different 3R regenerators from the respective TXPs. The model finally applies power consumption values for slave racks, shelves, and per wavelength switched in an OXC.

Taking essentially the same modeling approach, van Heddeghem *et al.* [14] build on the expertise of and partially confidential information from several equipment manufacturers in the STRONGEST [106] and TREND [107] projects to create a comprehensive, vendor-independent reference model for the structure and power consumption of devices of four layer networks: IP/MPLS, Ethernet, OTN, and WDM.

Regarding the IP layer, the model structurally differs from the one by Rizzelli *et al.* in dividing line cards into slot cards performing packet processing and port cards providing (optical) interfaces. This facilitates the specification of a variety of interface configurations, as port cards can host several interfaces of the same technology and line rate, where line rates vary from 10 Gbps to 100 Gbps. Slot cards have a capacity of 40 Gbps or 140 Gbps and may feature several port cards of lower aggregate line rate. Unlike in the previously reported model and the measurements of [65], slot cards and port cards account for approximately 75 % of the power consumption of a fully equipped router chassis in this model. In the WDM layer, the model omits chassis. Instead, it features power consumption values for TXPs of line rates ranging from 2.5 Gbps to 1 Tbps. Values for 100 Gbps and above are projections. 3R regenerators are modeled as two TXPs operated back-to-back. The model further defines several fiber amplifiers for different intermediate distances. For WDM line systems, ROADMs and OXCs, it differentiates between devices supporting 40 and 80 channels per fiber, respectively. The power consumption of ROADMs further varies with the share of droppable wavelengths and that of OXCs with the nodal degree and the number of local drop ports. For the intermediate layers, the publication gives power consumption values for ports supporting a range of line rates, including projections for not yet available configurations.

3.3.2.4 Dynamic Power Model

While detailed reference models representing state-of-the-art equipment allow precise studies on the green design and static dimensioning of multi-layer networks, an according model for the dynamic operation of network resources has been missing. To fill this gap, Bauknecht and the author of this monograph derived a dynamic power model for IP-over-WDM network devices [15]. We start from a breakdown of the static power consumption to network node elements similar to the model of van Heddeghem *et al.* [14]. Basing on documents by manufacturers and researchers, we further decompose these consumption values and attribute the according shares to building blocks of these elements. For instance, we distribute the consumption of line cards among network processors, memory, and ASICs, and that of chassis among chassis control, power supply, and cooling.

For these building blocks, we make realistic assumptions on applicable power saving schemes, from simple deactivation of resources if unused over partial deactivation (e. g. of memory and cores of multi-core network processors) to highly dynamic adaptation by DVFS. Taking into account the time scales on which the respective mechanisms work, we then determine the feasible operation state of the components depending on the essential network-level operation parameters: the set of active optical circuits with their respective bit rate and the electronically processed traffic volume. For instance, we assume the processing capacity of network processors to precisely match the packet traffic load thanks to fast chip-level adaptation mechanisms like DVFS. In contrast, we scale the amount of active memory serving as packet buffers with the capacity of active ports of a line card, since memory cannot be reactivated on the time scale of arriving packets. Regarding the on-off operation of larger node elements, we assume that we can deactivate any module whose installation is optional, i. e. we allow deactivation of TRXs, line cards, and chassis.

While our model includes all relevant components of the optical layer including racks (with power supply and cooling overhead), we identified only limited options for load-adaptive operation of optical resources. Specifically, TXPs and their chassis allow deactivation. Provided a dimensioning of installed resources, the model allows expressing the power consumption of a network node as a function of the circuits it terminates, the hierarchical structure of resources required for this (i. e. line cards and chassis housing the respective TXPs and port cards), and the total traffic volume it processes electronically.

An observation common to all power models derived from state-of-the-art equipment is that devices terminating optical circuits or performing electronic packet processing, in particular in the IP layer, exhibit a significantly larger energy consumption than devices for optical switching and amplification. Accordingly, a number of green networking publications specifically target energy savings in the upper layer, often disregarding the power consumption of purely optical devices, e. g. [63, 73, 76]. This monograph takes the same approach.

3.3.3 Load-Dependent Network Operation

In this section, we survey research contributions targeting energy savings by adapting the network configuration to the traffic load. We include publications investigating energy-efficient configurations for a single demand matrix, i. e. capacitated network design problems, as well as contributions on actual network reconfiguration for evolving traffic. Both fields are relevant to network reconfiguration since methods proposed for the former are repetitively applicable in the latter scenario. We primarily classify the publications by the extent of the considered network reconfiguration, and we use the options assumed for resource adaptation as the secondary classification criterion.

The most restrictive form of dynamic network operation only adapts the state of resources to their load without any change to the network configuration in terms of demand routing and topology, with the possible exception of deactivating idle links. If the resource adaptation only consists in deactivating nodes or links, energy savings are only possible if a number of demands reduce to zero and free resources of traffic. The *fixed upper, fixed lower* (FUFL) scheme of [21] follows this principle for the deactivation of optical circuits, but allows the concentration of traffic onto parallel circuits. The non-rerouting variant of the multi-hour planning scheme of [78], which aims at deactivating both line cards and chassis, is less restrictive: when consecutively deriving network settings for lower-load situations from the temporally adjacent higher-load situation, it allows redistributing the traffic flow of demands undergoing multi-path routing among the candidate paths.

The majority of dynamic operation schemes adapt the configuration of a single layer network, i. e. they allow the rerouting of traffic and the modification of the operation state of nodes or node components and existing links. With one exception, the respective publications either assume single-layer networks or address the upper, packet-switched layer of multi-layer networks. The next two sub-sections cover publications assuming a simple on-off behavior for the resources and more sophisticated power scaling schemes, respectively. Section 3.3.3.3 finally addresses contributions proposing reconfiguration in several network layers, which enables the addition of links in the virtual topology.

3.3.3.1 Resource Deactivation

A first set of publication assumes the deactivation of links, or equivalently of interfaces or line cards (featuring a single interface), as the only possibility of resource adaptation. The *dynamic upper, fixed lower* (DUFL) scheme of [21] and the case study in [67] address the respective reconfiguration problem in a centralized way by linear programming and greedy heuristics, respectively. The *green distributed algorithm* of [80] lets nodes decide locally on the deactivation of lightly loaded incident links and reroutes traffic in the remaining topology. Deactivated links are reactivated gradually upon high network load and instantaneously upon fault notification. Conversely, the distributed energy-aware traffic engineering scheme of [81] applies load-dependent link weights for the routing or re-routing of LSPs in order to divert traffic from currently inactive or lowly-loaded links, which are deactivated when unused.

The next degree of freedom in resource adaptation assumed by a number of publications is the deactivation of nodes if all adjacent links are inactive. In case of single-chassis nodes, this is equivalent to the deactivation of the chassis in addition to deactivating line cards. The measurements of [65] yield such a model, and the publication solves the resulting capacitated design problem by linear programming. The group of late Professor Neri addresses a similar design problem by a simple heuristic [70], which sequentially tests whether the deactivation of resources results in the violation of constraints, and by linear programming [71]. In addition, they apply the same types of solution methods to a related problem in the WDM layer [66]: this problem aims at saving energy by deactivating fiber amplifiers and OXCs, which is possible if no optical circuit traverses the respective fiber link or node. In the IP/MPLS layer, *green traffic engineering* according to [69] exploits more fine-grained resource adaptation similar to deactivating links and nodes: deactivating ports and line cards, which may have several ports. Unlike the aforementioned publications, which explicitly determine traffic routes in a centralized way, Amaldi *et al.* [77] spend considerable effort on centrally determining a link weight assignment which frees links and nodes of traffic under shortest-path routing with ECMP in order to prepare for resource deactivation.

The minimization of the numbers of active line cards and chassis is likewise the objective of multi-hour network design methods. Addis *et al.* [79] thereby use the global view of the diurnal traffic evolution to bound the number of line card activations per link and day in order to constrain the reconfiguration effort. In addition, they limit the state changes of chassis by accounting for the transient power consumption during chassis activation. Zhang *et al.* [78] take a different approach to limit the extent of reconfiguration in their rerouting-enabled multi-hour design scheme. Starting from a configuration optimized for the peak traffic period, they consecutively determine subsets of the active circuits, OXCs, line cards, and chassis required in adjacent reconfiguration periods of next-lower traffic demands by linear programming. Consequently, each reconfiguration step consists in either the activation or the deactivation of resources, but not both, which limits the number of resource state modifications over the day. The scheme does neither constrain the rerouting of traffic demands, nor – curiously – RWA for persistent optical circuits.

Kist and Aldraho [74] propose the concept of *bridged* operation modes for network nodes, in which the nodes forward all traffic between specified interfaces without processing it, as an alternative to completely switching off nodes along with links in order to save energy. The publication assumes that a small share of the power consumption of active nodes and links depends on the actual traffic load (10 % and 20 %, respectively), and a node in *bridged* mode incurs 10 % of its maximum power consumption. The authors solve the resulting capacitated design problem by linear programming.

3.3.3.2 Power Scaling

Analogously to the work on resource deactivation, some publications consider capacity and power scaling in links only. In the elastic scenario, the reconfiguration study of [10] adjusts the line rate of optical circuits between 25 Gbps, 50 Gbps, 75 Gbps, and 100 Gbps. This in turn enables omitting 3R regenerators, and avoiding their power consumption, due to extended optical reach at lower line rates. By deactivating *cables in bundled links*, Fisher *et al.* [68] essentially vary the capacity of network links in discrete steps, yielding a proportional reduction in power consumption. This form of capacity adaptation is similar to the activation and deactivation of servers in data centers depending on the workload. In [108], Kühn and Mashaly develop a sophisticated analytical model for this problem, which controls the server state by hysteresis. This model could also be applied to network links provided the explicit signaling of demand arrivals according to a statistical model.

Another set of publications additionally considers the dynamic operation of nodes. The distributed *energy-aware traffic engineering* scheme by Vasić and Kostić [73] assumes link rates adjustable in several steps with accordingly scaling power consumption. In addition, it leverages sleep modes of line cards, which reduce the power consumption attributed to unused links to zero. Router chassis without active line cards are likewise put to sleep. Bianzino *et al.* [72] apply one type of energy profile to both links and nodes. Their *idleEnergy* profile assumes a base power consumption for unused active components and a linear increase with the load up to the maximum power. Deactivated idle components do not consume any power. The authors also investigate two extremal variants of this profile: load-agnostic consumption with the base power equal the maximum, and fully proportional with a base power of zero. For a worst-case scenario for energy savings where add/drop traffic prevents the deactivation of any node, the paper investigates the benefit of energy-aware vs. conventional routing, assuming at most 15 % of the maximum power consumption to be load-dependent in the idleEnergy case. With conventional routing, the assumption of load-dependent energy profiles yields energy savings of 15 % for idleEnergy and 96 % for fully proportional over a load-agnostic profile. In contrast, the additional savings by energy-aware routing are negligible (0.2 % and 8.3 % respectively for these energy profiles).

Cardona Restrepo *et al.* [75] assume a different class of network components to exhibit load-dependent power consumption. Their *energy-profile aware routing* scheme applies energy profiles to nodes and defines the respective load as the amount of traffic the node processes or as the maximum traffic on its incident links. Similarly to [72], the evaluation shows that assuming shortest-path routing and energy profiles which scale with the load[9], in particular convex ones, yields significantly larger energy savings than additionally using energy-profile aware routing.

3.3.3.3 Multi-Layer Network Reconfiguration

A small set of publications addresses green network reconfiguration in multiple layers. The reported reconfiguration schemes are characterized by their ability to alter the virtual topology by adding links not foreseen in a base configuration, and they usually include RWA for the circuits of these links. In accordance with the dominance of the power consumption of equipment terminating optical circuits and performing electronic packet processing, the schemes aim at minimizing the power consumption of the upper layer and disregard contributions of optical switching and amplification.

Idzikowski *et al.* [21] evaluate maximum energy savings achievable by such reconfiguration using measurement-based traffic traces. Their *dynamic upper, dynamic lower* (DUDL) scheme solves the capacitated design problem minimizing the power consumption of active line cards terminating individual bi-directional optical circuits independently for each traffic matrix by mathematical programming. The problem including RWA proved too complex to be solved exactly, leaving gaps between primal and dual bound of up to 30 % upon reaching a solution time limit for the largest considered topology with 22 nodes. In terms of energy consumption, the primal bounds are on par with the solutions of the investigated single-layer variant DUFL.

Together with another set of co-authors, Idzikowski proposes a simple heuristic for a related multi-layer reconfiguration problem in [83]. In addition to line cards, this publication targets energy savings by deactivating line card chassis and fabric card chassis. It determines the respective numbers of active chassis as the minimum required to accommodate the active line cards of a node, disregarding the history of circuit establishments and teardowns which may in practice result in fragmentation of the chassis occupation. The heuristic tries to set up circuits to overcome network segmentation, to offload highly loaded links, and it tears lowly loaded circuits down if this does not result in overload elsewhere. By first allocating resources for new circuits, it respects the hitless reconfiguration principle. Demands are routed on the shortest path in the resulting topology. In [84], mostly the same authors compare their heuristic to two different methods: Heuristic optimization of the virtual topology by a genetic algorithm, which

[9] I. e. energy profiles which exhibit more load-dependence than on-off behavior.

performs comparably, and a simple heuristic aiming at deactivating complete virtual links (instead of individual circuits), which performs worse due to its more coarse-grained control of the network configuration. To limit reconfiguration, the cost function of the optimization includes a linear term for the traffic volume newly routed onto a virtual link. All these methods route traffic demands on their shortest path in the virtual topology and disregard RWA, i. e. the procedure for circuit realization of the former publication is omitted. Disregarding chassis, this study only considers the power consumption of line cards.

4 Centralized One-Step Reconfiguration Method

This chapter describes our centralized one-step reconfiguration method, which aims at minimizing power consumption by periodically adapting the network configuration to the traffic load. Like a number of other researchers [21, 54, 83, 84], we intend to adapt to the traffic on the time scale of diurnal load profiles, assuming reconfiguration intervals in the order of 15 to 30 minutes. This time scale enables using centralized methods to determine the new network configuration [71]. In exploiting all degrees of freedom of a two-layer network scenario, i. e. demand rerouting and virtual topology adaptation, our proposal resembles the work by Idzikowski, Bonetto, *et al.* [21, 83, 84]. To the best of our knowledge, we are alone in applying optimization methods to a green multi-layer network reconfiguration problem, i. e. a problem taking into account constraints of a previous configuration.

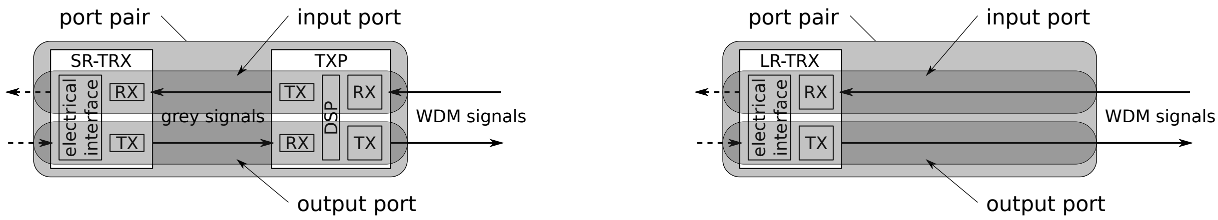
In section 4.1, we describe our assumptions on technological and operational constraints. Section 4.2 then states the requirements we impose on each new configuration, and we formulate the according reconfiguration problem in section 4.3. We finally detail two solution methods for this problem in section 4.4.

4.1 Technological and Operational Assumptions

While our optimization problem formulation and the according solution methods are in principle applicable to any two-layer network architecture featuring connection orientation in the upper layer and circuit switching in the lower one, we specifically assume the IP/MPLS over WDM network architecture in order to derive realistic parameters and constraints. In the following, we discuss our assumptions on capabilities of optical switches, the power model for both network layers, and temporal constraints on network reconfiguration which imply a one-step reconfiguration principle.

4.1.1 Optical Switching Capabilities

Given continuous improvements of the technology and architecture of optical nodes, we assume minimal restrictions in optical switching. The connection of interfaces of the upper-layer node, i. e. TXPs or long-reach TRXs, to the WDM node has *directionless*, *colorless*, and *contentionless* properties. I. e., any such interface can transmit and receive signals on any wavelength from



(a) short-reach TRX and TXP (b) long-reach TRX

Figure 4.1: Definition of port pairs and ports for different circuit termination options

and to any neighbor node regardless of the wavelengths used by other TXPs/TRXs. In addition, OXCs are non-blocking and feature full wavelength conversion capability. That is, a circuit can be switched from any wavelength on any incoming fiber to any wavelength on any outgoing fiber regardless of the resource occupation by other circuits.

We assume wavelength conversion to be accomplished in the optical domain. Accordingly, we do not need to account for electro-optical signal conversion in nodes forwarding optical circuits. We further assume that the optical wavelength conversion does not influence the optical reach of the signal.

4.1.2 Dynamic Resource and Power Model

We base our network reconfiguration problem on a reduced version of the dynamic resource operation and power model of section 3.3.2 which Bauknecht and the author of this monograph developed in [15]. The reduction aims at simplifying the problem formulation while preserving the essential properties: We disregard static power consumption as well as the contributions of tributary interfaces and add/drop traffic, since they are independent of the network configuration. We further assume that the optical switching equipment and the chassis of the optical nodes exhibit static, load-independent power consumption. The implication of this restriction is limited due to the small contribution of these components to the overall power consumption [14, 15]. For the same reason, we neglect the consumption of optical fiber amplifiers. With these assumptions, we are able to express the power consumption of a network node as a function of the amount of electrically switched transit traffic, the number of active ports terminating optical circuits, and the hierarchical components, i. e. line cards and line card chassis, required to accommodate these ports in the IP/MPLS router. Except for the consideration of the hierarchical components, our power model assumptions equal those made in [63] for static design.

As explained in section 2.1.3.1, optical circuits may be terminated either by TXPs, which are connected to short-reach TRXs on the line card, or directly by long-reach TRXs. TXPs and TRXs are bi-directional, i. e. they feature one input port and one output port to terminate two circuits in opposite directions. We assume that we can activate and deactivate the components serving one of the directions independently of the others. To abstract from whether a TXP or a long-reach TRX is used, we refer to the equipment terminating one end of an optical circuit by line card *port*. The two ports of opposite directions realized by one TXP or TRX, respectively, form a *port pair*. Figure 4.1 illustrates the representation of circuit terminating equipment by

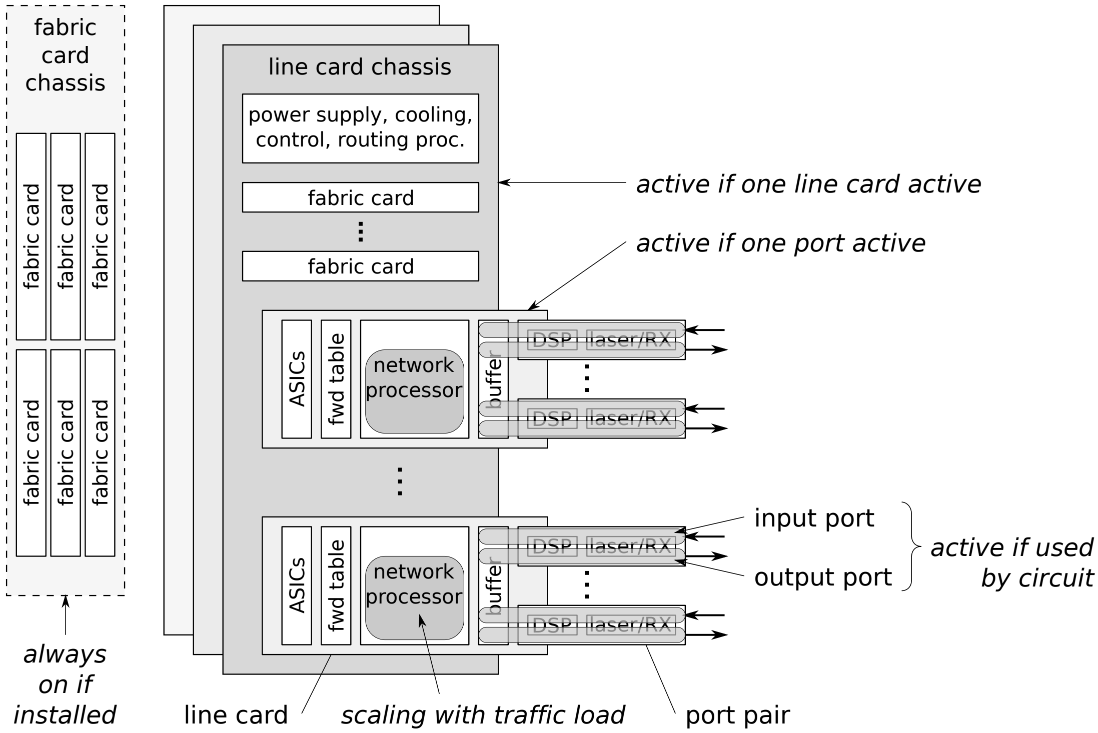


Figure 4.2: Dynamic resource operation model for a network node

port pairs and ports in the cases of a TXP with a short-reach TRX and of a long-reach TRX, respectively.

Building on such port pairs, Figure 4.2 depicts the relevant elements of the dynamic resource operation model. In addition to the structure of the hardware elements, the figure indicates the dynamic operation regime: Shaded boxes with rounded corners represent components directly dependent on the network configuration, i. e. ports terminating circuits and packet processing capacity scaling with the traffic load. Hierarchical components required to support active ports are given by rectangular shapes.

One line card features up to N_{PP} port pairs. One part of the memory on the line card serves as packet buffers, which are generally dimensioned proportionally to the interface capacity following the bandwidth-delay-product rule [109]. We accordingly assume shares of this part of the memory to be activated and deactivated with the capacity of the active ports of the line card. We therefore define the power consumption P_{P} per active port, which is accounted once at each end of an optical circuit, to comprise the proportional share of the memory power consumption in addition to the power consumed by the unidirectional components of TXP and TRX. We only consider ports of a uniform bit-rate capacity, and the ports consume a constant amount of power when active. This reflects the continuous operation of lasers and coding and modulation DSPs in the TXPs and TRXs.

Network processors on line cards are in charge of processing packets. We assume that a significant share of their power consumption scales with the workload, since the many-core architecture of these processors paves the way for a fast and fine-grained adaptation by frequency and voltage scaling along with sleep modes for some of the cores. Approximating the scaling behavior by a linear relation, we denote the power consumed for electrically switching one unit of traffic in one node by P_{T}.

The remaining components of the line card are modeled with a constant power consumption of P_{LC}. This value particularly accounts for the ASICs of the fabric interface, the remaining part of the memory hosting the forwarding table, and the static part of the power consumptions of NPs. We assume that these components are required as soon as one port of the line card is active. Otherwise the line card is switched off and consumes no power.

One line card chassis hosts up to N_{LC} line cards, which it interconnects by means of several fabric cards. In addition, the chassis provides power supply and cooling to the line cards and features control and routing processors. The efficiency of the power supply is assumed to be load-independent and factored into all component power values, which in turn give the respective gross consumption. Likewise, the power values of the major consumers, which account for most of the heat dissipation, are increased to compensate for the respective cooling effort. We assume constant power consumption for the remaining line card chassis components of P_{LCC}, which applies if at least one line card in the chassis is active.

For increased node capacity, several line card chassis may be interconnected by a fabric card chassis featuring further fabric cards. Disregarding the possibility to scale the number of active fabric cards with the number of active line card chassis, we consider the power consumption of fabric card chassis constant and omit it in the dynamic power model. When applying the model for network resource dimensioning (cf. section 5.1.2), we do however consider the static power consumption of these fabric cards and their chassis.

Our dynamic resource and power model allows expressing the power consumption of a network node as a function of the capacity of its hardware components and the dynamic configuration in terms of terminated circuits and processed packet traffic. By summing up the consumption of all nodes, we can use this model to compute the overall power consumption of a network for a given network configuration. We do so for designing and evaluating our network reconfiguration methods.

4.1.3 Reconfiguration Time Constraints

By principle, network reconfiguration is not feasible in zero time. In case of a centralized scheme for load-dependent reconfiguration, time is required for the signaling of network state information to the decision-making entity, for the computation of a new network configuration, for the signaling of the new configuration to the network nodes, and for the actual implementation of the configuration changes. While most currently deployed transport network equipment does not support dynamic reconfiguration, we can make reasonable assumptions on the technologically feasible scope and the according delay of the adaptation of the resources.

The previous section discusses our assumptions on the scope of reconfiguration. The reconfiguration time is dominated by a component not explicitly named in this section: Erbium doped fiber amplifiers (EDFA), which are widely deployed as line amplifiers, cannot immediately adapt their amplification power. Consequently, the sudden change of the signal strength of one wavelength in a fiber causes a step in the signal level of the remaining wavelengths, which impairs the transmission quality on these channels. This effect is exacerbated by a sequence of EDFAs, which is the common case on transport network links [110].

One remedy is to gradually adapt the signal strength of wavelengths during their establishment or teardown. We assume the required adaptation time to be in the order of several minutes. We further assume the reconfiguration of the remaining equipment to be feasible within this time budget. This includes booting newly activated line cards and line card chassis, respectively, and synchronizing their state, e. g. in terms of forwarding tables. We finally assume that the time required for rerouting traffic in the electrical layer and for signaling is negligible compared to the circuit adaptation time.

The assumed circuit reconfiguration times turn a reconfiguration scheme which reacts upon actual capacity shortages infeasible, since it could result in continued traffic blocking not tolerable in transport networks. Reconfiguration under these assumptions rather requires a forecast of the expected maximal traffic load, for which active resources are accordingly provisioned. To maximize energy savings, such a forecast likely takes observations of the current traffic into account and does not simply assume the traffic to follow stationary day and week profiles.

This monograph assumes that the necessary forecasts are available and does not discuss the problem of traffic prediction. Arguably, forecasting becomes more difficult or imprecise with increasing forecast horizon, which should therefore be kept to a minimum. On the other hand, the forecast has to cover the period in which the resulting configuration is active, the circuit adaptation time for reaching this configuration, and the time required to compute the respective configuration. These dependencies underlie the reconfiguration principle which we detail in the next section.

4.1.4 Periodic One-Step Reconfiguration Principle

We assume a strictly sequential, periodic network reconfiguration regime. Periodic reconfiguration, i. e. time-triggered reconfiguration in equidistant intervals, is commonly applied with centralized optimization approaches, e. g. in [21, 72]. It has the advantage of not requiring a metric or mechanism to detect when changes in the traffic load turn the current configuration sub-optimal. Conversely, the computation should indicate no need for reconfiguration if the load situation did not change significantly. In principle, subsequent reconfiguration intervals could overlap in order to adapt the network configuration to load changes in a timely manner despite the circuit adaptation time. However, the benefit of such an overlap is limited since new configurations would be constrained by previous settings. In addition, network operators are likely to prefer infrequent reconfiguration in transport networks, which they currently operate statically. We therefore opt for a strictly sequential procedure.

As discussed in section 3.2.3, many schemes allowing arbitrary reconfiguration of a network make use of a transition path, i. e. they move from one configuration to the next by a sequence of configuration changes. In this way, they can reuse resources freed in one step in a different setting in a subsequent step. Given the substantial circuit adaptation time, the limited traffic forecast horizon we assume disallows the use of a transition path. Following the paradigm of hitless reconfiguration [50], the next virtual topology configuration has rather to be reached in

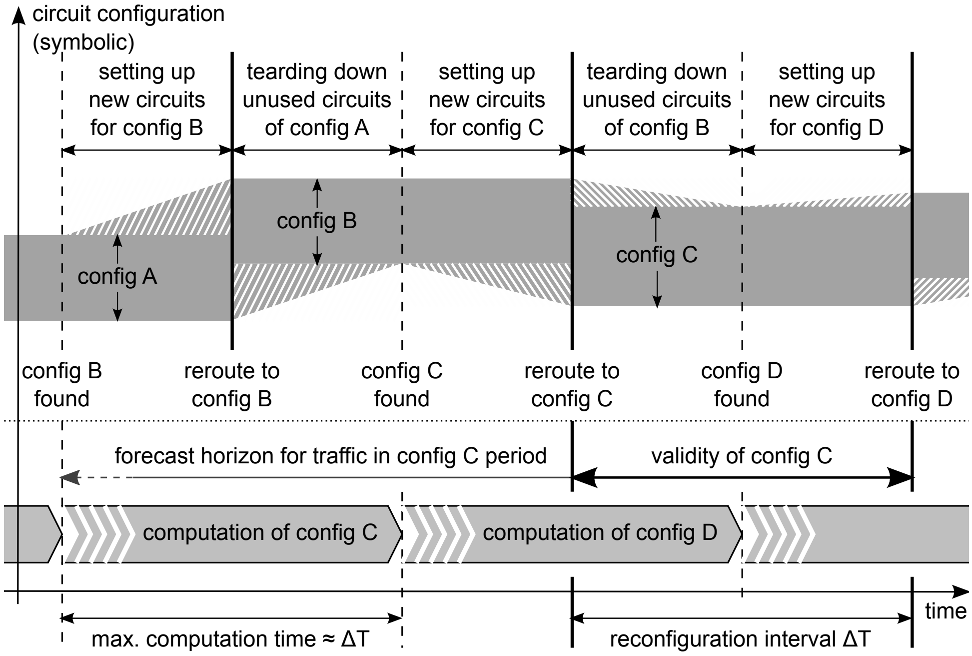


Figure 4.3: Time flow of one-step reconfiguration procedure

a single step. This *one-step reconfiguration* principle, which the author of this monograph first discussed in [111], implies an additional constraint on a new configuration: resources occupied by a circuit in the current configuration must not be used for a different circuit in the next configuration. Due to significantly shorter adaptation times, we do not impose any constraints on demand routing in subsequent configurations.

Figure 4.3 illustrates the time flow of the periodic one-step reconfiguration, where each configuration is valid for a period of ΔT. The upper part describes the actual network configuration: once the next configuration is determined, optical circuits required in addition to the active configuration are set up (cf. striped triangles). As discussed in the previous section, this requires a non-negligible amount of time. Upon completion, the traffic is rerouted as defined by the new configuration, freeing circuits unused in this configuration, which are then torn down slowly. We neglect the time required for rerouting, since it is small compared to the circuit adaptation time. Due to the sequential reconfiguration procedure, the reconfiguration interval ΔT is bounded below by the sum of circuit setup and teardown times.

The lower part of Figure 4.3 addresses the computation of new configurations. Computation can start as soon as the input values are available, i. e. the traffic forecast and the previous configuration. The latter limits the computation time to ΔT. To exploit this maximum time, traffic forecasts need to be available $1.5\Delta T$ before the respective configuration turns active, assuming that circuit establishment and teardown take half a reconfiguration interval. The corresponding total forecast horizon is then $2.5\Delta T$.

4.2 Requirements

The primary motivation of the reconfiguration scheme proposed and evaluated in this monograph is saving energy. Accordingly, it shall provide network configurations of minimal power consumption according to the model of section 4.1.2 for each reconfiguration interval, i. e. per predicted traffic matrix. In doing so, it needs to respect resource pre-occupation constraints implied by the one-step reconfiguration principle of section 4.1.4.

For several reasons, the power consumption of the new configuration has to be traded off against the necessary changes to the network. First, the resources allocated for a circuit consume energy during the gradual establishment and teardown of this circuit. Second, operators may want to avoid frequent changes of the operational state of resources to prevent accelerated aging. Finally, large-scale modifications of the configuration may be considered detrimental to the stability of network operation. Since the former two reasons do not concern traffic rerouting but only modifications of the virtual topology, which are also more heavy-weight than traffic rerouting regarding reconfiguration time, we opt to quantify the reconfiguration effort by the number of changed (i. e. established or torn down) circuits compared to the previous configuration. Hence, our scheme shall limit the number of circuit changes, while we do not constrain the rerouting of traffic.

Because of the resource pre-occupation constraints of one-step reconfiguration, traffic blocking can occur due to unfavorable previous configurations even if the traffic demand does not exceed the assumptions made for resource dimensioning. This is a price we have to pay for any scheme which deviates from static network operation without perfect knowledge of the development of traffic demands in a relevant future time period. However, traffic blocking could also occur with static network operation if the offered traffic exceeds the projections underlying resource dimensioning, in which case reconfiguration could mitigate traffic loss. Nevertheless, traffic blocking is hardly tolerable in transport networks due to QoS requirements. Therefore, our reconfiguration scheme shall prevent traffic blocking at any energetic cost if this is possible. If blocking is unavoidable, it shall minimize the amount of affected traffic. We only consider traffic lost due to the inability to accommodate the forecast traffic demands. We disregard traffic loss due to incorrect traffic forecasts, since prediction mechanisms are beyond the scope of this monograph.

4.3 Problem Definition

After discussing some restrictions of the scope, this section describes the one-step reconfiguration problem to be solved for each reconfiguration interval. While introducing all relevant notations and symbols, it does not provide a strict mathematical programming formulation (as given in section 4.4.2.2 for a subproblem). Instead, this section focuses on discussing and reasoning the technical considerations and constraints shaping our problem. The subsequent subsections address the available input parameters, define the components of a network configuration returned as solution, discuss the constraints, and detail the objective.

4.3.1 Delimitation of Scope

For a thorough investigation of the energy-aware network reconfiguration problem which we define in the following, we opt to exclude certain aspects and interrelated problems. A first such aspect is resilience, which is generally mandatory due to the high reliability requirements on transport networks. Following the traditional approach of protection, we would need to allocate backup resources for protection in each configuration. As outlined in section 3.2.2 and detailed in [93], however, restoration imposing fewer constraints than protection on network configurations is likewise imaginable. Since the selection of an appropriate resilience mechanism is beyond the scope of this monograph and different choices are likely to have varying impact on network reconfiguration, we do not provide for resilience in our reconfiguration problem. This exclusion is also made by a large number of publications on network design and reconfiguration, including most of those cited in chapter 3.

Network reconfiguration with hierarchical resources brings about fragmentation. For instance, the evolution of network configurations may result in partial utilization of the ports of several line cards or of line cards of several chassis. Consequently, more line cards or chassis, respectively, need to be powered on than minimally required to support the number of currently active circuits. Without complete knowledge of the future traffic evolution (and the according network configurations), it is impossible to determine an optimal allocation of line card ports. However, this allocation problem lends itself to the design of heuristics, which constitutes a research question going beyond the scope of this monograph. Since the allocation of line card ports may significantly impact the power consumption of the network, we opt to disregard the effect of sub-optimal utilization of hierarchical resources and assume the minimum number of line cards and chassis required to sustain the active circuits to be powered on. The same simplifying assumption is made regarding line card chassis and fabric card chassis, respectively, by the authors of [10, 83].

As discussed in section 3.2.4, a number of publications on network design and reconfiguration formulate QoS requirements regarding packet delay as objective or constraint. More precisely, they bound the propagation delay by limiting the virtual path length and reduce queuing delays to negligible levels by constraining the link utilization. Since the tolerable delay varies with the application and the admissible link utilization depends on stochastic traffic properties, the respective parameters are hard to quantify globally. We therefore chose not to include delay constraints in the problem formulation, but assume that the traffic forecasts represent peak values including sufficient headroom for statistical fluctuations. If a practical application requires constraining the propagation delay, a bound on the virtual path length could easily be included in the problem formulation and obeyed by the solution method of section 4.4.3.

4.3.2 Input Parameters

The input parameters describing the specific reconfiguration problem instance comprise

- the physical network along with the amount of installed resources,
- the maximal demand values between all pairs of different nodes in the time interval the new configuration applies to, and

- the active circuits of the previous configuration along with the resources they occupy.

These parameters are described in the following. Section 4.3.4 and section 4.3.5 introduce further physical and cost parameters, respectively, which are independent of the specific network scenario.

4.3.2.1 Dimensioned Physical Network

Analogously to the notation introduced in section 2.3.1, the physical network topology is specified as the graph

$$G_{\mathrm{p}} = \langle V, E_{\mathrm{p}} \rangle \tag{4.1}$$

Herein, V is the set of all nodes of the multi-layer network and E_{p} is the set of all directed fiber links:

$$E_{\mathrm{p}} \subseteq \{(m,n) \in V \times V | m \neq n\} \tag{4.2}$$

The geographical length of the physical link from node m to node n is given by l_{mn}. We further obtain the number of resources installed on each network component. For a directed physical link, it is the number of fibers

$$n_{\mathrm{F}\,mn} \in \mathbb{Z}_0^+ \qquad \forall\,(m,n) \in E_{\mathrm{p}} \tag{4.3}$$

Each fiber is able to carry N_λ optical channels of capacity B. For the network nodes, the relevant resources are the port pairs. Their number is given by

$$n_{\mathrm{PP}\,v} \in \mathbb{Z}_0^+ \qquad \forall\, v \in V \tag{4.4}$$

We assume the packet processing capacity of each node to be sufficient for the maximum traffic terminated by its port pairs. We further assume the minimum number of line cards to accommodate the respective port pairs to be installed, i. e.

$$n_{\mathrm{LC}\,v} = \left\lceil \frac{n_{\mathrm{PP}\,v}}{N_{\mathrm{PP}}} \right\rceil \qquad \forall\, v \in V \tag{4.5}$$

We likewise assume the minimum number of line card chassis to accommodate these line cards to be installed:

$$n_{\mathrm{LCC}\,v} = \left\lceil \frac{n_{\mathrm{LC}\,v}}{N_{\mathrm{LC}}} \right\rceil \qquad \forall\, v \in V \tag{4.6}$$

4.3.2.2 Demand Matrix

We assume traffic forecasting to yield one (peak) value for each node-to-node traffic demand in the interval the new configuration applies to. As proposed in section 2.3.1, we represent the individual demand values by d_{sd} and aggregate them into the demand matrix

$$\boldsymbol{D} = (d_{sd})\ ;\ s,d \in V \qquad \text{where} \quad d_{vv} = 0\ \ \forall\, v \in V \tag{4.7}$$

4.3.2.3 Previous Configuration

The relevant part of the previous configuration is the set of active optical circuits, since the circuits define the resource occupation and determine the reconfiguration cost (cf. section 4.3.4 and section 4.3.5.2, respectively). We do not consider the previous routing of demands, since it does neither imply relevant resource pre-occupation under our assumptions nor incur a cost for modification.

We define an optical circuit by the line card ports that terminate it and its route in the physical topology. The route is a sequence of physical links from the source node $i \in V$ to the target node $j \in V$ of the circuit, which we describe by a vector of the nodes it traverses:

$$\boldsymbol{R}_\mathrm{C} = [v_1, v_2, \ldots, v_{l+1}] \in V^{l+1} \qquad \text{such that } (v_x, v_{x+1}) \in E_\mathrm{p} \text{ for } x = 1, \ldots, l \,;\, v_1 = i \,;\, v_{l+1} = j \tag{4.8}$$

Herein, l is the length of the specific route in hops. We identify the ports by the index of the port pair they belong to. For the output port at source node i and the input port at target node j, respectively, the indices are

$$i_{\mathrm{P}\,i}, i_{\mathrm{P}\,j} \in \mathbb{Z}^+ \;;\quad 1 \le i_{\mathrm{P}\,i} \le n_{\mathrm{PP}\,i} \;;\quad 1 \le i_{\mathrm{P}\,j} \le n_{\mathrm{PP}\,j} \tag{4.9}$$

Therewith, we describe a circuit by the tuple

$$C = \left\langle i_{\mathrm{P}\,i}, i_{\mathrm{P}\,j}, \boldsymbol{R}_\mathrm{C} \right\rangle \tag{4.10}$$

As detailed in section 4.3.4, this information is sufficient to account for the pre-occupation constraints. We denominate the set of all active circuits of the previous configuration by P. For the sake of a more concise notation, we designate the subset of such circuits from node i to node j by

$$P_{ij} = \left\{ \left\langle i_{\mathrm{P}\,v_1}, i_{\mathrm{P}\,v_{l+1}}, [v_1, \ldots, v_{l+1}] \right\rangle \in P | v_1 = i \wedge v_{l+1} = j \right\} \tag{4.11}$$

4.3.3 Solution Structure

A solution of the network reconfiguration problem describes the new configuration in terms of active circuits and demand routing. We denominate the set of active unidirectional circuits by A, wherein each circuit is specified as defined in section 4.3.2.3. For the sake of a more concise notation, we define the set of active circuits from node i to node j analogously to Equation (4.11):

$$A_{ij} = \left\{ \left\langle i_{\mathrm{P}\,v_1}, i_{\mathrm{P}\,v_{l+1}}, [v_1, \ldots, v_{l+1}] \right\rangle \in A | v_1 = i \wedge v_{l+1} = j \right\} \tag{4.12}$$

In addition, we define the set of virtual links realized by the active circuits:

$$E_\mathrm{v} = \left\{ (i,j) \in V \times V | A_{ij} \neq \emptyset \right\} \tag{4.13}$$

We base the description of the routing of demands on the notion of routed demands. A routed demand D_r consists of a traffic volume t and the path $\boldsymbol{R}_\mathrm{D}$ it takes in the virtual topology:

$$D_\mathrm{r} = \langle t, \boldsymbol{R}_\mathrm{D} \rangle \tag{4.14}$$

The path is given as a vector of the nodes it traverses:

$$\boldsymbol{R}_\mathrm{D} = [v_1, v_2, \ldots, v_{l+1}] \in V^{l+1} \qquad \text{such that } A_{v_x v_{x+1}} \neq \emptyset \text{ for } x = 1, \ldots, l \tag{4.15}$$

The condition in Equation (4.15) enforces that all virtual links of the path are realized by circuits. We allow the traffic flows between a given pair of nodes to be split and routed along several paths in the virtual topology. Accordingly, the solution defines a set R_{sd} of routed demands for each pair of source node s and destination node d with predicted peak demand $d_{sd} > 0$. Finally, we denote the set of all routed demand sets by

$$R = \{R_{sd} \mid d_{sd} > 0\} \tag{4.16}$$

4.3.4 Constraints

The constraints of the network reconfiguration problem enforce that all demands are transported – as far as feasible given limited resources – and that the limitation and pre-occupation of resources are respected. In addition, constraints limit the circuit length to the maximum transparent optical reach. Equation (4.17) verifies that the sum of the routed demands corresponds to the predicted demand value for all node pairs:

$$\sum_{\langle t, \boldsymbol{R}_\mathrm{D} \rangle \in R_{sd}} t = d_{sd} \quad \forall\, (s,d) \in V \times V \setminus \{(v,v) | v \in V\} \tag{4.17}$$

For the sake of concise notation, we define the volume of traffic to be transported by the virtual link from node i to node j:

$$t_{ij} = \sum_{R_{sd} \in R} \sum_{\substack{\langle t, [v_1, \ldots, v_{l+1}] \rangle \in R_{sd} \text{ where} \\ \exists x \in \{1, \ldots, l\} : (v_x, v_{x+1}) = (i,j)}} t \qquad \forall\, (i,j) \in E_\mathrm{v} \tag{4.18}$$

Equation (4.19) postulates a sufficient number of active circuits of capacity B to accommodate the traffic on each virtual link. Since this condition cannot be guaranteed due to limited resources, it may be lifted by setting a blocking indication $b_{ij} \in \{0,1\}$ for the respective link. Γ is a large constant greater than any potentially blocked traffic volume. Blocking is further addressed in regard to the objective in section 4.3.5.3. The formulation assumes that both demand routes and circuit routes are loop-free, which is a reasonable assumption supported by the solution methods of section 4.4.

$$\left| A_{ij} \right| B + b_{ij} \Gamma \geq t_{ij} \quad \forall\, (i,j) \in E_\mathrm{v} \tag{4.19}$$

The next set of constraints ensures that installed resources are assigned to at most one previously existing, persisting, or newly established circuit. Since persistent circuits have identical representations in the sets of previous (P) and active (A) circuits, we obtain the set of all circuits in any of these three states as the union $P \cup A$ of these two sets. The line card ports terminating optical circuits are uniquely identified by their index in the circuit description. Accordingly, we postulate in Equation (4.20) and Equation (4.21) that at each node, each output port and input port, respectively, may be assigned at most once. In conjunction with Equation (4.9), this implies that a node terminates at most as many circuits as it features ports.

$$\left|\left\{\left\langle i_{\mathrm{P}\,v_1}, i_{\mathrm{P}\,v_{l+1}}, [v_1, \ldots, v_{l+1}]\right\rangle \in P \cup A \mid v_1 = i \wedge i_{\mathrm{P}\,v_1} = x\right\}\right| \leq 1 \quad \forall\, i \in V\ ,\ x = 1, \ldots, n_{\mathrm{PP}\,i} \tag{4.20}$$

$$\left|\left\{\left\langle i_{\mathrm{P}\,v_1}, i_{\mathrm{P}\,v_{l+1}}, [v_1, \ldots, v_{l+1}]\right\rangle \in P \cup A \mid v_{l+1} = j \wedge i_{\mathrm{P}\,v_{l+1}} = x\right\}\right| \leq 1 \quad \forall\, j \in V\ ,\ x = 1, \ldots, n_{\mathrm{PP}\,j} \tag{4.21}$$

In accordance with current constraints for the static resource configuration, we require the two ports of one port pair to connect to the opposite ports of *one* port pair of another node if both ports are active. Equation (4.22) and Equation (4.23) formalize this constraint.

$$\begin{aligned} &\textbf{if } \exists \left\langle i_{\mathrm{P}\,\bar{v}_1}, i_{\mathrm{P}\,i}, [\bar{v}_1, \ldots, \bar{v}_{\bar{l}}, i]\right\rangle \in P \cup A \textbf{ then } \bar{v}_1 = j \wedge i_{\mathrm{P}\,\bar{v}_1} = i_{\mathrm{P}\,j} \\ &\qquad \forall \left\langle i_{\mathrm{P}\,i}, i_{\mathrm{P}\,j}, [i, v_2, \ldots, v_l, j]\right\rangle \in P \cup A \end{aligned} \tag{4.22}$$

$$\begin{aligned} &\textbf{if } \exists \left\langle i_{\mathrm{P}\,j}, i_{\mathrm{P}\,\bar{v}_{\bar{l}+1}}, \left[j, \bar{v}_2, \ldots, \bar{v}_{\bar{l}+1}\right]\right\rangle \in P \cup A \textbf{ then } \bar{v}_{\bar{l}+1} = i \wedge i_{\mathrm{P}\,\bar{v}_{\bar{l}+1}} = i_{\mathrm{P}\,i} \\ &\qquad \forall \left\langle i_{\mathrm{P}\,i}, i_{\mathrm{P}\,j}, [i, v_2, \ldots, v_l, j]\right\rangle \in P \cup A \end{aligned} \tag{4.23}$$

In the optical layer, circuit routing needs to respect the fiber capacity on all physical links. Equation (4.24) formalizes this for a constant number of channels N_λ per installed fiber under the assumption of loop-free routes. Equation (4.25) finally bounds the length of multi-hop circuits to the maximum transparent optical reach L. We do not apply a length limit to circuits spanning a single physical link.

$$\underbrace{\left|\left\{\left\langle i_{\mathrm{P}\,v_1}, i_{\mathrm{P}\,v_{l+1}}, [v_1, \ldots, v_{l+1}]\right\rangle \in P \cup A \middle| \exists x \in \{1, \ldots, l\} : (v_x, v_{x+1}) = (m,n)\right\}\right|}_{\text{number of circuits using link } (m,n)} \leq N_\lambda \cdot n_{\mathrm{F}\,mn} \quad \forall\, (m,n) \in E_\mathrm{p} \tag{4.24}$$

$$\sum_{x=1}^{l} l_{v_x v_{x+1}} \leq L \qquad \forall \left\langle i_{\mathrm{P}\,v_1}, i_{\mathrm{P}\,v_{l+1}}, [v_1, \ldots, v_{l+1}]\right\rangle \in A : l > 1 \tag{4.25}$$

4.3.5 Objective

The requirements on network reconfiguration in section 4.2 translate into three objectives: minimizing the power consumption of the network, limiting the reconfiguration effort, and avoiding (or minimizing) traffic blocking. We opt for jointly minimizing these three objectives by adding up respective terms in the cost function. The following subsections detail the respective terms and the resulting function.

The linear combination of power consumption and reconfiguration effort is justified since resources of circuits under modification likewise consume power. Tran and Killat [26] use a cost function resembling these two terms. However, their reconfiguration term only counts newly established circuits and the cost of the new configuration is expressed in terms of circuits and transit traffic, ignoring hierarchical node components. Idzikowski, Bonetto, *et al.* [83, 84] likewise define cost functions with an additive reconfiguration term, which however quantifies rerouted traffic. A linear combination of different objectives is also found in [11].

Since we intend to give priority to the objective of avoiding traffic blocking, intuition suggests to minimize the amount of lost traffic first and to address the remaining objectives by a second optimization problem featuring the minimal blocking value as a constraint. This corresponds to the two-step approach applied for a different set of objectives in [49]. In addition, the observations by Cinkler *et al.* [47] discourage from translating constraints into cost penalties. However, sequentially solving two optimization problem instances tends to be problematic for on-line computation, which implies a limited time budget to find the next configuration. We therefore propose a single optimization problem and add a large-valued blocking penalty term to the cost function. The limited extent of traffic blocking observed in section 5.5.4 indicates that this approach is viable under practical conditions.

4.3.5.1 Power Consumption

The power consumption of the new configuration is defined by the resources used for its circuits, the electrically switched traffic, and the respective power values from section 4.1.2. The total power consumption, i. e. energetic cost, of the network is thus obtained by Equation (4.26):

$$c_{\mathrm{P}} = P_{\mathrm{P}} \sum_{v \in V} a_{\mathrm{P}\,v} + P_{\mathrm{LC}} \sum_{v \in V} a_{\mathrm{LC}\,v} + P_{\mathrm{LCC}} \sum_{v \in V} a_{\mathrm{LCC}\,v} + P_{\mathrm{T}} \sum_{v \in V} t_{\mathrm{T}\,v} \tag{4.26}$$

Herein, $a_{\mathrm{P}\,v}$ is the number of active ports of node v required to terminate all outgoing and incoming circuits:

$$a_{\mathrm{P}\,v} = \sum_{\bar{v} \in V \setminus \{v\}} |A_{v\bar{v}} \cup A_{\bar{v}v}| \qquad \forall\, v \in V \tag{4.27}$$

As motivated in section 4.3.1, we compute the power consumption of line cards and chassis based on the best-case assumption that only the minimum number required to operate all circuits is active. We do however respect the bi-directional nature of line card interfaces, since the optimal allocation of ports from port pairs is trivial (cf. section 4.4.1). Thus, we determine the number of active port pairs $a_{\mathrm{PP}\,v}$ on node v from the set of circuits and derive the according number of line cards $a_{\mathrm{LC}\,v}$ and chassis $a_{\mathrm{LCC}\,v}$:

$$a_{\mathrm{PP}\,v} = \sum_{x=1}^{n_{\mathrm{PP}\,v}} \begin{cases} 1 & \text{if } \exists \left\langle x, i_{\mathrm{P}\,v_{l+1}}, [v, v_2, \ldots, v_{l+1}] \right\rangle \in A \vee \exists \left\langle i_{\mathrm{P}\,v_1}, x, [v_1, \ldots, v_l, v] \right\rangle \in A \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, v \in V \tag{4.28}$$

$$a_{\mathrm{LC}\,v} = \left\lceil \frac{a_{\mathrm{PP}\,v}}{N_{\mathrm{PP}}} \right\rceil \qquad \forall\, v \in V \tag{4.29}$$

$$a_{\mathrm{LCC}\,v} = \left\lceil \frac{a_{\mathrm{LC}\,v}}{N_{\mathrm{LC}}} \right\rceil \qquad \forall\, v \in V \tag{4.30}$$

Finally, we obtain the transit traffic volume $t_{\mathrm{T}\,v}$ electronically processed at node v from the routed demands. As postulated in section 4.1.2, we disregard the processing of add and drop traffic since it is independent of the network configuration:

$$t_{\mathrm{T}\,v} = \sum_{R_{sd} \in R} \sum_{\substack{\langle t,[v_1,v_2,\dots,v_l,v_{l+1}]\rangle \in R_{sd} \\ \text{where } v \in \{v_2,\dots,v_l\}}} t \qquad \forall\, v \in V \tag{4.31}$$

4.3.5.2 *Reconfiguration Effort*

As discussed in section 4.2, we quantify the network reconfiguration effort by the number of newly established and torn-down circuits. For the reconfiguration cost, we weight this number by the reconfiguration penalty δ. Extracting the number of modified circuits from the sets of previous and active ones, we obtain the cost term

$$c_{\mathrm{R}} = \delta \cdot |(P \cup A) \setminus (P \cap A)| \tag{4.32}$$

The choice to quantify the reconfiguration effort by the number of modified circuits is partly justified by the power consumption due to transient circuits. However, the reconfiguration penalty is not supposed to represent the energetic cost of such circuits, since this energetic cost is not fix but depends on the resource occupation at the terminating nodes. Instead, we vary δ, which allows us to trade off the reconfiguration effort and the energy consumption of the resulting configuration. The penalty is reasonably bounded above by the average energetic cost of a circuit since larger values impede the teardown of unused circuits, which contradicts the idea of energy-oriented network reconfiguration.

4.3.5.3 *Traffic Blocking*

Section 4.2 formulates two objectives with respect to traffic blocking. First, blocking shall be avoided by any means if possible. Second, if this is infeasible, the amount of blocked traffic shall be minimized. For the first objective, it is insufficient to consider the volume of blocked traffic weighted by a high penalty in the cost function: blocking a minimal amount of traffic could result in lower total cost than operating the additional circuits required to transport it. We therefore count the number of virtual links where blocking occurs and weight it by the link blocking penalty β_{L}. Alternatively, one could apply a fixed penalty as soon as blocking occurs on any virtual link. Our choice has the benefit of minimizing the number of links which incur blocking. This tends to limit the number of traffic flows affected by blocking at the expense of potentially higher volumes of blocked traffic.

For the case that blocking is unavoidable, we add the volume of blocked traffic weighted by the traffic blocking penalty β_{T} to the cost function. This favors the configuration resulting in the least amount of blocked traffic. We simplify the problem formulation by considering the positive difference between the traffic routed onto a virtual link and the capacity of the circuits realizing it as blocked. Thus, we may count the same excess traffic at subsequent bottlenecks. However, we consider this approximation admissible due to the expected scarcity of blocking events. We obtain the according blocking cost term:

$$c_{\mathrm{B}} = \beta_{\mathrm{L}} \sum_{(i,j)\in E_{\mathrm{v}}} b_{ij} + \beta_{\mathrm{T}} \sum_{(i,j)\in E_{\mathrm{v}}} \max\left\{0; t_{ij} - \left|A_{ij}\right| B\right\} \tag{4.33}$$

In order to assure the priority of the blocking avoidance objective, the penalties β_{L} and β_{T} should be greater than the worst-case cost of switching traffic electrically in all nodes and operating additional circuits. I. e., the reference cost for circuits shall include the power consumption of two line cards and two chassis in addition to that of the ports terminating the circuit.

4.3.5.4 Cost Function

As discussed above, we realize the multi-objective optimization by adding up the different cost terms in one cost function. The resulting overall objective is:

$$\textbf{minimize} \quad c_{\mathrm{P}} + c_{\mathrm{R}} + c_{\mathrm{B}} \tag{4.34}$$

4.4 Solution Methods

This section describes two methods to solve the reconfiguration problem specified in section 4.3. According to Equation (4.26), (4.32), (4.33), and (4.34), the routing of circuits has no influence on the cost function of this problem. It is however relevant to the feasibility of a solution. This motivates us to reduce the complexity of the problem by separating the circuit realization, i. e. the routing in the physical topology along with the allocation of line card ports, from the remaining problem of optimizing the configuration of the upper layer. More precisely, we first determine the upper-layer configuration and then verify the feasibility of its circuits. Such a sequential problem decomposition into virtual topology definition followed by heuristic circuit routing is also found in [64], which likewise assumes full wavelength conversion capability.

Section 4.4.1 details our heuristic circuit realization method. For the optimization of the upper layer configuration, we propose two solution methods. The first one, presented in section 4.4.2 jointly optimizes circuit configuration and demand routing. Being based on mathematical programming, it provides exact solutions or lower bounds for the objective function. Since this method is unlikely to solve larger problem instances within the tight time bounds implied by the network reconfiguration procedure, we designed the alternative solution method described in section 4.4.3. It applies heuristic optimization by simulated annealing to the virtual topology. Unlike the first method, it does not optimize demand routing. Instead, we route demands along their shortest path during the heuristic topology optimization procedure. Afterwards, we apply a simple heuristic to mitigate specific issues of shortest-path demand routing.

The author of this monograph described these methods or preliminary versions thereof in several publications. A first version of the heuristic optimization method disregarding resource constraints and circuit routing is presented in [112]. Preserving these exclusions, we introduced the demand rerouting heuristic and propose a preliminary version of the linear programming formulation in [113]. We finally lifted the restrictions in [111], describing and evaluating the final version of the heuristic virtual topology optimization method under realistic assumptions similar to chapter 5.

4.4.1 Heuristic for Circuit Realization

Due to the resource pre-occupation constraints and the optimization objective of limiting circuit modifications, a significant share of the previously active circuits is generally maintained in a subsequent configuration. The task of circuit realization accordingly consists in routing and resource allocation for additionally required circuits under the constraints implied by the persisting ones. A related task is the selection of circuits for teardown if one of several circuits connecting a pair of nodes is no longer needed. Since the optimization-heuristic method of section 4.4.3 iteratively tests different configuration candidates, it may also be necessary to undo teardown decisions and select the best circuit to maintain. These three tasks are addressed in the following subsections.

As argued in the following, a simple heuristic achieves optimal circuit realization in many relevant cases. For this heuristic, it is sufficient to consider one circuit to be established or torn down, respectively, at a time. We accordingly specify methods to allocate resources for one circuit and to select one circuit for teardown, respectively. For the algorithmic descriptions, we assume that the set of active circuits A is initialized with all circuits of the previous configuration, i. e. $A \leftarrow P$, when computation of a new configuration starts. The algorithms then incrementally modify the set A.

4.4.1.1 Circuit Setup

The circuit setup heuristic aims at allocating the resources required to establish a unidirectional circuit between two given nodes. The respective resources are one output port at the source node, one input port at the target node, and optical channels along a path between these nodes in the physical topology.

Since we disregard the structural assignment of ports to hierarchical components (i. e. line cards), the requirement that the two ports of a port pair may only connect to one other port pair represents the only constraint for port allocation. The allocation is resource-optimal in the sense that it blocks the minimal amount of resource if it uses inverse ports of port pairs sustaining a circuit in the opposite direction. We first look for an according unidirectional inverse circuit among the persistent circuits, then among the newly established ones, and finally among those intended for teardown. While in the last case the selected port pairs are likely to operate only the new circuit in the next configuration, this strategy minimizes the resources required during transition. If no single inverse circuit is found, we allocate one unused port pair each in the source and destination node. Circuit establishment fails if no such port pairs are available.

After allocating the ports we route the circuit. A circuit occupies the least amount of resources in terms of optical channels if it follows the shortest path according to hop count in the physical topology. Nevertheless, a detour could be beneficial if it spares resources on some link which are required to establish circuits that are otherwise not feasible. However, our investigations showed that capacity constraints are hardly an issue on physical links in the considered scenarios thanks to the relatively high channel capacity of a fiber and the assumed wavelength conversion capability. We thus try to route the circuit on the shortest path in terms of hop count, breaking ties in favor of the geographically shorter path, and try to allocate an optical channel on each link. If resources are depleted on some links, we exclude these links and try to realize the circuit on the next shortest path by repeating the procedure. Our strategy thus falls into the *adaptive* category according to [98]. We abort with a failure to set up the circuit if the exclusion of links leaves source and target node disconnected or the geographical length of the resulting path exceeds the maximum transparent optical reach L. As an exception, we do allow circuits on single physical links longer than L. Algorithm 4.1 details the circuit realization heuristic.

4.4.1.2 Circuit Teardown

The circuit teardown heuristic selects the most appropriate circuit connecting a given source node with a given target node for deactivation if less capacity is required on the respective virtual link. Obeying arguments complementary to those of the previous subsection, it sequentially evaluates three criteria.

Due to our reconfiguration scheme with pre-occupation constraints, resources of discontinued circuits are not immediately available for use in different settings. Hence, it is more advantageous to abandon new circuits intended for establishment since their resources directly become available again. We therefore primarily select circuits from this category and only resort to previously existing circuits if no such circuits exist.

If the category selected according to this first criterion contains several candidates, we next consider the port pairs terminating these circuits, since port pairs are generally scarcer than optical channels. We preferentially tear down circuits whose port pairs do not sustain a circuit in the opposite direction. In this way, the port pairs will be freed and become available for arbitrary use in this or the subsequent configuration. If several candidates remain, we select an arbitrary one of the longest circuits in terms of hop count in the physical topology, thereby freeing the maximum number of optical channels. Algorithm 4.2 formalizes this procedure.

4.4.1.3 Circuit Reactivation

The reactivation heuristic selects the most appropriate circuit connecting a given source node with a given target node for removal from the set of circuits intended for teardown. It reversely follows the rationale of the teardown heuristic: preferentially, circuits whose port pairs maintain a circuit in the opposite direction are spared deactivation. If this leaves several options, one of the shortest circuits in terms of hop count in the physical topology is maintained. This procedure is described in Algorithm 4.3.

Algorithm 4.1 Resource allocation for circuit from node i to node j

{allocation of ports}
found ← **false**
for InverseCircs **in** $\left[A_{ji} \cap P_{ji}\ ;\ A_{ji} \setminus P_{ji}\ ;\ P_{ji} \setminus A_{ji}\right]$ **do** {sets of decreasing priority}
  for all $\langle y, x, [j, v_2, \ldots, v_l, i]\rangle \in$ InverseCircs **do**
    if $\nexists \langle x, y, [i, \bar{v}_2, \ldots, \bar{v}_l, j]\rangle \in A_{ij} \cup P_{ij}$ **then** {no inverse circuit on port pair}
      $(i_{\mathrm{P}\,i}, i_{\mathrm{P}\,j}) \leftarrow (x, y)$
      found ← **true**
      break for InverseCircs
    end if
  end for
end for
if not found **then** {allocate unused port pairs}
  $I_{\mathrm{P}\,i} \leftarrow \left\{x \in \{1, \ldots, n_{\mathrm{PP}\,i}\} \mid \nexists \left\langle x, i_{\mathrm{P}\,v_{l+1}}, [i, v_2, \ldots, v_{l+1}]\right\rangle \in A \cup P\right.$
    $\left. \wedge\ \nexists \langle i_{\mathrm{P}\,v_1}, x, [v_1, \ldots, v_l, i]\rangle \in A \cup P\right\}$
  $I_{\mathrm{P}\,j} \leftarrow \left\{y \in \{1, \ldots, n_{\mathrm{PP}\,j}\} \mid \nexists \left\langle y, i_{\mathrm{P}\,v_{l+1}}, [j, v_2, \ldots, v_{l+1}]\right\rangle \in A \cup P\right.$
    $\left. \wedge\ \nexists \langle i_{\mathrm{P}\,v_1}, y, [v_1, \ldots, v_l, j]\rangle \in A \cup P\right\}$
  if $I_{\mathrm{P}\,i} = \emptyset$ **or** $I_{\mathrm{P}\,j} = \emptyset$ **then**
    return $\emptyset$ {circuit infeasible since ports depleted}
  end if
  $(i_{\mathrm{P}\,i}, i_{\mathrm{P}\,j}) \leftarrow \left(\min_{x \in I_{\mathrm{P}\,i}} x, \min_{y \in I_{\mathrm{P}\,j}} y\right)$
end if
{routing and allocation of optical channels}
$E_{\mathrm{E}} \leftarrow \emptyset$ {set of edges excluded due to resource depletion}
repeat
  $\boldsymbol{R}_{\mathrm{C}} = [v_1, \ldots, v_{l+1}] \leftarrow$ nodes forming shortest path from i to j in graph $\langle V, E_{\mathrm{p}} \setminus E_{\mathrm{E}}\rangle$
  if $\boldsymbol{R}_{\mathrm{C}} = \emptyset$ **then** {no (more) path found}
    return $\emptyset$ {circuit infeasible since no path (with sufficient resources) exists}
  end if
  feasible ← **true**
  $l_{\mathrm{G}} \leftarrow 0$ {geographical path length}
  for $x = 1$ **to** l **do**
    $l_{\mathrm{G}} \leftarrow l_{\mathrm{G}} + l_{v_x v_{x+1}}$
    if $n_{\mathrm{F}\,v_x v_{x+1}} N_\lambda - 1$
      $\leq \left|\left\{\left\langle \bullet, \bullet, \left[\bar{v}_1, \ldots, \bar{v}_{\bar{l}+1}\right]\right\rangle \in A \cup P \mid \exists y \in \{1, \ldots, \bar{l}\} : \left(\bar{v}_y, \bar{v}_{y+1}\right) = (v_x, v_{x+1})\right\}\right|$ **then**
      feasible ← **false** {resources depleted on physical link (v_x, v_{x+1})}
      $E_{\mathrm{E}} \leftarrow E_{\mathrm{E}} \cup \{(v_x, v_{x+1})\}$
    end if
  end for
  if $l_{\mathrm{G}} > L$ **and** $|\boldsymbol{R}_{\mathrm{C}}| > 2$ **then**
    return $\emptyset$ {circuit infeasible since multi-hop path exceeds maximum optical reach}
  end if
until feasible
return $C \leftarrow \left\langle i_{\mathrm{P}\,i}, i_{\mathrm{P}\,j}, \boldsymbol{R}_{\mathrm{C}}\right\rangle$

Algorithm 4.2 Selection of circuit from node i to node j for teardown

if $A_{ij} \setminus P_{ij} \neq \emptyset$ **then** {circuits for setup exist}
  $T_{ij} \leftarrow A_{ij} \setminus P_{ij}$ {set of candidates for teardown}
else
  $T_{ij} \leftarrow A_{ij}$
end if
$T_{\text{unidir}\,ij} \leftarrow \left\{ \langle x, y, [i, v_2, \ldots, v_l, j] \rangle \in T_{ij} \middle| \nexists \langle y, x, [j, \bar{v}_2, \ldots, \bar{v}_{\bar{l}}, i] \rangle \in A_{ji} \right\}$
if $T_{\text{unidir}\,ij} \neq \emptyset$ **then**
  $T_{ij} \leftarrow T_{\text{unidir}\,ij}$
end if
return $C \leftarrow \arg\max_{\langle x,y,[i,v_2,\ldots,v_l,j] \rangle \in T_{ij}} l$

Algorithm 4.3 Selection of circuit from node i to node j for reactivation

$T_{ij} \leftarrow P_{ij} \setminus A_{ij}$ {set of candidates for reactivation}
$T_{\text{bidir}\,ij} \leftarrow \left\{ \langle x, y, [i, v_2, \ldots, v_l, j] \rangle \in T_{ij} \middle| \exists \langle y, x, [j, \bar{v}_2, \ldots, \bar{v}_{\bar{l}}, i] \rangle \in A_{ji} \right\}$
if $T_{\text{bidir}\,ij} \neq \emptyset$ **then**
  $T_{ij} \leftarrow T_{\text{unidir}\,ij}$
end if
return $C \leftarrow \arg\min_{\langle x,y,[i,v_2,\ldots,v_l,j] \rangle \in T_{ij}} l$

4.4.2 Mathematical Optimization of the Upper Layer

Our solution method jointly optimizing the circuit configuration and the routing of demands in the upper layer formulates and solves this problem as a MILP. While the formulation includes constraints on port pairs, we do not verify whether the fiber capacity allows routing the circuits. This relaxation of lower-layer constraints, which is also applied in [11, 51], serves for reducing the complexity of the problem. In particular, the linear program formulation does not allow emulating the adaptive shortest-path routing heuristic described in section 4.4.1.1. Hence, we would need to optimize circuit routing along with the upper-layer configuration, which makes the problem prohibitively complex to solve. The slow convergence of the two-layer optimization applied for the DUDL (dynamic upper, dynamic lower) scheme in [21] for moderately-sized networks illustrates this issue.

This limitation leaves two application scenarios for the MILP-based solution method. First, the exact method can provide a benchmark – at least in terms of a lower, dual bound for the objective value – for the evaluation of other methods under the best-case assumption that the installed fiber capacity is sufficient. Second, we can combine solving the MILP with a subsequent step trying to realize the resulting circuits by means of the heuristics of section 4.4.1. In the practically rare cases where some circuits cannot be routed, one can either add constraints to the MILP to restrict the number of circuits between the concerned nodes and solve the problem anew, or one can apply a heuristic similar to section 4.4.3.4 to reroute traffic from the infeasible circuits. For the evaluation in chapter 5, we verify whether the circuit configuration is implementable but do not provide any remedy if not.

We express the optimization of the upper-layer configuration as a multi-commodity flow problem. Since we do not intend to restrict the routing of demands, we opt for the node-link formulation [18], which is advantageous in this case. To limit complexity, we use its aggregated form introduced in section 2.3.1.2. In the next subsection, we define a number of parameters and variables not yet introduced in the problem definition of section 4.3 but required for the MILP formulation. On this basis, section 4.4.2.2 presents the actual problem formulation. Section 4.4.2.3 comments on its complexity.

4.4.2.1 Additional Definitions

The omission of circuit routing and fiber capacity limits in the problem formulation requires some additional concepts to cover the remaining physical constraints. In particular, the solution of the MILP shall not postulate circuits that are infeasible due to excessive length. We therefore consider the set of feasible virtual links, which contains all node pairs interconnectable by a path of a geographical length inferior to the maximum transparent reach or consisting of only one hop:

$$E_\mathrm{f} = \Big\{ (i,j) \in V \times V \setminus \{(v,v) | v \in V\} \Big| \exists \text{ path } \boldsymbol{R}_\mathrm{C} = [v_1 = i, v_2, \ldots, v_l, v_{l+1} = j] \text{ in graph } G_\mathrm{p} \text{ such that } l = 1 \vee \sum_{x=1}^{l} l_{v_x v_{x+1}} \leq L \Big\} \tag{4.35}$$

On each of these links, we quantify the number of active circuits in the previous and new configuration, respectively, by p_{ij} and a_{ij}. By r_{ij}, we designate the number of modified circuits. Conceptually, we can relate these quantities to the circuit sets of the generic problem definition:

$$p_{ij} = |P_{ij}| \qquad \forall\, (i,j) \in E_\mathrm{f} \tag{4.36}$$

$$a_{ij} = |A_{ij}| \qquad \forall\, (i,j) \in E_\mathrm{f} \tag{4.37}$$

$$r_{ij} = |(A_{ij} \cup P_{ij}) \setminus A_{ij} \cap P_{ij})| \qquad \forall\, (i,j) \in E_\mathrm{f} \tag{4.38}$$

In order to express resource constraints in terms of port pairs and to account for the power consumption of hierarchical components depending on active port pairs, we make use of a concept partially developed in the Diploma thesis project documented in [114]. It centers on determining the number of port pairs occupied by circuits interconnecting two nodes in either direction. Formally, we consider the port pairs dedicated to circuits on the *undirected* virtual link between these nodes. For the MILP formulation, we define the set of feasible undirected virtual links as a set of node pairs, where the order of the nodes in each pair is arbitrary:

$$E_\mathrm{fu} \subset E_\mathrm{f} : \quad ((i,j) \in E_\mathrm{fu}) \oplus ((j,i) \in E_\mathrm{fu}) \quad \forall\, (i,j) \in E_\mathrm{f} \tag{4.39}$$

For the number of port pairs occupied at each end of an undirected link $(i,j) \in E_\mathrm{fu}$ during the transition and in the new configuration, respectively, we introduce the variables $m_{\mathrm{PP}\,ij}$ and $a_{\mathrm{PP}\,ij}$.

In the aggregated commodity flow formulation, we subsume all traffic originating from one node by one commodity. As introduced in section 2.3.1.2, we accordingly define the set of commodities as the set of nodes being source to at least one demand:

$$K = \{v \in V | \exists\, d \in V : d_{vd} > 0\} \tag{4.40}$$

For each commodity, we establish the signed net demand value entering the network at each node:

$$d_v^{(k)} = \begin{cases} \sum_{d \in V} d_{vd} & \text{if } v = k \\ -d_{kv} & \text{if } v \neq k \end{cases} \qquad \forall\, k \in K, v \in V \tag{4.41}$$

We finally introduce flow variables $f_{ij}^{(k)}$ for each commodity k on each feasible link $(i, j) \in E_\text{f}$. In addition, we define variables $t_{\text{B}\,ij}$ for the traffic volume blocked on these links.

4.4.2.2 Mixed Integer Linear Program Formulation

This section describes the mathematical problem formulation as commonly done by defining input parameters, variables, objective, and constraints. It finally addresses a restriction of the problem used to reduce computation time for the evaluation in chapter 5.

4.4.2.2.1 Parameters and Variables

Table 4.1 lists all required input parameters of the MILP along with their value sets. All of these parameters have been introduced throughout the present chapter. Likewise, Table 4.2 presents

Table 4.1: Parameters of the MILP formulation

Physical, system, and algorithm parameters		
B	$\in \mathbb{R}^+$	Capacity of a circuit
N_LC	$\in \mathbb{Z}^+$	Maximum number of line cards per line card chassis
N_PP	$\in \mathbb{Z}^+$	Maximum number of port pairs per line card
P_LC	$\in \mathbb{R}_0^+$	Power consumption per line card
P_LCC	$\in \mathbb{R}_0^+$	Power consumption per line card chassis
P_P	$\in \mathbb{R}_0^+$	Power consumption per line card port
P_T	$\in \mathbb{R}_0^+$	Power consumption per switched traffic unit
δ	$\in \mathbb{R}_0^+$	Reconfiguration penalty
β_L	$\in \mathbb{R}_0^+$	Penalty for blocking on some virtual link
β_T	$\in \mathbb{R}_0^+$	Penalty per blocked traffic volume
Scenario-dependent parameters		
V		Set of all nodes
K	$\subseteq V$	Set of commodities, i. e. nodes sourcing traffic demands
E_f	$\subset V \times V$	Set of feasible virtual links
E_fu	$\subset V \times V$	Set of undirected feasible virtual links
$n_{\text{PP}\,v}$	$\in \mathbb{Z}_0^+$	Number of node pairs installed at node $v \in V$
p_{ij}	$\in \mathbb{Z}_0^+$	Number of previously active circuits on link $(i, j) \in E_\text{f}$
$d_v^{(k)}$	$\in \mathbb{R}$	Net demand value of commodity $k \in K$ at node $v \in V$

Table 4.2: Variables of the MILP formulation

Per node $v \in V$		
$a_{\mathrm{LC}\,v}$	$\in \mathbb{Z}_0^+$	Number of active line cards
$a_{\mathrm{LCC}\,v}$	$\in \mathbb{Z}_0^+$	Number of active line card chassis
Per (directed) feasible virtual link $(i,j) \in E_\mathrm{f}$		
a_{ij}	$\in \mathbb{Z}_0^+$	Number of active circuits in the new configuration
r_{ij}	$\in \mathbb{Z}_0^+$	Number of established or torn-down circuits
b_{ij}	$\in \{0,1\}$	Binary variable indicating blocking
$t_{\mathrm{B}\,ij}$	$\in \mathbb{R}_0^+$	Traffic volume blocked
$f_{ij}^{(k)}$	$\in \mathbb{R}_0^+$	Flow of commodity $k \in K$
Per undirected feasible virtual link $(i,j) \in E_\mathrm{fu}$		
$a_{\mathrm{PP}\,ij}$	$\in \mathbb{Z}_0^+$	Number of port pairs required at each end in new configuration
$m_{\mathrm{PP}\,ij}$	$\in \mathbb{Z}_0^+$	Number of port pairs required at each end during reconfiguration

the minimum set of variables required for the MILP formulation. This set differs from the set of attributes specified in the problem description of section 4.3 for different reasons.

Most notably, we find none of the elements of the solution specification according to section 4.3.3. As previously discussed, the set of circuits is reduced to the number of circuits per virtual link since the MILP disregards circuit realization. Routed demands are not directly obtained either due to the aggregated form of the commodity flow formulation. While we can map routed demands to commodity flow variables, this mapping is not uniquely invertible:

$$f_{ij}^{(k)} = \sum_{d \in V \setminus \{k\}} \sum_{\substack{\langle t, [v_1, \ldots, v_{l+1}] \rangle \in R_{kd} \text{ where} \\ \exists x \in \{1, \ldots, l\} : (v_x, v_{x+1}) = (i,j)}} t \qquad \forall\, k \in K; (i,j) \in E_\mathrm{f} \tag{4.42}$$

The authors of [23] discuss a simple heuristic to obtain one valid set of routed demands from a certain assignment of flow variables. However, we abstain from performing such a decomposition since the evaluation of a solution is feasible based on the aggregated flow values alone.

Further variables used in the problem definition and missing in Table 4.2 are directly derivable from the MILP variables. For instance, the number of active ports on a node $a_{\mathrm{P}\,v}$ is obtained from the number of circuits a_{ij} it terminates. Likewise, the amount of transit traffic is computable from the flow variables.

4.4.2.2.2 Objective

The objective in Equation (4.43) exactly corresponds to the cost function specified in Equation (4.26), (4.32), (4.33), and (4.34). It formally differs from it by expressing the number of active ports by that of active circuits and the transit traffic by the flow variables. In addition, the blocked traffic volumes per link are directly given by variables $t_{\mathrm{B}\,ij}$.

$$\begin{aligned}\textbf{minimize}\quad & 2P_{\mathrm{P}} \sum_{(i,j)\in E_{\mathrm{f}}} a_{ij} + P_{\mathrm{LC}} \sum_{v\in V} a_{\mathrm{LC}\,v} + P_{\mathrm{LCC}} \sum_{v\in V} a_{\mathrm{LCC}\,v} + P_{\mathrm{T}} \sum_{k\in K} \sum_{(i,j)\in E_{\mathrm{f}}:i\neq k} f_{ij}^{(k)} \\ & + \delta \sum_{(i,j)\in E_{\mathrm{f}}} r_{ij} + \beta_{\mathrm{L}} \sum_{(i,j)\in E_{\mathrm{f}}} b_{ij} + \beta_{\mathrm{T}} \sum_{(i,j)\in E_{\mathrm{f}}} t_{\mathrm{B}\,ij} \end{aligned} \tag{4.43}$$

4.4.2.2.3 Constraints

Equation (4.44) to (4.56) give the constraints. Equation (4.44) postulates commodity flow conservation and incorporates the requirement to route all demands according to Equation (4.17). Equation (4.45) enforces sufficient circuit capacity to accommodate the traffic on each virtual link unless some of the traffic is blocked. This corresponds to Equation (4.19) except that the blocked traffic volume is directly counted. Equation (4.46) then derives the link blocking indication from the blocked traffic volume, where Γ is again a constant greater than any possibly blocked traffic volume.

The next set of equations determines the numbers of port pairs, which is used to compute the number of active higher-level resources and to enforce resource constraints. Assuming the most economical allocation strategy, the number of port pairs required for the circuits interconnecting a given pair of nodes is the maximum of the numbers of circuits in each direction. In a linear formulation, we have to replace the maximum operator by the postulation that the number of port pairs is greater or equal to either of the numbers of circuits, and if relevant the objective should aim at minimizing the number of port pairs. Equation (4.47) and (4.48) are the respective inequalities for the port pairs active in the new configuration. The minimization is indirectly achieved via Equation (4.53) and the cost term for active line cards in the objective.

Regarding resource constraints, the number of port pairs required during the transition to the new configuration is of relevance. Under the same optimal allocation assumption, we require the maximum of the numbers of active circuits in each direction in the previous and the new configuration for each link. This is expressed in Equation (4.49) to (4.51), making use of the previously determined number of port pairs in the new configuration to save one equation. Equation (4.52) finally limits the number of transiently used port pairs per node to the number of installed ones. To evaluate this feasibility constraint, it is not necessary to explicitly minimize the contribution $m_{\mathrm{PP}\,ij}$ of each adjacent link. These equations reflect the postulations of Equation (4.20) and (4.21) in conjunction with Equation (4.22) and (4.23) without explicit assignment of ports to circuits.

Equation (4.53) and (4.54) compute the number of active line cards and line card chassis, respectively, from that of active port pairs and line cards. They correspond to Equation (4.29) and (4.30), transforming the ceiling operator into an inequality that additionally requires minimizing $a_{\mathrm{LC}\,v}$ and $a_{\mathrm{LCC}\,v}$, respectively. This minimization is provided by the objective.

Finally, we compute the number of circuit modifications for each virtual link. Conceptually, the set operation of Equation (4.32) in total or Equation (4.38) per link, respectively, translates into the absolute value of the difference of the numbers of active circuits in the previous and the new configuration per link. For linearization, taking the absolute value is replaceable by computing the maximum of the argument and the negative argument, yielding Equation (4.55) and (4.56) with the previously discussed transformation. The minimization of r_{ij} is again provided by the objective.

$$\sum_{j \in V:(v,j) \in E_\mathrm{f}} f_{vj}^{(k)} - \sum_{i \in V:(i,v) \in E_\mathrm{f}} f_{iv}^{(k)} = d_v^{(k)} \qquad \forall\, k \in K, v \in V \tag{4.44}$$

$$a_{ij}B - \sum_{k \in K} f_{ij}^{(k)} + t_{\mathrm{B}\,ij} \geq 0 \qquad \forall\, (i,j) \in E_\mathrm{f} \tag{4.45}$$

$$b_{ij} \cdot \Gamma - t_{\mathrm{B}\,ij} \geq 0 \qquad \forall\, (i,j) \in E_\mathrm{f} \tag{4.46}$$

$$a_{\mathrm{PP}\,ij} - a_{ij} \geq 0 \qquad \forall\, (i,j) \in E_\mathrm{fu} \tag{4.47}$$

$$a_{\mathrm{PP}\,ij} - a_{ji} \geq 0 \qquad \forall\, (i,j) \in E_\mathrm{fu} \tag{4.48}$$

$$m_{\mathrm{PP}\,ij} - a_{\mathrm{PP}\,ij} \geq 0 \qquad \forall\, (i,j) \in E_\mathrm{fu} \tag{4.49}$$

$$m_{\mathrm{PP}\,ij} - p_{ij} \geq 0 \qquad \forall\, (i,j) \in E_\mathrm{fu} \tag{4.50}$$

$$m_{\mathrm{PP}\,ij} - p_{ji} \geq 0 \qquad \forall\, (i,j) \in E_\mathrm{fu} \tag{4.51}$$

$$n_{\mathrm{PP}\,v} - \sum_{j \in V:(v,j) \in E_\mathrm{fu}} m_{\mathrm{PP}\,vj} - \sum_{i \in V:(i,v) \in E_\mathrm{fu}} m_{\mathrm{PP}\,iv} \geq 0 \qquad \forall\, v \in V \tag{4.52}$$

$$a_{\mathrm{LC}\,v} \cdot N_\mathrm{PP} - \sum_{j \in V:(v,j) \in E_\mathrm{fu}} a_{\mathrm{PP}\,vj} - \sum_{i \in V:(i,v) \in E_\mathrm{fu}} a_{\mathrm{PP}\,iv} \geq 0 \qquad \forall\, v \in V \tag{4.53}$$

$$a_{\mathrm{LCC}\,v} \cdot N_\mathrm{LC} - a_{\mathrm{LC}\,v} \geq 0 \qquad \forall\, v \in V \tag{4.54}$$

$$r_{ij} - \left(a_{ij} - p_{ij}\right) \geq 0 \qquad \forall\, (i,j) \in E_\mathrm{f} \tag{4.55}$$

$$r_{ij} - \left(p_{ij} - a_{ij}\right) \geq 0 \qquad \forall\, (i,j) \in E_\mathrm{f} \tag{4.56}$$

4.4.2.2.4 Problem Restriction

Investigations of the convergence behavior of MILP solutions computed by a solver software in [114] showed that introducing the blocking terms significantly increases the time required to solve the problem or to reduce the gap between primal and dual solution, respectively. We therefore restrict the problem as follows for most evaluation runs and only revert to the original problem formulation if the restricted one proves unsolvable.

In the following, we describe the aspects in which the restricted version of the problem not allowing any traffic blocking differs from the original problem formulation detailed above. Regarding variables, we remove those indicating blocking, i. e. b_{ij} and $t_{\mathrm{B}\,ij}$. This turns the constraint in Equation (4.46) unnecessary, and we have to strictly enforce enough capacity on each link, replacing Equation (4.45) by

$$a_{ij}B - \sum_{k \in K} f_{ij}^{(k)} \geq 0 \qquad \forall\, (i,j) \in E_\mathrm{f} \tag{4.57}$$

Finally, we reformulate the objective to exclude the blocking terms, yielding (instead of Equation (4.43)):

$$\textbf{minimize} \quad 2P_\text{P} \sum_{(i,j)\in E_\text{f}} a_{ij} + P_\text{LC} \sum_{v\in V} a_{\text{LC}\,v} + P_\text{LCC} \sum_{v\in V} a_{\text{LCC}\,v} + P_\text{T} \sum_{k\in K} \sum_{(i,j)\in E_\text{f}:i\neq k} f_{ij}^{(k)} + \delta \sum_{(i,j)\in E_\text{f}} r_{ij} \tag{4.58}$$

4.4.2.3 Complexity and Solvability

MILP problems are generally hard to solve; many of them belong to the class of NP-complete problems [19, 20]. While this property is likely to hold for our problem defined in the previous section 4.4.2.2, we refrain from providing a formal proof since its applicability to instances of realistic size is of more relevance to the objectives of this monograph than the formal classification of the problem.

To quantify the problem size, we can refer to the number of variables according to Table 4.2 and the number of constraint equations detailed in section 4.4.2.2.3. These quantities depend on the following scenario parameters: the number of network nodes $|V|$, the number of feasible virtual links $|E_\text{f}|$, and the number of commodities $|K|$, i. e. of nodes sourcing traffic demands. For an estimation of the problem size, we can approximate the other parameters by the number of nodes assuming that every node sources demands and that the virtual topology is potentially fully meshed. We then obtain $|K| \approx |V|$ and $|E_\text{f}| \approx |V|(|V|-1) \approx |V|^2$ for larger network topologies. In accordance with remarks on the aggregated multi-commodity flow formulation in section 2.3, the number of variables is dominated by the flow variables $f_{ij}^{(k)}$ scaling with $|V|^3$. It is worth noting that the number of discrete variables (both integer and binary), which arguably have dominant impact on the performance of exact solution methods, only quadratically depends on the number of nodes. The number of constraint equations likewise scales with $|V|^2$.

As indicated in section 2.4.1, the problem structure and the ability to determine tight lower bounds is more important for the convergence of the branch and bound algorithm than the absolute number of variables and constraints. This is underpinned by the findings of the Diploma thesis project documented in [114], which investigated the execution time performance of an open-source solver software when confronted with the MILP defined in section 4.4.2.2 and preliminary or restricted versions thereof. For instance, using the reconfiguration penalty in the objective function diversifies the cost bounds assigned to different nodes of the solution tree and thus results in a faster completion of the branch and bound algorithm, although it implies two additional constraint inequations (Equation (4.55) and (4.56)) and one integer variable (r_{ij}) per feasible link. This effect is also visible in the results in section 5.5.1. Similarly, introducing bounds on the installed port pairs and pre-occupation constraints did not significantly reduce the solution performance despite an even larger number of additional constraint equations. In contrast, allowing the blocking of traffic, which only requires one binary and one real variable along with one additional constraint per feasible link, increased the time required to reach a given gap between the best identified primal solution and the dual bound to a multiple. Presumably, this is due to the optimizer primarily exploring the additionally allowed but undesirable part of the solution space which blocks a significant amount of traffic.

Due to these observations, we take an empirical approach and determine the maximum problem size for which the MILP is solvable by evaluating the convergence behavior of a solver software for realistic problem instances of different sizes. Section 5.5.1 reports the most relevant findings for the scenarios defined in section 5.1. In his Diploma thesis, Schöck [114] provides more details on a similar investigation.

4.4.3 Heuristic Optimization of the Virtual Topology with a Simple Routing Heuristic

Since the exact solution method detailed in section 4.4.2 is unlikely to solve larger problem instances in reasonable time, we propose an alternative solution method based on heuristic optimization. Selecting among the meta-heuristic methods outlined in section 2.4.3, we opted for simulated annealing (SA) for several reasons. In view of on-line application, we prefer a more light-weight sequential search method over evolutionary algorithms. This argument is supported by the Master thesis project documented in [115], which explored the application of genetic algorithms to a preliminary version of our problem. Testing a number of solution representations and recombination strategies, it did not obtain any better solutions from this method than from simulated annealing while incurring the inherent overhead of creating and manipulating a population comprising many solutions.

Among the sequential search methods, we exclude iterative improvement since unable to escape local optima and rarely-applied GRASP. Finally choosing between tabu search and SA, we selected the latter due to its light-weight implementation, which does not require the exhaustive evaluation of neighborhoods. In addition, Sommer [43] found SA outperforming tabu search for a different network optimization problem.

Meta-heuristic methods conceived for combinatorial optimization problems cannot determine suitable values for continuous variables of mixed-integer problems. Therefore, our simulated-annealing based method only optimizes the virtual topology while applying a simple strategy for demand routing. Sengezer and Karasan [52] take the same approach for a tabu-search based heuristic. Unlike them, we do however not optimize the number of active circuits on feasible virtual links but only determine the virtual links which may be supported by circuits. After routing the demands, we derive the number of required circuits on these links from the traffic load. This procedure reduces the complexity of the optimization problem, facilitating on-line application. In consequence, we cannot apply a residual-capacity based demand routing strategy as used in [52], but simply route demands on their shortest paths in the virtual topology. Gençata and Mukherjee [54] apply this routing scheme with a simple virtual topology adaptation heuristic.

Conceptually, our choice reduces the reconfiguration problem to a binary minimization problem, where the existence of a virtual link is indicated by an associated binary variable. In terms of our previous definitions, the problem targets defining the set of active virtual links E_{v}. We accordingly define the perturbation for our SA based method to add or remove one virtual link in this set. This is in line with the observation in section 2.5.3 that the perturbation should only perform a small modification. After adapting the shortest-path routing of demands to the new topology and deriving the numbers of required circuits, we immediately try to allocate all resources for additional circuits and account for the cost of traffic blocking if this is not feasible.

The deterministic shortest-path routing of demands may result in disadvantageous configurations with unnecessary blocking and lowly utilized circuits. After the optimization of the virtual topology, we therefore run a post-processing heuristic trying to resolve these issues by rerouting parts of the traffic from concerned virtual links in a first-fit manner. For a preliminary version of our reconfiguration problem disregarding resource constraints, we investigated the benefit of applying such a heuristic compared to pure topology optimization in [113] and obtained moderate improvements of energy savings of circa 2 %. Sabella *et al.* [46] use a similar procedure to reroute traffic demands in order to improve the network configuration returned by a heuristic. However, they target the objective of reducing congestion instead of turning circuits superfluous.

Comparing a simple deactivation heuristic working on complete virtual links with heuristic schemes adapting individual circuits, Bonetto *et al.* [84] observe a significant penalty in the solution quality due to the coarse granularity of adaptation. By optimizing the set of virtual links and improving demand routing by post-processing, our method has the potential to overcome this issue. In support of this statement, results in section 5.5 only show a moderate penalty in the solution quality between our method and the MILP-based scheme.

Due to its iterative nature and the repeated evaluation of candidate solutions, the meta-heuristic method requires some additional concepts and definitions compared to the generic problem description in section 4.3. We address these aspects in section 4.4.3.1. Section 4.4.3.2 details the problem-specific routines used by the SA algorithm. We then discuss this algorithm and our cooling schedule in section 4.4.3.3. Section 4.4.3.4 finally describes the post-processing heuristic.

4.4.3.1 Additional Concepts and Definitions

The perturbation of the virtual topology is likely to produce candidate solutions with partitioned topologies not providing any path for some demands. Such topology-related blocking is not covered by the blocking cost terms in section 4.3.5.3. Analogously to the variables for traffic blocking due to resource shortage, we introduce the variable b_{VT} counting the number of node-to-node demands for which we cannot find any path and the variable $t_{\mathrm{B\,VT}}$ for the total volume of these demands. In the cost function, we weight the latter by the cost coefficient β_{T} of the traffic volume not transported due to insufficient resources, and we apply a penalty of β_{D} for each blocked demand. This yields the following blocking cost term replacing Equation (4.33):

$$c_{\mathrm{B}} = \beta_{\mathrm{L}} \sum_{(i,j)\in E_{\mathrm{v}}} b_{ij} + \beta_{\mathrm{T}} \sum_{(i,j)\in E_{\mathrm{v}}} \max\left\{0; t_{ij} - \left|A_{ij}\right| B\right\} + \beta_{\mathrm{D}} \cdot b_{\mathrm{VT}} + \beta_{\mathrm{T}} \cdot t_{\mathrm{B\,VT}} \tag{4.59}$$

We use a more comprehensive representation of candidate solutions than proposed in section 4.3.3. In a strict sense, a solution of the topology optimization problem is given by the set E_{v} of active virtual links. This set is accordingly part of the solution representation. However, it does not uniquely specify a network configuration – and thus potentially its cost – since our demand routing and circuit adaptation procedures are not strictly deterministic due to arbitrary iteration orders and arbitrary selection among several shortest paths. We therefore include the sets of routed demands R and active circuits A in order to define a unique network configuration by the solution representation.

Due to the generally high number of perturbations performed during one optimization run, minimizing the computational effort for determining the network configuration for a new virtual topology is highly beneficial for the overall computation time. Incrementally adapting the configuration is one means to reduce this effort. In order to incrementally adapt demand routing, we need to retain blocking information in addition to the sets of routed demands. We thus define a solution of the optimization-heuristic method as the tuple:

$$X = \left\langle E_\mathrm{v}, R, A, [b_{ij}], [t_{\mathrm{B}\,ij}], b_\mathrm{VT}, t_\mathrm{B\,VT} \right\rangle \tag{4.60}$$

Therein, we require $E_\mathrm{v} \subseteq E_\mathrm{f}$, making use of the set of virtual links with feasible circuit realization according to physical path length as defined in Equation (4.35). This accelerates convergence of the solution by excluding disadvantageous parts of the solution space which necessarily bring about traffic blocking. Since the excluded solutions are also not required to move from one good solution candidate to another by several subsequent perturbations, the restriction is not detrimental to the heuristic search process.

4.4.3.2 Heuristic Topology Optimization

This section details two problem-specific functions for the SA based method: the creation of an initial solution and the perturbation. The third problem-specific operation, the computation of the cost of solutions which these two functions return, is performed according to the cost equations for the generic problem in section 4.3.5 when using the blocking cost according to Equation (4.60).

Algorithm 4.4 describes the creation of an initial solution. We set the virtual topology to the one realized by the circuits of the previous configuration. For two reasons, we expect the optimal solution to be close to the previous virtual topology, which makes the latter a good starting point for an efficient randomized search of the solution space. First, the reconfiguration cost term favors a small number of circuit modifications, and any change to the virtual topology necessitates the addition or removal of circuits. Second, significant changes between the demands in subsequent reconfiguration intervals are rare since the traffic is generally evolving in a steady manner.

Next, we route each node-to-node demand on its shortest path in terms of hops in the virtual topology, choosing an arbitrary path in case of several options. At this stage, we do not split demands but create one routed demand tuple $D_\mathrm{r} = \langle d_{sd}, \boldsymbol{R}_\mathrm{D} \rangle$ per non-zero demand, which is added to the respective routed demand set. If no path exists for a demand due to segmentation of the virtual topology, we cannot create a routed demand and instead increase the counters for the number b_VT and volume $t_\mathrm{B\,VT}$ of blocked demands.

We finally address the set A of active circuits. In order to minimize circuit reconfiguration, we initialize it by the set P of previously active circuits. Then, we iterate over all active virtual links $(i, j) \in E_\mathrm{v}$ and try to adapt the active circuits to the traffic load t_{ij} resulting from demand routing. In case we need fewer circuits than previously existing, we identify the ones to be torn down by means of the heuristic of section 4.4.1.2 and remove them from the active circuit set. If the capacity is insufficient, we try to allocate resources for additional circuits according to

Algorithm 4.4 Creation of initial solution

$E_\mathrm{v} \leftarrow \emptyset; R \leftarrow \emptyset; b_\mathrm{VT} \leftarrow 0; t_\mathrm{B\,VT} \leftarrow 0$
for all $(i,j) \in E_\mathrm{f}$ **do** {activate previously active virtual links}
  if $P_{ij} \neq \emptyset$ **then**
    $E_\mathrm{v} \leftarrow E_\mathrm{v} \cup \{(i,j)\}$
  end if
end for
for all $(s,d) \in V \times V$ such that $d_{sd} > 0$ **do** {route all demands}
  $\boldsymbol{R}_\mathrm{D} \leftarrow$ nodes along shortest path (in hops) from s to d in graph $\langle V, E_\mathrm{v} \rangle$
  if $\boldsymbol{R}_\mathrm{D} = \emptyset$ **then** {no route due to graph partitioning}
    $b_\mathrm{VT} \leftarrow b_\mathrm{VT} + 1$
    $t_\mathrm{B\,VT} \leftarrow t_\mathrm{B\,VT} + d_{sd}$
    $R_{sd} \leftarrow \emptyset$
  else
    $R_{sd} \leftarrow \{\langle d_{sd}, \boldsymbol{R}_\mathrm{D} \rangle\}$
  end if
  $R \leftarrow R \cup \{R_{sd}\}$
end for
$A \leftarrow P$
$E'_\mathrm{v} \leftarrow E_\mathrm{v}$
while $E'_\mathrm{v} \neq \emptyset$ **do** {adapt circuits on active virtual links to traffic load}
  $(i,j) \leftarrow \arg\min_{(i,j) \in E'_\mathrm{v}}$ length of shortest path from i to j in graph G_p
  $E'_\mathrm{v} \leftarrow E'_\mathrm{v} \setminus \{(i,j)\}$
  while $B\left|A_{ij}\right| - t_{ij} \geq B$ **do** {superfluous capacity on link}
    $C \leftarrow$ SELECTCIRCUITFORTEARDOWN(i,j)
    $A \leftarrow A \setminus \{C\}$
  end while
  $t_{\mathrm{B}\,ij} \leftarrow 0; b_{ij} \leftarrow 0$
  while $B\left|A_{ij}\right| < t_{ij}$ **do** {insufficient capacity on link}
    $C \leftarrow$ ALLOCATERESOURCESFORCIRCUIT(i,j)
    if $C = \emptyset$ **then** {no resources to realize circuit}
      $t_{\mathrm{B}\,ij} \leftarrow t_{ij} - B\left|A_{ij}\right|$
      $b_{ij} \leftarrow 1$
      break while $B\left|A_{ij}\right| < t_{ij}$
    else
      $A \leftarrow A \cup \{C\}$
    end if
  end while
end while
return $X \leftarrow \left\langle E_\mathrm{v}, R, A, [b_{ij}], [t_{\mathrm{B}\,ij}], b_\mathrm{VT}, t_\mathrm{B\,VT} \right\rangle$

section 4.4.1.1. If successful, we add the new circuits to the set of active ones and otherwise count the excessive traffic as blocked.

This iteration is done from shorter to longer virtual links in terms of the shortest-path realization (in hops) in the physical topology. This is expected to improve resource utilization in case of scarce fiber capacity: Being considered first, short circuits have a higher chance to be realized on their shortest paths. Circuits interconnecting more distant nodes are more likely to have several possible paths of similar length than shorter circuits. Hence, it is probable that an alternative path taken due to depletion of fiber capacity will not result in the occupation of significantly more resources. If sufficient resources are available, the iteration order is irrelevant.

The perturbation of a candidate solution consists of randomly adding or removing one link in the virtual topology and adapting the remaining configuration similarly to the creation of the initial solution. Algorithm 4.5 specifies this procedure. Unless the configuration allows no virtual link to be added or removed, in which case the respective opposite operation is performed, we start by randomly determining whether to add or remove a link. The latter is done with the link removal probability p_{LR}, the former with the complementary probability of $(1 - p_{\mathrm{LR}})$.

Algorithm 4.5 Perturbation of solution X

$r \leftarrow$ random number in $[0,1)$
if $(r < p_{\mathrm{LR}}$ **and** $E_{\mathrm{v}} \neq \emptyset$ $)$ **or** $E_{\mathrm{v}} = E_{\mathrm{f}}$ **then** {remove one virtual link}
  randomly select one $(i,j) \in E_{\mathrm{v}}$
  $E_{\mathrm{v}} \leftarrow E_{\mathrm{v}} \setminus \{(i,j)\}$
  for all $(s,d) \in V \times V$ such that {reroute demands from removed virtual link}
    $(\exists R_{sd} \in R) \wedge (\exists \langle t, [v_1,\ldots,v_{l+1}]\rangle \in R_{sd}, x \in \{1,\ldots,l\} : (v_x, v_{x+1}) = (i,j))$ **do**
    $R_{sd} \leftarrow \emptyset$
    $\boldsymbol{R}_{\mathrm{D}} \leftarrow$ nodes along shortest path (in hops) from s to d in graph $\langle V, E_{\mathrm{v}}\rangle$
    if $\boldsymbol{R}_{\mathrm{D}} = \emptyset$ **then** {no route due to graph partitioning}
      $b_{\mathrm{VT}} \leftarrow b_{\mathrm{VT}} + 1$
      $t_{\mathrm{B\,VT}} \leftarrow t_{\mathrm{B\,VT}} + d_{sd}$
    else
      $R_{sd} \leftarrow \{\langle d_{sd}, \boldsymbol{R}_{\mathrm{D}}\rangle\}$
    end if
  end for
else {add one virtual link}
  randomly select one $(i,j) \in E_{\mathrm{f}} \setminus E_{\mathrm{v}}$
  $E_{\mathrm{v}} \leftarrow E_{\mathrm{v}} \cup \{(i,j)\}$
  $b_{\mathrm{VT}} \leftarrow 0; t_{\mathrm{B\,VT}} \leftarrow 0$
  for all $(s,d) \in V \times V$ such that $d_{sd} > 0$ **do** {adapt demand routing}
    $\boldsymbol{R}_{\mathrm{D}} = [v_1,\ldots,v_{l+1}] \leftarrow$ nodes along shortest path (in hops) from s to d in graph $\langle V, E_{\mathrm{v}}\rangle$
    if $\boldsymbol{R}_{\mathrm{D}} = \emptyset$ **then** {no route due to graph partitioning}
      $R_{sd} \leftarrow \emptyset$
      $b_{\mathrm{VT}} \leftarrow b_{\mathrm{VT}} + 1$
      $t_{\mathrm{B\,VT}} \leftarrow t_{\mathrm{B\,VT}} + d_{sd}$
    else if $R_{sd} = \emptyset$ **or** $\left(R_{sd} = \left\{\left\langle \bar{t}, [\bar{v}_1,\ldots,\bar{v}_{\bar{l}+1}]\right\rangle\right\}$ **and** $\bar{l} > l\right)$ **then** {new or shorter route}
      $R_{sd} \leftarrow \{\langle d_{sd}, \boldsymbol{R}_{\mathrm{D}}\rangle\}$
    end if
  end for
end if

Continuation of Perturbation of solution X

```
E'_f ← E_f
Ẽ_v ← ∅                                                {virtual links requiring capacity upgrade}
while E'_f ≠ ∅ do                                      {adapt circuits to traffic load}
  (i, j) ← argmin_{(i,j)∈E'_f} length of shortest path from i to j in graph G_p
  E'_f ← E'_f \ {(i, j)}
  t_Bij ← 0; b_ij ← 0
  while B|A_ij| − t_ij ≥ B do                          {superfluous capacity on link}
    C ← SelectCircuitForTeardown(i, j)
    A ← A \ {C}
  end while
  if B|A_ij| < t_ij then                               {insufficient capacity on link}
    Ẽ_v ← Ẽ_v ∪ {(i, j)}
  end if
end while
while Ẽ_v ≠ ∅ do
  (i, j) ← argmin_{(i,j)∈Ẽ_v} length of shortest path from i to j in graph G_p
  Ẽ_v ← Ẽ_v \ {(i, j)}
  while B|A_ij| < t_ij do
    if P_ij \ A_ij ≠ ∅ then                            {circuits for reactivation exist}
      C ← SelectCircuitForReactivation(i, j)
    else
      C ← AllocateResourcesForCircuit(i, j)
    end if
    if C = ∅ then                                      {no resources to realize circuit}
      t_Bij ← t_ij − B|A_ij|
      b_ij ← 1
      break while B|A_ij| < t_ij
    else
      A ← A ∪ {C}
    end if
  end while
end while
return X' ← ⟨E_v, R, A, [b_ij], [t_Bij], b_VT, t_BVT⟩
```

To deactivate a link, we randomly select one of the active virtual links with equal probability and remove it from the set of active links. Since this operation does not improve the routing possibilities for any demand but turns some demand routes infeasible, it is sufficient to reroute the demands previously using the deactivated link. For each of these demands, we determine the shortest path in the remaining virtual topology, replace the previous route by this path if one exists, or otherwise increase the demand blocking values. It is worth noting that we must not reset the blocking values first as they may account for blocking already existent prior to the perturbation.

To activate a link, we randomly select one of the inactive feasible links with equal probability and add it to the set of active links. The additional link may enable a shorter route for any demand, and it may allow the routing of previously blocked demands. We therefore reset the demand blocking values and compute the shortest path for each non-zero demand in the new virtual topology. If we find a path for a previously blocked demand or a shorter path for an already routed one, we update the routed demand entry. Otherwise, we increase the demand blocking counters.

Next, we adapt the active circuit configuration to the changed traffic load. Hereby, the nature of the perturbation requires some extensions compared to the creation of an initial solution. First, we need to eliminate circuits on deactivated links, which no longer carry traffic. We hence iterate over the set of all feasible virtual links instead of the set of active links, again in the order of increasing shortest physical path length for the same reasons. If the circuit capacity exceeds demand, we deactivate circuits analogously to the creation of an initial solution. The heuristic from section 4.4.1.2 thereby preferentially selects circuits not present in the previous configuration, i. e. circuits newly created for the initial solution or during a previous perturbation. Unlike in the case of previously existing circuits, we can immediately release the resources allocated for these circuits and reuse them for different new circuits.

This potential reuse of resources is the reason why we do not try to create additional circuits on links with insufficient capacity while iterating over all feasible links, but only retain them in the set $\tilde{E}_\mathrm{v}$ while eliminating all unneeded circuits. Next, we iterate over this set in the same order and try to increase the circuit capacity on these links. If possible, we undo the deactivation of circuits of the previous configuration, since this does not bind any additional resources. In case, we select among several candidates by the heuristic of section 4.4.1.3. We otherwise try to allocate resources for new circuits, accordingly updating the set of active circuits or the link blocking values, which have previously been reset. This concludes the perturbation and we return the resulting candidate solution.

4.4.3.3 Control of the Heuristic Optimization Procedure

Algorithm 4.6 describes our realization of the SA procedure, which controls the randomized search starting from the initial solution and making use of perturbations. We apply a geometric temperature function and an adaptive temperature length and base the termination condition on the evolution of the cost of accepted solutions.

We start by setting the temperature ϑ of the annealing process to a fixed initial temperature ϑ_0, and we multiply the temperature by a cooling factor γ upon each cooling step. Similarly to [116], we let the number of perturbations between these steps depend on the acceptance behavior: We reduce the temperature after performing ν_max perturbations or accepting α_max solutions, whichever happens earlier. The variables ν and α count the respective events. In the study in [42], this acceptance-based dynamic temperature length proved to significantly reduce the optimization time compared to a fixed temperature length without affecting the quality of the outcome. We use the solution acceptance condition of the basic simulated annealing algorithm discussed in section 2.5.1.

We aim at terminating the optimization procedure when finding a significantly better solution by leaving a local optimum becomes unlikely. We assume that this condition is met when the cost of accepted solutions ceases to change noticeably. To express this termination condition,

Algorithm 4.6 SA based main algorithm

```
ϑ ← ϑ_0; η ← 0; η_r ← 0; c_max ← 0; c_min ← ∞          {initialize control variables}
X̂ ← X ← CreateInitialSolution(P, D)                   {X̂: best encountered solution}
ĉ ← c ← cost of X                                     {X: currently accepted solution}
loop
  ν ← 0; α ← 0                                        {reset control variables}
  while ν < ν_max and α < α_max do                    {dynamic temperature length not reached}
    ν ← ν + 1; η ← η + 1; η_r ← η_r + 1                {update control variables}
    X' ← Perturbate(X)                                {X': new candidate solution}
    c' ← cost of X'
    r ← random number in [0, 1)
    if c' ≤ c or r < exp(−(c' − c)/ϑ) then            {accept candidate solution}
      if c' < c then                                  {improvement of accepted solution cost}
        η ← 0
      end if
      α ← α + 1
      X ← X'; c ← c'
      c_max ← max{c_max, c}
      c_min ← min{c_min, c}
    end if
    if c' < ĉ then                                    {retain new best solution}
      X̂ ← X'; ĉ ← c'
    end if
    if η = η_max then                                 {terminate due to lack of cost improvement}
      break loop
    end if
    if η_r = η_max then
      if (c_max − c_min)/c_min < ρ then               {terminate due to small accepted cost range}
        break loop
      end if
      η_r ← 0; c_max ← 0; c_min ← ∞                   {reset control variables}
    end if
  end while
  ϑ ← ϑ · γ                                           {annealing}
end loop
return X̂
```

we record the minimum and maximum cost $c_{\min}$ and $c_{\max}$ of solutions accepted in intervals of $\eta_{\max}$ perturbations. We terminate if the relative difference between these values, i. e. the relative cost range of accepted solutions, is less than ρ. We likewise abort if no improvement of the cost of the accepted solution has occurred during the last $\eta_{\max}$ perturbations. The variables η_r and η serve for counting the respective perturbations.

The Bachelor thesis project documented in [42] compared different adaptive control schemes for simulated annealing. It found our algorithm to perform similarly to the more sophisticated scheme proposed in [41] in trading off optimization time and cost. In addition, our algorithm proved more robust with regard to parameter settings.

4.4.3.4 Post-Processing Algorithm

The post-processing of the solution obtained by heuristic optimization aims at resolving two issues caused by shortest-path demand routing: traffic blocking while capacity is available on an alternative path and lowly utilized circuits. It does so by rerouting traffic in a first-fit manner, i. e. it uses the first feasible alternative route.

For this purpose, the algorithm iterates over all active virtual links as described in Algorithm 4.7. Since traffic blocking is a more severe issue than energetically suboptimal configurations, we first consider links with insufficient capacity. In this way, we preferentially use available spare capacity to accommodate previously blocked traffic. The remaining spare capacity afterwards serves for rerouting traffic to free circuits and save energy. For reasons similar to those in section 4.4.3.2, we consider the links with insufficient capacity in increasing order of their shortest physical realization: With increasing distance of the end points of the link, we expect more alternative routes which do not incur a significant detour. We thus start with the links presumably having a smaller number of viable circuit routes.

In the second phase, the algorithm considers the set of links with sufficient capacity. Here, we opt for the opposite iteration order, starting with the longest virtual links according to their physical realization. By doing so, we intend to accumulate traffic not efficiently transported by circuits on long links on a set of next shorter links, where it may fill partially used circuits. In subsequent iteration steps, the traffic may then be further rerouted to shorter links to eliminate more circuits.

As discussed below, rerouting may reduce the traffic load on links other than the currently considered one. Likewise, the removal of circuits turned unnecessary by rerouting may free resources and in turn enable the establishment of additional circuits. For these reasons, we try to adapt the capacity of each virtual link to the actual traffic load prior to starting the rerouting procedure. I. e., we first try allocating resources and setting up circuits on links with insufficient capacity, and we tear down circuits in case of excess capacity. Depending on whether blocking occurs after these adaptations, we continue by executing the respective variant of the rerouting heuristic according to Algorithm 4.8 or Algorithm 4.9, respectively. In case additional circuits or rerouting have resolved blocking, we then reset the blocking indicator of the link. Since rerouting may also have deviated traffic from already considered links, we finally reiterate over all active virtual links in order to eliminate superfluous circuits.

The procedure to resolve blocking on a link (i, j), specified in Algorithm 4.8, uses any alternative route it encounters and reroutes the share of the blocked traffic this route can accommodate to it. At first, the set of candidate links for alternative paths $\tilde{E}_{\mathrm{v}}$ is defined as the set of active virtual links excluding the currently considered one. As the alternative route, we select the shortest path in terms of hops from node i to node j in the graph supported by $\tilde{E}_{\mathrm{v}}$.

If such a path exists, we verify whether the spare capacity of existing circuits on each of its links suffices to carry the blocked (excess) traffic volume t_{e}. If not, we try to increase the capacity of

Algorithm 4.7 Post-processing of solution X

$E'_\text{v} \leftarrow E_\text{v}$
while $E'_\text{v} \neq \emptyset$ **do** {iterate over active virtual links}
  $\tilde{E}_\text{v} \leftarrow \{(i,j) \in E'_\text{v} \,|\, B\,|A_{ij}| < t_{ij}\}$ {links with insufficient capacity}
  if $\tilde{E}_\text{v} \neq \emptyset$ **then**
    $(i,j) \leftarrow \arg\min_{(i,j)\in\tilde{E}_\text{v}}$ length of shortest path from i to j in graph G_p
  else
    $(i,j) \leftarrow \arg\max_{(i,j)\in E'_\text{v}}$ length of shortest path from i to j in graph G_p
  end if
  $E'_\text{v} \leftarrow E'_\text{v} \setminus \{(i,j)\}$
  $C \leftarrow$ AllocateResourcesForCircuit(i,j)
  while $C \neq \emptyset$ **and** $t_{ij} > B\,|A_{ij}|$ **do** {try setting up additional circuits if needed}
    $A \leftarrow A \cup \{C\}$
    $t_{\text{B}\,ij} \leftarrow \max\{0; t_{\text{B}\,ij} - B\}$
    $C \leftarrow$ AllocateResourcesForCircuit(i,j)
  end while
  while $B\left(|A_{ij}| - 1\right) \geq t_{ij}$ **do** {remove superfluous circuits}
    $C \leftarrow$ SelectCircuitForTeardown(i,j)
    $A \leftarrow A \setminus \{C\}$
  end while
  if $B\,|A_{ij}| < t_{ij}$ **then** {traffic blocked on considered link}
    PostprocessTrafficBlockingVirtualLink(i,j)
  else
    PostprocessNonblockingVirtualLink(i,j)
  end if
  if $B\,|A_{ij}| \geq t_{ij}$ **then** {sufficient capacity on considered link}
    $b_{ij} \leftarrow 0$
  end if
end while
for all $(i,j) \in E_\text{v}$ **do** {eliminate superfluous circuits on all active virtual links}
  while $B\,(|A_{mn}| - 1) \geq t_{ij}$ **do**
    $C \leftarrow$ SelectCircuitForTeardown(i,j)
    $A \leftarrow A \setminus \{C\}$
  end while
end for

the link by allocating resources for an additional circuit. If this leaves some link without any spare capacity, we try the next alternative path without the depleted link. For this purpose, we remove the link from $\tilde{E}_\text{v}$ and release the resources previously allocated for additional circuits. Otherwise, we proceed by rerouting a traffic volume t_r corresponding to the minimum of the spare capacity along the path or the excess traffic (whichever is smaller) to the candidate path. Finally, we undo the resource allocation for new circuits not needed due to tighter capacity limits on other links. If we cannot reroute the blocked traffic entirely, we continue the search for an alternative path for the remaining excess traffic. In this case, we initially reconsider the same path and eliminate the most limiting link, of which the resources are now depleted.

Algorithm 4.8 Post-processing of traffic-blocking virtual link (i, j)

$t_\mathrm{e} \leftarrow t_{ij} - B\left|A_{ij}\right|$ {blocked excess traffic}
$\tilde{E}_\mathrm{v} \leftarrow E_\mathrm{v} \setminus \{(i,j)\}$ {candidate links for alternative path}
BlockingAvoidanceLevel $\leftarrow 0$ {control variable for extending candidate link set}
repeat
  $\boldsymbol{R}_\mathrm{D} = [v_1, \ldots, v_{l+1}] \leftarrow$ nodes along shortest path (in hops) from i to j in graph $\langle V, \tilde{E}_\mathrm{v}\rangle$
  if $\boldsymbol{R}_\mathrm{D} = \emptyset$ **then** {no path found (resource depletion)}
    if BlockingAvoidanceLevel $= 0$ **then**
      $\tilde{E}_\mathrm{v} \leftarrow (E_\mathrm{v} \cup \{(\bar{j},\bar{i}) \in V \times V \mid (\bar{i},\bar{j}) \in E_\mathrm{v}\}) \setminus \{(i,j)\}$ {allow inverse links}
      BlockingAvoidanceLevel $\leftarrow 1$
      continue
    else if BlockingAvoidanceLevel $= 1$ **then**
      $\tilde{E}_\mathrm{v} \leftarrow E_\mathrm{f} \setminus \{(i,j)\}$ {allow all feasible links}
      BlockingAvoidanceLevel $\leftarrow 2$
      continue
    else
      break repeat {no further rerouting possible}
    end if
  end if
  $t_\mathrm{r} \leftarrow t_\mathrm{e}$ {traffic reroutable to $\boldsymbol{R}_\mathrm{D}$}
  $A_\mathrm{a} \leftarrow \emptyset$ {set of additional circuits}
  for $x = 1$ **to** l **do**
    if $t_\mathrm{r} + t_{v_x v_{x+1}} > B\left|A_{v_x v_{x+1}}\right|$ **then** {too little capacity on alternative path link}
      $C \leftarrow$ SELECTCIRCUITFORREACTIVATION(v_x, v_{x+1})
      if $C = \emptyset$ **then**
        $C \leftarrow$ ALLOCATERESOURCESFORCIRCUIT(v_x, v_{x+1})
      end if
      if $C \neq \emptyset$ **then**
        $A \leftarrow A \cup \{C\}$; $A_\mathrm{a} \leftarrow A_\mathrm{a} \cup \{C\}$
      end if
    end if
    if $t_{v_x v_{x+1}} \geq B\left|A_{v_x v_{x+1}}\right|$ **then** {capacity on alternative path link depleted}
      $\tilde{E}_\mathrm{v} \leftarrow \tilde{E}_\mathrm{v} \setminus \{(v_x, v_{x+1})\}$
      $A \leftarrow A \setminus A_\mathrm{a}$ {release resources of additional circuits}
      continue repeat {check next alternative path}
    end if
    $t_\mathrm{r} \leftarrow \min\left\{t_\mathrm{r}; B\left|A_{v_x v_{x+1}}\right| - t_{v_x v_{x+1}}\right\}$
  end for
  REROUTETRAFFICVOLUMEFROMVIRTUALLINKTOROUTE$(t_\mathrm{r}, (i,j), \boldsymbol{R}_\mathrm{D})$
  $t_\mathrm{e} \leftarrow t_\mathrm{e} - t_\mathrm{r}$; $t_{\mathrm{B}\,ij} \leftarrow t_{\mathrm{B}\,ij} - t_\mathrm{r}$
  for $((i', j') \in A_\mathrm{a}$ **do**
    if $B\left(\left|A_{i'j'}\right| - 1\right) \geq t_{i'j'}$ **then**
      $A \leftarrow A \setminus \{(i', j')\}$ {eliminate unneeded circuits}
    else
      $E_\mathrm{v} \leftarrow E_\mathrm{v} \cup \{(i', j')\}$ {activate link in case this is its first circuit}
    end if
  end for
until $t_\mathrm{e} = 0$

In case the elimination of capacity-depleted links leaves no path from node i to node j, we extend the set of candidate links $\tilde{E}_\mathrm{v}$ as described in the following and repeat the previously described procedure. In a first step, we add the inverse links to all active virtual links, allowing the use of line card ports of port pairs supporting circuits on unidirectionally active links. This constitutes only a minor modification to the topology determined by the heuristic optimization and allows using resources otherwise blocked due to the port pair constraint. If this does not suffice to resolve blocking, we next allow circuits on all feasible virtual links. Traffic not reroutable in this setting remains blocked. If we reroute traffic to an inactive virtual link and establish a circuit on it, we add this link to the set of active ones. This allows the rerouting of additional traffic to it in subsequent iterations.

The rerouting of traffic to avoid lowly loaded circuits according to Algorithm 4.9 corresponds to that for blocking resolution except for three aspects: it exclusively uses the active links determined by the optimization step, it reroutes excess traffic only to a single path of sufficient capacity, and it only does so if advantageous in terms of optimization goals. By excess traffic t_e, we now designate the share of traffic on the considered link not filling a complete circuit.

We define the cost c_{C_o} of the partially filled circuit C_o we aim at eliminating by rerouting as the difference in the cost of the network configuration including and excluding this circuit, respectively. In doing so, we only count the direct contribution of the circuit in terms of energetic cost of ports, line cards, and chassis as well as change cost, disregarding indirect effects like resulting traffic blocking or increased packet processing on alternative paths. We account for the latter when evaluating a specific alternative path in Algorithm 4.9.

Based on this definition of the cost of a circuit, we reroute excess traffic from a virtual link (i, j) by the following procedure formalized in Algorithm 4.9: First, we identify the potentially obsolete circuit C_o on the link by means of the heuristic of section 4.4.1.2, compute its cost c_{C_o} and remove it from the set of active circuits in order to reuse its resources if it is being established. We start the search for alternative routes in the set $\tilde{E}_\mathrm{v}$ of all active virtual links excluding (i, j). As a candidate alternative route, we select the shortest path in terms of hops from node i to node j in the graph supported by $\tilde{E}_\mathrm{v}$.

Next, we verify the capacity along this path and the cost c_a of rerouting the excess traffic to it. The cost is initialized by that of traffic processing in the transit nodes. Then, the capacity of each link is checked: If necessary and feasible, resources for an additional circuit are allocated and the path cost is increased by the cost of this circuit, which we define analogously as the difference in network cost excluding and including this additional circuit. If the spare capacity of the link remains insufficient or the accumulated cost of the alternative path exceeds that of the circuit to be replaced, we proceed by considering a next alternative route without this link, i. e. we repeat the procedure after removing the link from $\tilde{E}_\mathrm{v}$. If we cannot find another route from i to j due to the elimination of links, we abort without rerouting the excess traffic and maintain the circuit C_o we intended to replace, adding it again to the set of active circuits.

Algorithm 4.10 finally describes the procedure of moving a certain traffic volume t_r from the virtual link (i, j) to a given route. This procedure is used by both variants of the link post-processing algorithm according to Algorithm 4.8 and 4.9, respectively, after finding an alternative route.

Algorithm 4.9 Post-processing of non-blocking virtual link (i, j)

```
t_e ← t_ij − B(|A_ij| − 1)                                    {excess traffic for rerouting}
C_o ← SelectCircuitForTeardown(i, j)
c_Co ← cost of C_o
A ← A \ {C_o}                          {de-allocate resources in case C_o is being established}
Ẽ_v ← E_v \ {(i, j)}                                   {candidate links for alternative path}
repeat
  R_D = [v_1, ..., v_{l+1}] ← nodes along shortest path (in hops) from i to j in graph ⟨V, Ẽ_v⟩
  if R_D = ∅ then                                    {no path with sufficient spare capacity}
    A ← A ∪ {C_o}                        {undo circuit deactivation since traffic not rerouted}
    break repeat
  end if
  A_a ← ∅                                                          {set of additional circuits}
  c_a ← (l − 1) · t_e · P_T                 {cost of rerouting, initialized by transit traffic processing}
  if c_a ≥ c_Co then                      {rerouting would be more expensive than obsolete circuit}
    A ← A ∪ {C_o}                        {undo circuit deactivation since traffic not rerouted}
    break repeat
  end if
  for x = 1 to l do
    if t_e + t_{v_x v_{x+1}} > B |A_{v_x v_{x+1}}| then      {too little capacity on alternative-path link}
      C ← SelectCircuitForReactivation(v_x, v_{x+1})
      if C = ∅ then
        C ← AllocateResourcesForCircuit(v_x, v_{x+1})
      end if
      if C ≠ ∅ then
        c_a ← c_a + cost of C
        A ← A ∪ {C}; A_a ← A_a ∪ {C}
      end if
      if C = ∅ or c_a ≥ c_Co then   {required additional circuit not feasible or too expensive}
        Ẽ_v ← Ẽ_v \ {(v_x, v_{x+1})}
        A ← A \ A_a                                   {release resources of additional circuits}
        continue repeat
      end if
    end if
  end for
  RerouteTrafficVolumeFromVirtualLinkToRoute(t_e, (i, j), R_D)
until true
```

In order to move traffic between links, we have to manipulate the sets of routed demands. We therefore first establish the set $\tilde{R}_{ij}$ of demands routed over the link (i, j). We then reroute arbitrary demands from this set to the alternative path until the desired traffic volume is reached, possibly splitting the last demand into fractions respectively remaining on (i, j) and being moved to the new path.

The shortest-path routing of demands during the heuristic optimization ensures loop-free routes, which are desirable to minimize traffic processing and resource utilization. In order to maintain

Algorithm 4.10 Rerouting of traffic volume t_r from virtual link (i,j) to route $\left[\bar{v}_1,\ldots,\bar{v}_{\bar{l}+1}\right]$

$\tilde{R}_{ij} \leftarrow \bigcup_{R_{sd}\in R}\{\langle t,[v_1,\ldots,v_{l+1}]\rangle \in R_{sd} \mid \exists x \in \{1,\ldots,l\} : (v_x,v_{x+1}) = (i,j)\}$
while $t_\mathrm{r} > 0$ **do**
  select one $D_\mathrm{r} = \langle t,[v_1,\ldots,v_{l+1}]\rangle$ from $\tilde{R}_{ij}$
  $\tilde{R}_{ij} \leftarrow \tilde{R}_{ij} \setminus \{D_\mathrm{r}\}$
  $p_i \leftarrow \min\left\{p \in \{1,\ldots,l\} \mid v_p = i\right\}$; $p_j \leftarrow \max\left\{p \in \{1,\ldots,l\} \mid v_p = j\right\}$
  $x \leftarrow \min\left\{p \in \{1,\ldots,p_i\} \mid \exists \bar{p} \in \{1,\ldots,\bar{l}\} : v_p = \bar{v}_{\bar{p}}\right\}$
  $\bar{x} \leftarrow \max\left\{p \in \{1,\ldots,\bar{l}\} \mid \bar{v}_p = v_x\right\}$
  $y \leftarrow \max\left\{p \in \{p_j,\ldots,l\} \mid \exists \bar{p} \in \{\bar{x},\ldots,\bar{l}\} : v_p = \bar{v}_{\bar{p}}\right\}$
  $\bar{y} \leftarrow \min\left\{p \in \{\bar{x},\ldots,\bar{l}\} \mid \bar{v}_p = v_y\right\}$
  $R_{v_1 v_{l+1}} \leftarrow R_{v_1 v_{l+1}} \setminus \{D_\mathrm{r}\} \cup \left\{\left\langle \min\{t_\mathrm{r},t\}, \left[v_1,\ldots,v_x,\bar{v}_{\bar{x}+1},\ldots,\bar{v}_{\bar{y}},v_{y+1},\ldots,v_{l+1}\right]\right\rangle\right\}$
  if $t_\mathrm{r} \geq t$ **then**
    $t_\mathrm{r} \leftarrow t_\mathrm{r} - t$
  else
    $R_{v_1 v_{l+1}} \leftarrow R_{v_1 v_{l+1}} \cup \{\langle t - t_\mathrm{r},[v_1,\ldots,v_{l+1}]\rangle\}$
    $t_\mathrm{r} \leftarrow 0$
  end if
end while

this property when rerouting, we cannot simply replace the link (i,j) in a demand route by the alternative route, since the alternative route may contain additional nodes also present in the original demand route. We thus determine the index x of the first node of the original demand route that also occurs in the alternative route (at index $\bar{x}$). Likewise, we identify the index y of the last node of the original demand route coinciding with a node of the alternative route (at index $\bar{y}$). We then obtain a loop-free new demand route only using subsets of the original and the alternative routes by inserting the section from the $\bar{x}$-th to the $\bar{y}$-th node of the alternative route into the original demand route, replacing its section from the x-th to the y-th node. If both routes have no nodes other than i and j in common, this procedure replaces the link (i,j) in the demand route by the alternative route.

5 Evaluation

This chapter evaluates network settings periodically determined by our reconfiguration methods presented in section 4.4 relative to those obtained by a reference resource scaling scheme. In doing so, we focus on the requirements stated in section 4.2, which are reflected in our objective function derived in section 4.3.5. Since energy savings motivate our reconfiguration problem, we primarily consider the power consumption of the network, optionally including the transient consumption during reconfiguration. In addition, we evaluate the potential impact on network operation in terms of circuit modifications and traffic blocking. By providing the gaps between the primal solution and the dual bound for the MILP-based solution method, we give an indication of the quality of the configurations our methods determine relative to an optimal solution of our reconfiguration problem.

In section 5.1, we present and motivate the evaluation scenario in terms of network topology, resource dimensioning, traffic, and power model parameters. We discuss the value ranges of the parameters controlling the solution methods in section 5.2, define the reference resource scaling scheme in section 5.3, and outline the simulation setup in section 5.4. On this basis, we present and analyze our results in section 5.5. Section 5.6 concludes this chapter with a discussion of the main observations.

5.1 Evaluation Scenario

Network reconfiguration implies specific conditions for finding a new configuration. In particular, the traffic load generally evolves smoothly over time, resulting in limited traffic changes between subsequent network configurations. Network reconfiguration methods shall find suitable configurations under such conditions. Accordingly, we aim at evaluating our methods in realistic reconfiguration scenarios. We therefore refrain from stochastically generating arbitrary problem instances, as it is done for the evaluation of heuristic optimization methods e. g. in [43]. Instead, we resort to reference network topologies widely employed in network optimization research and traffic traces collected in real networks.

Network *re*configuration is characterized by the existence of a previous configuration influencing the feasibility of a new configuration and the effort to reach it. Generally, the previous configuration is itself influenced by anterior settings, and it is thus not necessarily optimal in terms of the reconfiguration objective. In order to evaluate the solutions found by our reconfiguration methods under such conditions, we simulate series of periodic reconfigurations, using the configuration obtained for one period as the previous setting in the next period.

In section 5.1.1, we address the reference network topologies and traffic data which we use for the evaluation of our reconfiguration methods. Section 5.1.2 then describes the procedure to determine a resource dimensioning for these networks. We finally present the parameters of our dynamic resource and power model in section 5.1.3.

5.1.1 Network Topologies and Traffic

The temporal traffic variations are of particular importance for the realism of a network reconfiguration scenario. To the best of our knowledge, scientific literature does unfortunately not propose a suitable dynamic model for aggregated traffic in transport networks. Recent traffic data from commercial networks is likewise not available since considered as a business secret by network operators. Like many researchers working on network reconfiguration [21, 26, 56, 72, 83, 117], we therefore resort to publicly available traffic traces derived from measurements in national and transnational research and educational networks (REN). This restricts the study to a set of reference network topologies for which such traces are available.

Table 5.1: Reference network characteristics and available traffic traces [118]

reference network		demand trace	
name	nodes	length	granularity
Abilene	12	5 months	5 min
Nobel-Germany	17	1 day	5 min
Géant	22	4 months	15 min
Germany50	50	1 day	5 min

Table 5.1 lists the four REN-based network topologies we use in the evaluation along with the numbers of their nodes and the duration and temporal granularity of the respective traffic traces. These topologies and traces are provided in the SNDLib [118]; the extraction of the Géant traffic traces is detailed in [119]. The sizes of these topologies in terms of the number of their nodes cover the range typically applied in the core network reconfiguration studies surveyed in section 3.2.3. Researchers have proposed larger reference networks of up to 1000 nodes, e. g. within the STRONGEST project [106]. However, architectural concepts for future commercial core networks discussed by network operators – like *TeraStream* of Deutsche Telekom – likewise assume tens of core nodes [120].

Being based on actual REN, the reference networks come with real-world coordinates for their nodes. We define the length of a physical link as the air-line distance of its end nodes, and we limit the length of multi-hop circuits by a typical maximum optical reach of $L = 3000\,\text{km}$. As indicated in section 4.3.4, we allow circuits on longer physical links, implicitly assuming 3R regeneration. While the set of physical links defines the geographical length and thus the feasibility of a circuit, it has no further influence on possible network configurations or their cost. Consequently, the nodal degree of the reference networks is irrelevant to the performance of our reconfiguration methods.

According to accompanying documentation, all traffic traces are derived from accounting data recorded in REN in the years 2004 and 2005. If a reference topology has less nodes than the REN it is based upon, traffic from and to missing nodes is attributed to the geographically closest node. In some cases, the demand values are scaled by an unknown factor for anonymization. The traces of the Abilene and Géant networks, which span several months, include periods without any data. At some additional time instants, the data is known to be erroneous. For the evaluation, we use an error-free period of 14 days from each of these traces. The selection of a short section of the traces is additionally motivated by constraints on computation resources and time. By selecting a multiple of one week, we ensure that the energy savings we obtain correctly represent potential differences in the traffic profile between working days and weekends. For the Nobel-Germany and the Germany50 networks, where a detailed trace is only available for one day, we repeatedly concatenate this single-day profile to likewise obtain a 14 day trace. We therewith obtain a strictly periodic trace, which is not a realistic model for the traffic evolution over two weeks. However, it allows us to compute as many network configurations as for the former two network topologies, which reduces statistical variations of the evaluation metrics for the stochastic SA-based method. For the exact solution method, we only use two consecutive instances of the single-day profile.

In accordance with the temporal granularity of the coarsest of the demand traces, we set the reconfiguration interval to $\Delta T = 15\,\text{min}$. This is also in line with the technological constraints we assume in section 4.1.3. For traces of a finer temporal granularity, we take the maximum of the demand values in the reconfiguration interval.

At the beginning of the simulation, we need to compute a first configuration without the constraints implied by a previous setting. We do so by assuming no resource occupation by pre-existing circuits and setting the reconfiguration penalty to $\delta = 0$. Since this situation is untypical for network reconfiguration, we exclude this configuration along with the results for the following four reconfiguration intervals (corresponding to the first hour) from the evaluation. By then, we expect the network configuration to be sufficiently influenced by previous settings for a realistic reconfiguration situation.

As observed in [21], the average traffic load influences network reconfiguration in addition to the relative load variations given by the demand traces. More precisely, the ratio of the traffic demands to the capacity of an optical circuit is decisive for the efficiency of and effort for energy-oriented reconfiguration. Like the authors of [62], we therefore express traffic values relative to the capacity of one circuit, and we refer to this unit by *circuit equivalent*. We evaluate our reconfiguration methods at different traffic load points, which we generate by scaling all demand values of a trace by one factor. To quantify this scaling, we make use of the peak demand matrix $\boldsymbol{D}_{\text{peak}}$ containing the maxima of the individual demands over the 14-day trace, i. e.

$$\boldsymbol{D}_{\text{peak}} = \left(d_{\text{peak}\,sd}\right)_{s,d\in V} = \left(\max_t d_{sd}(t)\right) \quad , \tag{5.1}$$

where $d_{sd}(t)$ is the demand value between node s and node d in the reconfiguration interval starting at time t. We designate a load point by the average over all non-zero entries of the peak demand matrix:

$$\bar{d}_{\text{peak}} = \frac{\sum_{s,d\in V} d_{\text{peak}\,sd}}{\left|\left\{(s,d) \in V \times V \,|\, d_{\text{peak}\,sd} \neq 0\right\}\right|} \tag{5.2}$$

For all network topologies, we scale the traces such that this average peak demand $\bar{d}_{\text{peak}}$ varies between 0.1 and 2.0 circuit equivalents. This range is motivated as follows: Values lower than 0.1 circuit equivalents result in minimal topologies assuring connectivity with little room for reconfiguration [111]. Values higher than 2.0 circuit equivalents require unrealistically high port numbers in network nodes.

5.1.2 Dimensioning of Available Resources

The reconfiguration problem defined in section 4.3 applies explicit constraints on two types of resources: the number of port pairs $n_{\text{PP}v}$ on each node $v \in V$ and the number of fibers $n_{\text{F}ij}$ on each physical link $(i, j) \in E_{\text{p}}$. The number of hierarchical node components like line cards and chassis is implicitly given by that of port pairs. In order to impose resource constraints in the simulated series of reconfigurations, we need reasonable assumptions on the amount of installed resources.

We assume network resources to be dimensioned as required under the current paradigm of static network operation. Since QoS requirements make blocking intolerable in core networks, the installed capacity needs to suffice for all demands in a static configuration. To obtain an according network dimensioning, we compute an energetically optimized network configuration for the peak demand matrix $\boldsymbol{D}_{\text{peak}}$ and assume the resources used in this configuration to be installed. We do so by a variant of the heuristic solution method according to section 4.4.3 which disregards resource constraints and the previous configuration. In exchange, we apply an extended power model considering the consumption of fabric card shelves and additional fabric cards in case of multi-chassis configurations. Minimizing the power consumption of the installed resources instead of CAPEX corresponds to future green network planning. However, it is likely to yield a similar configuration since the complexity of components generally defines both their power consumption and cost [64, 102].

Dimensioning resources for the peak demand matrix may not provide the minimal resource configuration satisfying all demands since the peaks of different demands do generally not occur simultaneously. However, it is in line with overdimensioning commonly done in core networks due to demand uncertainty. Following the same rationale, other researches define a network reconfiguration scenario in the same way [21]. We evaluate the sensitivity of the performance of the reconfiguration methods to the available resources by considering networks dimensioned for peak demand matrices scaled by factors of $\sigma_{\text{dim}} \in \{0.8, 2.0\}$ in addition to the basic dimensioning corresponding to $\sigma_{\text{dim}} = 1.0$. While the first case sheds some light on the benefit of reconfiguration in mitigating resource shortages, the second one mimics a situation where spare resources provided for resilience are available for reconfiguration.

5.1.3 Parameterization of Resource and Power Model

To quantify the parameters of the resource and power model defined in section 4.1.2, we assume Cisco equipment[1] with short-reach interfaces and 40 Gbps TXPs [15]: One line card features up to $N_{\text{PP}} = 3$ port pairs and one line card chassis contains up to $N_{\text{LC}} = 16$ line cards. If a

[1] We refer to Cisco equipment due to the public availability of data from this vendor.

node consists of more than one line card chassis, one fabric card chassis is needed, and a set of interconnecting fabric cards is installed per group of three line card chassis.

We use normalized power consumption values since we are interested in relative energy saving figures. This choice has the additional benefit of making our results less dependent on current technology, since the ratios between the power consumption of different node components is arguably less sensitive to technological progress than the absolute consumption. We normalize all power values to the consumption of a port pair. As we assume the ports of one TXP to be separately activatable, the according power value of 1.0 also applies to one unidirectional optical circuit excluding hierarchical components. For the latter components, we select integer power values approximating the ratios reported for the above-mentioned Cisco configuration in [15]. Table 5.2 lists the resulting values for the hierarchical power model. For resource dimensioning, we additionally apply a cost of 20.0 for the fabric cards interconnecting up to three line card chassis. Electrically processing one circuit equivalent of transit traffic in one node comes at an energetic cost of $P_T = 10^{-4}$.

Table 5.2: Normalized power consumption of node components

		power model	
component	symbol	hierarchical	flat
port	P_P	0.5	1.16666666667
line card	P_{LC}	3.0	0.0
line card chassis	P_{LCC}	16.0	0.0

In order to control the influence of the parameterization of the hierarchical resource and power model on the effects of reconfiguration, we additionally use a power model which disregards the node component hierarchy. To maintain the relation between energy terms and reconfiguration and blocking terms in the cost function, we attribute the power consumption of line cards and chassis to ports. We do so by distributing this power consumption among the maximum number of ports supported by the respective components. The resulting values for this flat power model are also listed in Table 5.2.

5.2 Algorithmic Parameters

The parameters detailed in the previous section define the network scenario in which we want to study reconfiguration in terms of physical topology, traffic, component hierarchy, and power consumption. In addition, we need to specify some parameters intrinsic to the reconfiguration problem stated in section 4.3, namely the cost coefficients for circuit modifications and blocking. Finally, we also have to fix a number of parameters controlling our two solution methods. We address these additional parameters in the following.

5.2.1 Penalty Parameters

The reconfiguration penalty δ controls the trade-off between the reconfiguration effort and the energy consumption of the new network configuration. We evaluate its impact by varying its value $\delta \in \{0.0, 0.5, 1.0\}$, i. e. within the range suitable for energy-oriented network operation. With $\delta = 0.0$, we completely disregard the reconfiguration effort, whereas with $\delta = 1.0$ equaling the cost of operating one circuit which does not require additional active line cards, we eliminate the incentive to tear down such a circuit when it is no longer needed. Larger penalty values would thus contradict the objective of minimizing the energy consumption. $\delta = 0.5$ represents a moderate penalty value in-between these extremes.

Unlike circuit modifications, we want to avoid traffic blocking at any energetic cost. Consequently, the blocking penalties do not need to control a trade-off, but their value shall ensure that operating additional circuits is less expensive than blocking traffic. However, an excessive blocking cost is undesirable for the SA-based solution method: It may be worthwhile to accept candidate solutions blocking some traffic during the randomized search process in order to cover a sufficiently diverse subset of the solution space. Excessively high solution cost due to blocking would likely prevent this. We thus set $\beta_{\mathrm{L}} = \beta_{\mathrm{T}} = 40$, which is slightly higher than the worst-case cost of one circuit – which comprises operating one additional chassis with one line card and one port at each end. For the SA-based solution method, we need to additionally define the penalty for blocking demands. Since demand blocking is not addressed by the post-processing algorithm of section 4.4.3.4, we make this type of blocking unlikely by setting the respective penalty value to $\beta_{\mathrm{D}} = 2 \cdot \beta_{\mathrm{L}} = 80$.

5.2.2 Computation Time Limits for Exact Solution Method

The MILP formulation of section 4.4.2.2 together with the parameter values described above completely specifies an optimization problem instance, which we can hand to a solver software. It is generally possible to accelerate the computation of an optimal solution by tuning the use of different heuristics by the software. However, the configuration of the solver is out of scope of this monograph. Nevertheless, we need to restrict the time granted to compute one solution in order to complete the overall simulation within an acceptable timeframe.

The Diploma thesis project documented in [114] extensively studied the convergence behavior of the MILP solution using the software setup according to section 5.4. It observed that the gap between primal and dual bound first shrinks rapidly[2], whereas further improvements require a significantly longer computation time. For the Abilene network, most instances of the restricted problem according to section 4.4.2.2.4 were optimally solved within five minutes. Due to effort constraints, we apply this time limit for all topologies. In case no solution to the primal problem is found within the time limit, we iteratively increase this limit in an exponential manner in order to obtain a solution which allows continuing the simulation of subsequent reconfigurations.

If the restricted problem is unsolvable, i. e. traffic blocking is not avoidable, we revert to the complete problem of section 4.4.2.2, for which we apply an initial time limit of one hour in ac-

[2]This is in line with the observation for a similar problem in [49] that a relatively early termination of the branch-and-bound algorithm already yields high-quality solutions.

cordance with the recommendations in [114]. We likewise increase the time limit exponentially if no solution to the primal problem is found. We discuss the resulting gaps between primal solution and dual bound and document extended computation times in section 5.5.1.

5.2.3 Control of Heuristic Optimization Method

As detailed in section 4.4.3.3, our SA based solution method is controlled by a number of parameters steering the randomized search and its termination. The combination of these parameters influences both the quality of the returned solution and the number of considered candidate solutions and thus the computation time. While in principle longer computations tend to yield better solutions, unfavorable combinations of the control parameters may result in low solution quality despite long optimization runs.

The Bachelor thesis project documented in [42] systematically investigated the performance of a large set of parameter combinations with regard to solution quality and iterations. Taking the evaluation approach discussed in section 5.1, it simulated consecutive reconfiguration in realistic scenarios. It observed that we can find good solutions when setting low initial temperatures that result in a significantly lower fraction of initially accepted solutions than recommended in literature [36] and cited in section 2.5.2. Arguably, this is due to the limited variation of the traffic in consecutive reconfiguration intervals, which favors solutions close to the initial solution derived from the previous configuration. Table 5.3 lists the parameter set recommended for the Géant network in [42], which we also apply to the smaller topologies. Among the investigated parameter sets, it yielded almost minimal cost while terminating after relatively few iterations. An analog study for the larger Germany50 topology provided the parameters for this network listed in Table 5.3.

Table 5.3: Control parameters for SA based method

parameter		network topology: Abilene, Nobel-Germany, Géant	network topology: Germany50
initial temperature	ϑ_0	2	2
cooling factor	γ	0.95	0.95
max. temperature length	$\nu_{\max}$	1000	2000
max. accepted solutions per temp.	$\alpha_{\max}$	50	500
max. perturb. w/o improvement	$\eta_{\max}$	2000	8000
rel. target range of accepted cost	ρ	10^{-3}	10^{-3}

In addition to the parameters of the cooling schedule, we introduce one further parameter governing the probabilistic selection of the type of perturbation to perform: adding or removing one virtual link. Since preliminary studies favored an equal probability for both types, we set the probability of removing a virtual link to $p_{\mathrm{LR}} = 0.5$.

5.3 Reference Resource Scaling Scheme

Evaluating our reconfiguration methods against a static network operation regime according to current paradigms would yield energy savings strongly dependent on the network dimensioning. In addition, such an evaluation would confound energy savings enabled by the assumed power scaling features of network equipment with the additional benefit of computing optimized network configurations. We therefore define a simple resource scaling scheme, to which we compare the energy consumption and the operational impact of our reconfiguration methods.

This reference scheme assumes the circuit adaptation times and traffic forecasts according to section 4.1.3 and follows the periodic reconfiguration principle of section 4.1.4. Inspired by the current static network operation principle, it assumes the network to be configured for an expected peak load. Operating resources dynamically as provided in section 4.1.2, it periodically adapts the active resources within this configuration to the load.

The adaptation comprises scaling the packet-processing capacity of network nodes to the actual transit traffic, and deactivating unneeded circuits along with superfluous line cards and chassis. The configuration is static in the sense that traffic routes in the virtual topology are fixed and resources are assigned to specific circuits, along with which they may be activated and deactivated. We thus do allow concentrating traffic on one of several circuits realizing one virtual link in order to free circuits for deactivation. It should be noted that the deactivation of all circuits on a virtual link is only possible if all traffic demands routed over this link are zero. This scaling scheme corresponds to FUFL (fixed upper, fixed lower) in [21].

5.4 Simulation Setup

The evaluation was performed by means of a Java software based on the event-driven simulation library IKR SimLib [121, 122]. The representation of network topologies and path computation are supported by the JGraphT library [123]. For each reconfiguration interval, our Java software creates a problem instance object from the previous configuration, the demand matrix parsed from the traffic trace, and static properties of the simulation run such as network topology and cost parameters. The optimization-heuristic solution method as well as the baseline resource scaling scheme according to section 5.3 are implemented in the simulation software. They compute a new network configuration directly based on the problem instance object.

For the exact solution method, we make use of the MILP solver software SCIP[3] [28]. Our simulation software calls the solver for each reconfiguration interval after expressing the problem instance as a parameterized linear program. Once the SCIP process terminates, the simulation software translates the solution of the linear program into a network configuration, complementing it by the realization of optical circuits obtained by the heuristics detailed in section 4.4.1.

As argued in section 5.2.2, we use the default configuration of SCIP regarding the use of cut and branch heuristics. We set a time limit as detailed in section 5.2.2 and a memory limit of

[3] SCIP version 3.0.1 using the LP solver SoPlex in version 1.7.1

25 GB. The latter proved irrelevant in practice since the SCIP processes occupied significantly less memory.

5.5 Results

This section presents and analyzes results for the different metrics by which we evaluate the network settings obtained by our reconfiguration methods in comparison to the resource scaling scheme. We address reconfiguration effort in section 5.5.2, energy consumption in section 5.5.3, and traffic blocking in section 5.5.4. For an indication of the quality of the configurations relative to an optimal solution of our reconfiguration problem and to support the interpretation of some of the results, we start by reporting on the convergence of the exact solution method in section 5.5.1.

For the sake of concise presentation, we only plot results for selected parameter combinations and for the Abilene and Géant networks in this section. Unless explicitly noted, we observe similar effects for the Nobel-Germany and Germany50 topologies. For completeness, annex B gives the results for these networks. There and in the following, we denote the resource scaling scheme according to section 5.3 by *RS* , and we refer to the solution methods for our reconfiguration problem by the underlying optimization method: *MILP* for the linear programming-based complete optimization of the upper layer detailed in section 4.4.2, and *SA* for the heuristic optimization of the virtual topology according to section 4.4.3.

5.5.1 Convergence of Exact Solution Method

We discuss two metrics for the convergence of the linear programming-based solution method: the gap between the primal solution cost and the dual bound at the expiry of the computation time, and extensions of this time required to find any primal solution. For the largest of the considered topologies, Germany50, solutions proved to converge prohibitively slowly. We therefore do not apply the exact solution method to this topology.

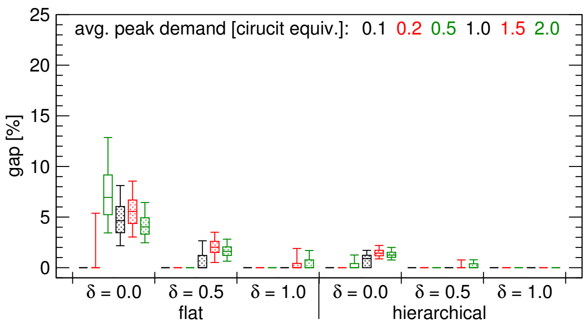
For Abilene, all optimization runs found a primal solution within the 5 minutes granted for the restricted problem. For Géant, the same applies for most of the 1344 optimization runs per parameter combination. Table 5.4 lists the exceptions. For the hierarchical power model, one extension to 10 min proved always sufficient, whereas one further extension step was required for the flat power model. It is worth noting that we observe most extensions for a reconfiguration penalty of $\delta = 0.0$ in sufficiently dimensioned networks: the limited influence of the previous setting in these constellations prevents a fast delimitation of the solution space. The basic optimization time of 1 h granted for the unrestricted problem proved sufficient to find a primal solution in all cases.

For the restricted problem, Figure 5.1 gives an overview of the distribution of the percentaged gaps between the cost of the primal solution and the dual bound at the end of the (basic or extended) optimization time. The subfigures represent one network topology and dimensioning, for which they give all combinations of power model and reconfiguration penalty. For each of these combinations, they further differentiate six traffic load scenarios with average peak

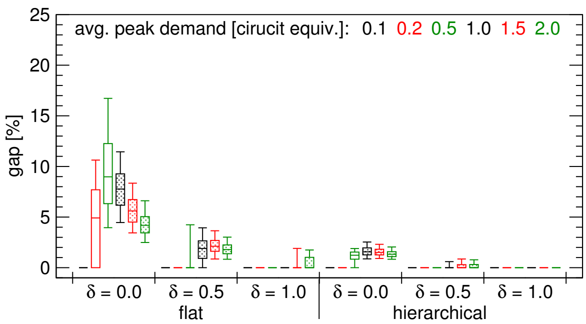
Table 5.4: Occurrence of extended optimization times for Géant

problem type			number of instances							extended
power model	σ_{dim}	δ	$\bar{d}_{\text{peak}} =$	0.1	0.2	0.5	1.0	1.5	2.0	duration
flat	1.0	0.0		2						80 min
hierarchical	0.8	0.0				1		1		10 min
		0.5				1	12	7	4	10 min
		1.0						1		10 min
	1.0	0.0		44	3	56	3	3	2	10 min
		0.5				2		1		10 min
	2.0	0.0				13	11	10	7	10 min
		0.5						1		10 min

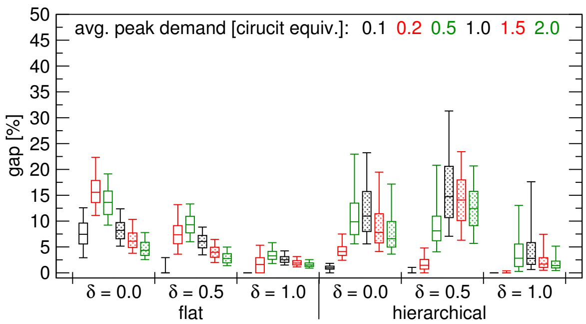
demands of $\bar{d}_{\text{peak}} \in \{0.1, 0.2, 0.5, 1.0, 1.5, 2.0\}$. For each of these scenarios, they finally visualize the distribution of gaps observed for the different reconfiguration intervals by means of box plots, where the margins of the box represent the 25 % and 75 % quantiles and the whiskers give the 5 % and 95 % quantiles, respectively. The middle lines in the boxes indicate the medians. In the cases where no boxes are visible, all percentiles except the 95 % quantile collapse to 0 % gap.



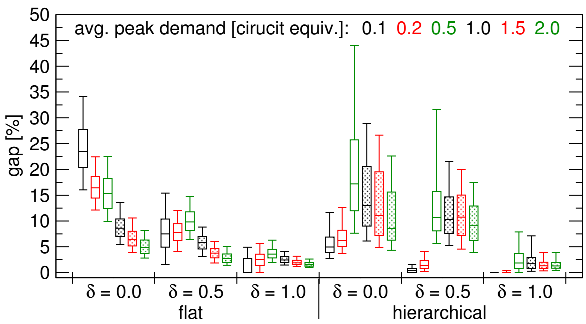
(a) Abilene, $\sigma_{\text{dim}} = 0.8$



(b) Abilene, $\sigma_{\text{dim}} = 1.0$



(c) Géant, $\sigma_{\text{dim}} = 0.8$

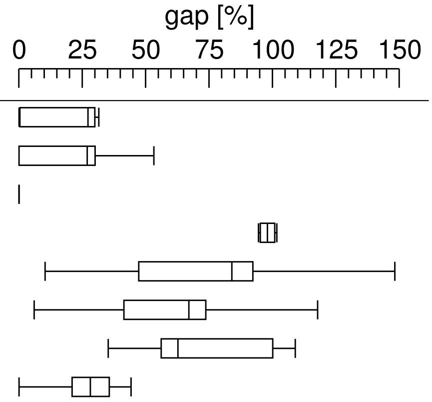


(d) Géant, $\sigma_{\text{dim}} = 1.0$

Figure 5.1: Gap between cost of primal solution and dual bound for the restricted problem per power model, reconfiguration penalty, and traffic load

Table 5.5: Cases of reversion to the unrestricted MILP model and resulting gaps

scenario					solutions of unrestricted problem
topology	σ_{dim}	power model	δ	$\bar{d}_{\text{peak}}$	number
Abilene	1.0	hierarchical	0.0	0.20	7
			0.5	0.20	8
			1.0	0.20	41
Géant	0.8	flat	0.0	0.10	4
				0.20	15
			0.5	0.20	313
			1.0	0.20	344
		hierarchical	1.0	0.10	17



We first observe a dependency of the gaps on the network size, obtaining gaps mostly below 10 % for the smaller Abilene network as compared to mostly below 30 % for Géant. This is expected since the problem size increases with the number of network nodes, cf. section 4.4.2.3. Next, the gaps increase with the dimensioning factor, since more generously dimensioned resources increase the solution space. This trend continues moderately for $\sigma_{\text{dim}} = 2.0$ (not plotted). Except for Géant with the hierarchical power model, we observe a trend of decreasing gaps with increasing reconfiguration penalty. This is explained by previous solutions restricting the promising subset of the solution space. For Géant with the hierarchical power model, the effect only becomes visible for $\delta = 1.0$. Finally, the gap depends on the traffic load, with the largest values mostly at moderate average peak demands of $\bar{d}_{\text{peak}} \in \{0.5, 1.0\}$: For small traffic load, few installed resources limit the degrees of freedom in reconfiguration, whereas for high load, many demands are best served by static direct circuits. Hence, the most demanding reconfiguration problems are found in between.

Considering the gaps in absolute terms, we find that for Abilene with the exception of the flat power model with $\delta = 0.0$, the cost of all solutions is less than 5 % above the optimum, with many problem instances optimally solved. For this topology, the results of the exact solution method thus allow a sufficiently accurate estimation of the maximum benefits achievable by solving the reconfiguration problem of section 4.3. For Géant, a 10 % gap margin is only kept for a reconfiguration penalty of $\delta = 1.0$ in the majority of cases. When discussing energy saving figures in section 5.5.3, we consider this margin the maximum tolerable for generalizing observations and apply particular caution interpreting results for larger gaps as incurred for lower reconfiguration penalties.

Table 5.5 lists those scenarios where we had to revert to the unrestricted problem in more than one reconfiguration interval[4]. For each scenario, the table gives the number of concerned problem instances (out of the 1344 reconfiguration intervals per simulation run) and box plots for the resulting gaps. With 75 % quantiles at around 30 % gap for Abilene and around 100 % gap for

[4]Single occurrences of the unrestricted problem were observed in four additional scenarios for Abilene and twelve further scenarios for Géant.

Géant, the gaps are significantly larger than those of the restricted problem despite the longer optimization time. This is in line with the observations in [114] motivating the use of the restricted problem. Except for two scenarios (Géant, $\sigma_{\text{dim}} = 0.8$, flat, $\delta \in \{0.5, 1.0\}$, $\bar{d}_{\text{peak}} = 0.20$), the low number of instances of the unrestricted problem prevents it from significantly influencing metrics averaged over the simulation run.

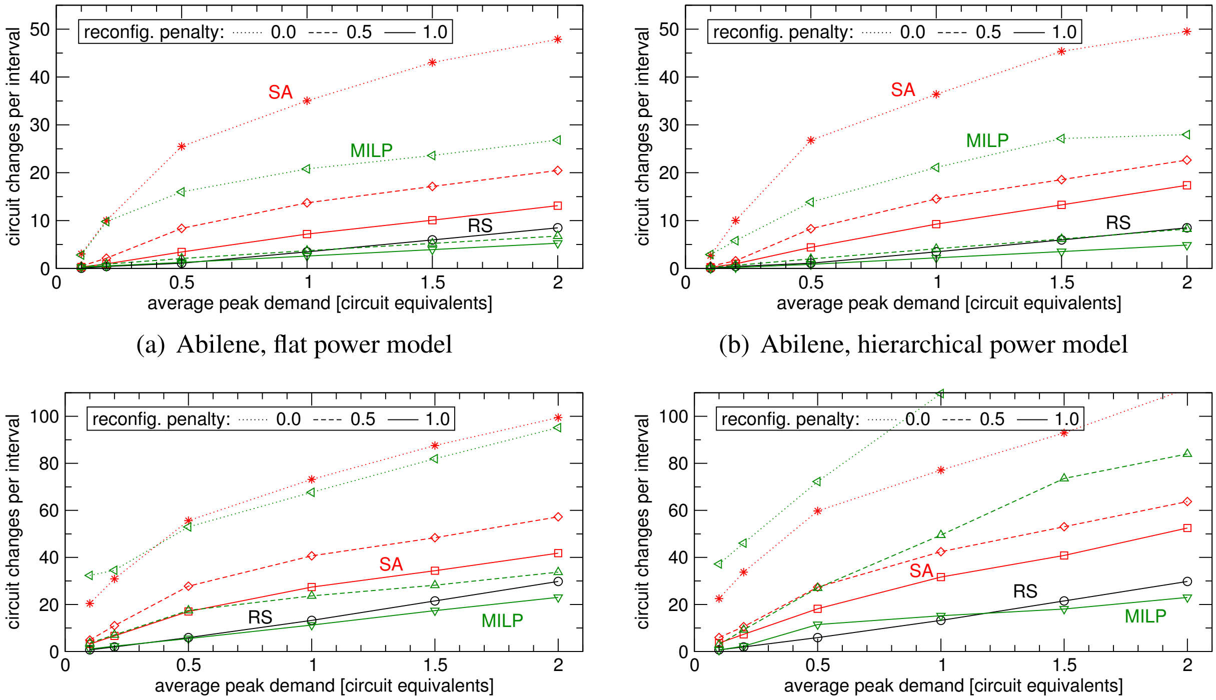
5.5.2 Reconfiguration Effort

In accordance with the arguments in section 4.2, we quantify the reconfiguration effort by the number of circuits newly established or torn down. Figure 5.2 plots the average number of such circuit modifications per reconfiguration step over the traffic load for different reconfiguration methods and parameterizations. In all cases, this number increases with the traffic load. This is plausible since we vary the load by scaling traffic profiles. Consequently, the amplitude of traffic variations increases, bringing about more circuit adaptations. The hierarchical power model enables increased energy savings by adapting several circuits in order to free and deactivate line cards. This results in a higher number of circuit modifications compared to the flat power model in some constellations.

All subfigures indicate a significant impact of the reconfiguration method and the reconfiguration penalty on the number of circuit modifications. For both MILP and SA, increasing the reconfiguration penalty δ reduces circuit changes. The largest effect is generally achieved by introducing a reconfiguration cost at all, i. e. by moving from $\delta = 0.0$ to $\delta = 0.5$. In most cases, MILP achieves fewer circuit modifications than SA for the same reconfiguration penalty. This trend indicates that a certain reconfiguration penalty value has different effects on the two solution methods. The observation does not hold for Géant with the hierarchical power model and $\delta \in \{0.0, 0.5\}$, where large gaps in Figure 5.1(d) indicate a potentially low quality of the MILP solution. For the maximum reasonable penalty value of $\delta = 1.0$, MILP is able to achieve approximately as few circuit modifications as the RS scheme, clearly outperforming SA. In this respect, MILP profits from optimizing traffic routing.

For reconfiguration penalties $\delta > 0.0$, the dependency of the reconfiguration effort on network dimensioning is relatively small. We do therefore not include plots for dimensioning factors other than $\sigma_{\text{dim}} = 1.0$. The dimensioning tends to influence the number of circuit modifications by two oppositional effects: While fewer installed resources may necessitate more circuit adaptations to accommodate the traffic, more resources bring about more degrees of freedom for reconfiguration. This likewise results in more changes. The dependency between dimensioning and reconfiguration effort therefore varies between scenarios with the dominance of either effect. Without reconfiguration cost ($\delta = 0.0$), the reconfiguration methods exploit additional resources, resulting in a significant increase of circuit changes with dimensioning.

Due to the different network sizes, the absolute numbers of circuit modifications do not allow a comparison between the topologies. Figure 5.3 therefore plots the average number of such modifications per reconfiguration step relative to the average number of active circuits over the traffic load. The influence of solution method and reconfiguration penalty is essentially the same as discussed for Figure 5.2. For the dependency of the fraction of changed circuits on the traffic load, we have to consider the resulting load on network links rather than the average peak demand, since a network configuration has to accommodate the accumulating



(a) Abilene, flat power model

(b) Abilene, hierarchical power model

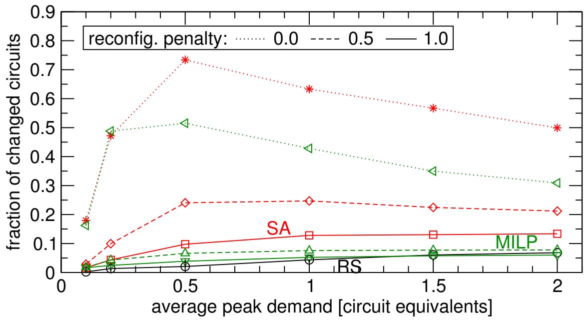
(c) Géant, flat power model

(d) Géant, hierarchical power model

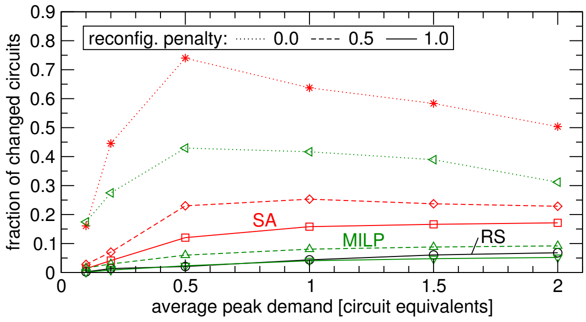
Figure 5.2: Average number of circuits established or torn down during each reconfiguration step for $\sigma_{\text{dim}} = 1.0$ per solution method and reconfiguration penalty over the traffic load

traffic. In the larger Géant network serving a higher number of node-to-node demands, the same average peak demand $\bar{d}_{\text{peak}}$ results in higher link loads compared to Abilene. Accordingly, the curves for Géant appear shifted to the left in comparison to those of Abilene in Figure 5.3. In principle, for low loads the fraction of changed circuits is restricted by a limited number of installed resources, which essentially ensure connectivity. This effect is visible for Abilene in Figure 5.3(a) and 5.3(b) for MILP and SA, whereas for Géant, the corresponding load point is outside of the plotted range. For high loads, an increasing share of circuits becomes permanent, accordingly decreasing the relative number of modified circuits. We observe this effect in all plots to varying extent. Between these two extremes, the fraction of changed circuits exhibits a more or less pronounced maximum.

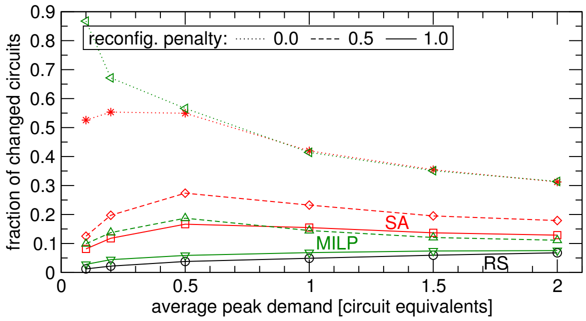
The ranges of the relative numbers of circuit modifications are similar in all topologies. For non-zero reconfiguration penalty, the average fraction of changed circuits does not exceed 30 % in any case. For SA and $\delta = 0.5$, it mostly ranges between 20 % and 25 %, with lower values for low loads. With $\delta = 1.0$, it reduces to between 10 % and 18 %. Provided that gaps are sufficiently small, MILP mostly achieves less than 10 % of modified circuits, which is on par with the RS scheme. The availability of only one day worth of averaged traffic data for the Nobel-Germany and Germany50 networks implies less variation in the concatenated traffic trace. This leads to a reduced share of modified circuits of 13 % to 20 % for SA with $\delta = 0.5$. For the large Germany50 network with a potentially more meshed virtual topology, RS incurs more relative circuit changes than for the other topologies, namely up to 12 %. With up to 90 %, the share of modified circuits obtained without reconfiguration cost (i. e. $\delta = 0.0$) by both reconfiguration methods appears prohibitively large.



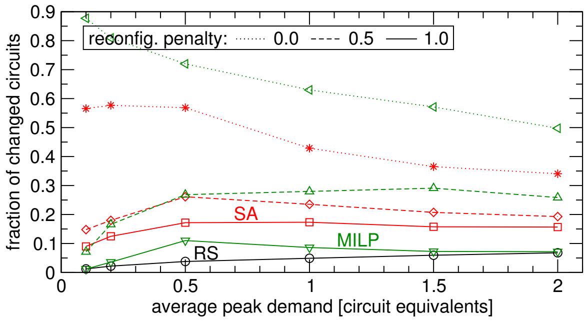
(a) Abilene, flat power model



(b) Abilene, hierarchical power model



(c) Géant, flat power model



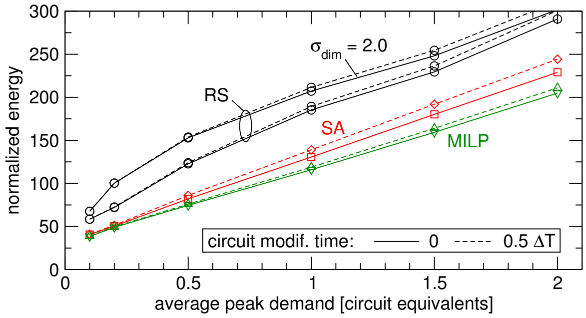
(d) Géant, hierarchical power model

Figure 5.3: Average fraction of circuits established or torn down during each reconfiguration step for $\sigma_{\text{dim}} = 1.0$ per solution method and reconfiguration penalty over the traffic load

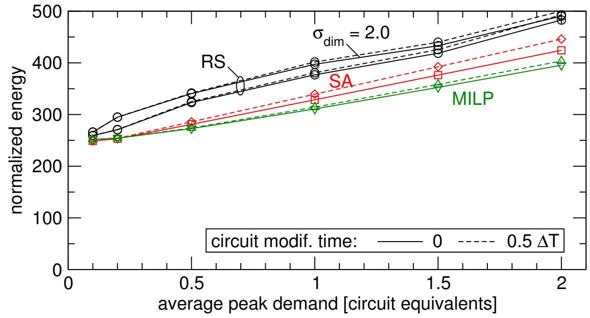
5.5.3 Energy Consumption

Energy consumption is the central metric in evaluating our reconfiguration methods aiming at energy savings. We report the average consumption of all load-adaptively operated resources according to the model of section 4.1.2 over the 14-day traffic trace, assuming component power values corresponding to the respective flat or hierarchical power model. The instantaneous power consumption is computed as specified in section 4.3.5.1 for the cost function term, i. e. we neglect the assignment of port pairs to line cards and account for the power consumption of the minimum number of hierarchical components required to operate all active circuits. For the RS scheme, this resembles the evaluation approach taken in [10].

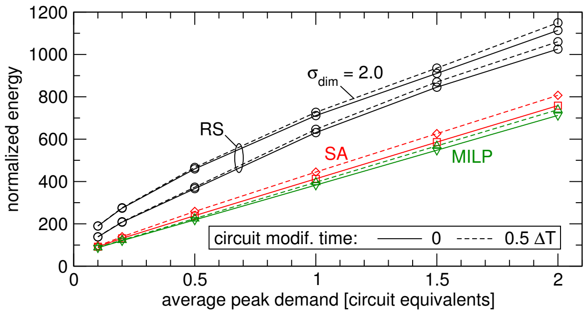
Figure 5.4 plots this average power consumption relative to the consumption of one active pair of ports. In all cases, the energy consumption grows approximately linearly with the traffic load as more resources are required to accommodate increasing traffic volumes. However, the slope and offset varies significantly with network topology and power model. The influence of the former is directly explained by the different numbers of nodes and overall traffic volumes. The latter influence mainly concerns the offset of the curves and has an evident explanation for small traffic load: While only few circuits are required, each node has to operate at least one line card chassis and one line card in order to maintain connectivity. This results in a high quasi-static energy consumption visible as the offset for the hierarchical power model. For the flat model, we only see the (relatively small) consumption of the few active circuits. A similar effect of lowly utilized chassis remains with increasing traffic load.



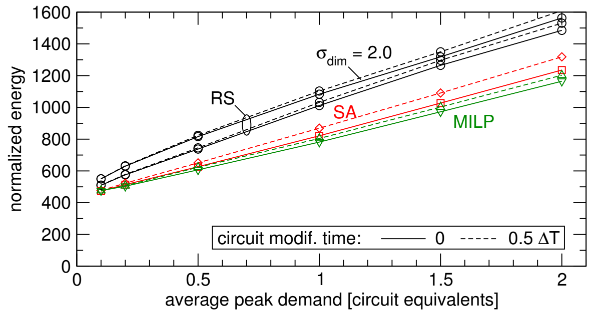
(a) Abilene, flat power model



(b) Abilene, hierarchical power model



(c) Géant, flat power model



(d) Géant, hierarchical power model

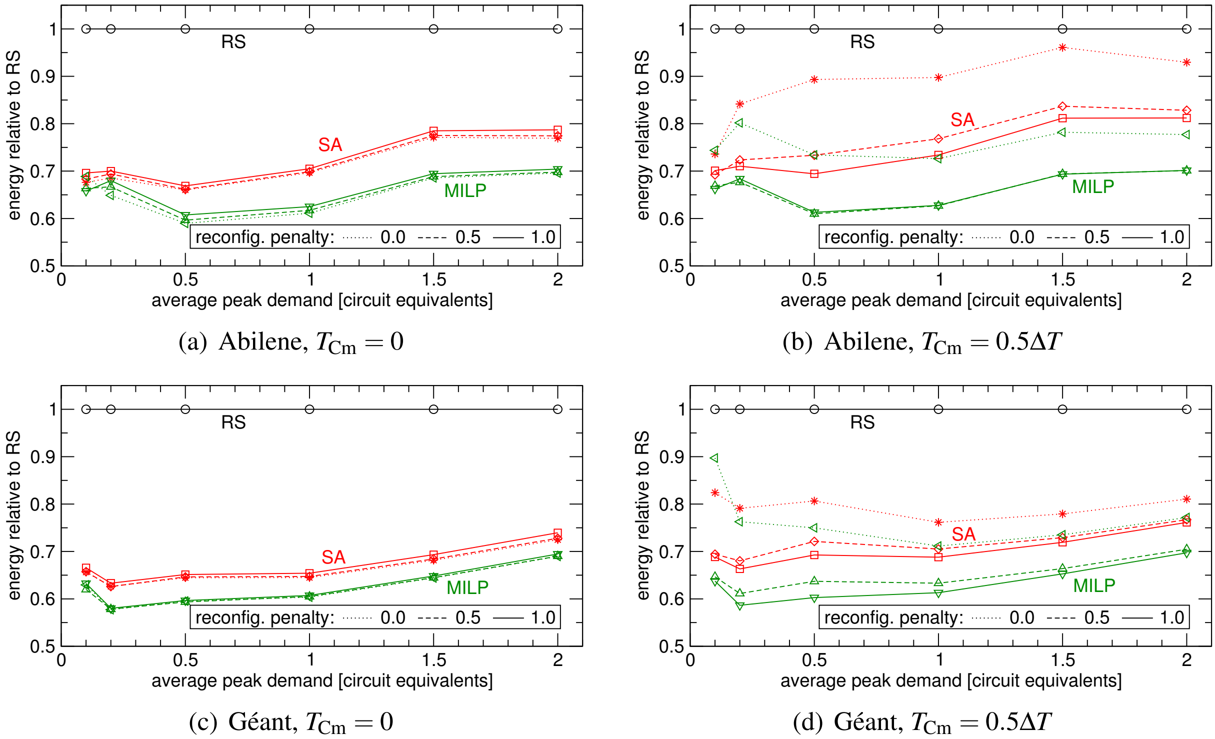
Figure 5.4: Normalized energy consumption of load-adaptive resources for $\sigma_{\text{dim}} = 1.0$ and $\delta = 1.0$ per solution method and circuit modification time over the traffic load

In each subfigure, which represents one network topology and power model, we compare the energy consumption achieved by the different reconfiguration methods. For each method, we see results for two extremal assumptions regarding the power consumption of resources during circuit setup and teardown: Either these resources consume no power at all, or they consume as much power as active resources for the maximum time they may be in a transient state. In accordance with our assumptions in section 4.1.4, this maximum circuit modification time is $T_{\text{Cm}} = 0.5\Delta T$. We denote the latter case by this circuit modification time value and refer to the former one by $T_{\text{Cm}} = 0$. In all cases, the energy consumption is moderately higher for $T_{\text{Cm}} = 0.5\Delta T$ than for $T_{\text{Cm}} = 0$. The difference is slightly more pronounced for SA, which is in line with the higher fraction of changed circuits observed for this solution method in section 5.5.2.

Since our reconfiguration methods adapt the resource configuration to the traffic load, the energy consumption for MILP and SA hardly depends on the network dimensioning[5]. We thus only plot the consumption for these methods for $\sigma_{\text{dim}} = 1.0$. For RS, a more generous dimensioning implies a basic network configuration optimized for a higher traffic load. Consequently, the configuration comprises more circuits, which carry small amounts of traffic preventing their deactivation. Hence, the energy consumption for RS increases with the dimensioning factor, as depicted exemplarily for $\sigma_{\text{dim}} = 2.0$ in Figure 5.4.

Energy savings achieved by our reconfiguration methods relative to the RS scheme thus depend on network dimensioning. Since our methods provide configurations fitting the respective

[5]In some cases, the consumption decreases minimally with increasing dimensioning factor, as more resources increase the possibilities for energy-efficient network reconfiguration.



(a) Abilene, $T_{\mathrm{Cm}} = 0$

(b) Abilene, $T_{\mathrm{Cm}} = 0.5\Delta T$

(c) Géant, $T_{\mathrm{Cm}} = 0$

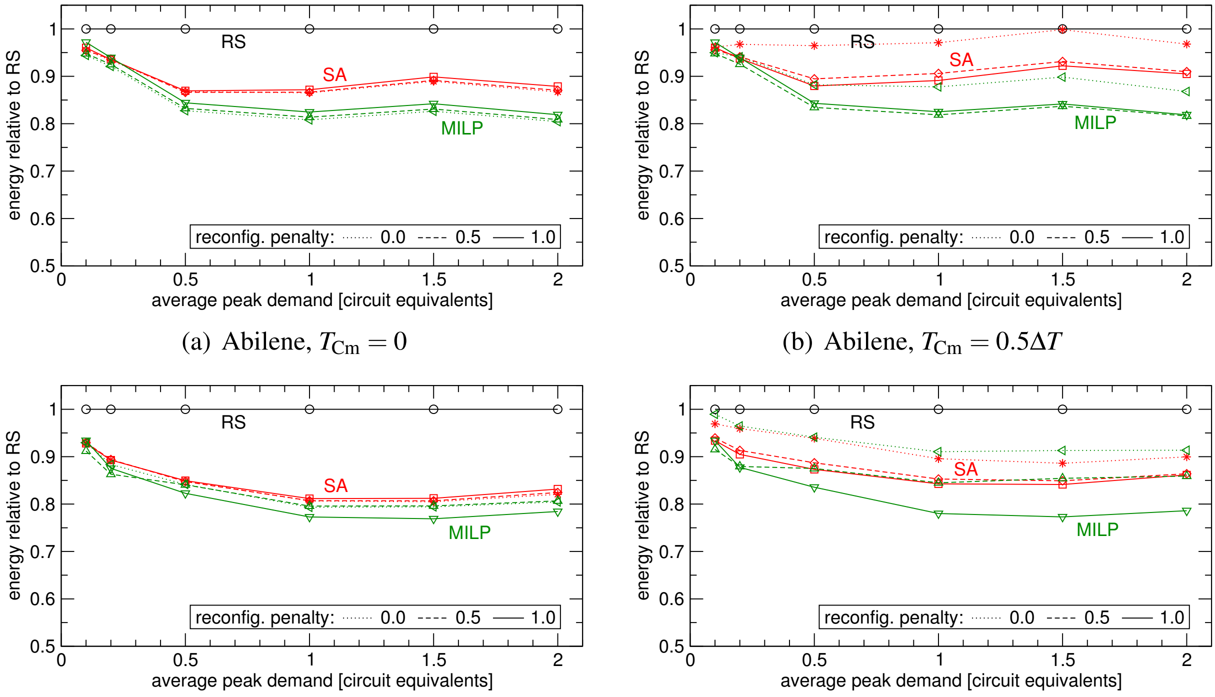
(d) Géant, $T_{\mathrm{Cm}} = 0.5\Delta T$

Figure 5.5: Load-dependent energy consumption relative to resource scaling for the flat power model and $\sigma_{\mathrm{dim}} = 1.0$ per solution method and reconfiguration penalty over the traffic load

traffic demand without explicit spare capacity, we base the reference RS scheme on a static configuration which accommodates the peak demand likewise without explicit spare capacity. We obtain such a configuration from network dimensioning with $\sigma_{\mathrm{dim}} = 1.0$.

Figure 5.5 plots the average energy consumption for the reconfiguration methods relative to that of RS for the flat power model over the traffic load. Unlike Figure 5.4, it separates the cases $T_{\mathrm{Cm}} = 0$ and $T_{\mathrm{Cm}} = 0.5\Delta T$ in different subfigures and shows the effect of varying the reconfiguration penalty. In case of instantaneous resource activation and deactivation ($T_{\mathrm{Cm}} = 0$), a smaller reconfiguration penalty results in slightly lower energy consumption for both solution methods and all topologies. This corresponds to expectations since reducing the influence of the previous setting allows the new configuration to better suit the new traffic situation, i. e. to carry the traffic at lower energetic cost. With increasing circuit modification time, we have to account for the transient consumption of circuits under modification. For the maximum circuit modification time of $T_{\mathrm{Cm}} = 0.5\Delta T$, the additional transient consumption is generally greater than the difference in the energy consumptions of the new configurations. In this case, the maximum reconfiguration penalty of $\delta = 1.0$ yields the lowest overall energy consumption in most constellations, with the exception of MILP for Abilene where a $\delta = 0.5$ proves most energy-efficient. The negative effect of many modified circuits is particularly evident without reconfiguration cost ($\delta = 0.0$).

Figure 5.6 gives the average energy consumption relative to RS for the hierarchical power model analogously to Figure 5.5 for the flat model. We generally observe the same dependencies of the energy consumption on the reconfiguration penalty – except for MILP in the Géant topol-



(a) Abilene, $T_{\mathrm{Cm}} = 0$ (b) Abilene, $T_{\mathrm{Cm}} = 0.5\Delta T$

(c) Géant, $T_{\mathrm{Cm}} = 0$ (d) Géant, $T_{\mathrm{Cm}} = 0.5\Delta T$

Figure 5.6: Load-dependent energy consumption relative to resource scaling for the hierarchical power model and $\sigma_{\mathrm{dim}} = 1.0$ per solution method and reconfiguration penalty over the traffic load

ogy: Here, the new configuration (given for $T_{\mathrm{Cm}} = 0$) turns out to be more energy-efficient for a reconfiguration penalty of $\delta = 1.0$ than for $\delta \in \{0.0, 0.5\}$. This is a consequence of the limited convergence of the MILP solutions indicated by relatively large gaps for $\delta < 1.0$ in Figure 5.1(d). Thus, the reduced solution quality in these cases is reflected in both the energy consumption of the new configuration and the number of changed circuits (as reported in section 5.5.2). The latter further increases the overall energy consumption for maximal circuit reconfiguration time ($T_{\mathrm{Cm}} = 0.5\Delta T$). It is however worth noting that these solutions do still perform similarly to the solutions of SA with the same reconfiguration penalty values in terms of energy consumption.

In the following, we discuss and quantify energy savings by our reconfiguration methods assuming a suitable setting of the reconfiguration penalty. We first consider the flat power model and Figure 5.5: MILP solutions achieve a reduction of the load-dependent energy consumption of about 30 % to 40 % over the RS scheme for both the Abilene and the Géant networks. The SA method, which exploits fewer degrees of freedom in the optimization, reaches between 18 % and 32 % of savings for Abilene. In the larger Géant topology, the optimization of the virtual topology allows a more fine-grained control of the network setting, yielding moderately higher savings ranging between 24 % and 34 %. In all cases, the savings decrease for high traffic load as a higher number of parallel circuits enables better power scaling with the reference scheme. For low loads the behavior of the savings varies, mainly due to differences in the energy consumption for RS. Except for the cases of maximum circuit reconfiguration times without reconfiguration penalty ($T_{\mathrm{Cm}} = 0.5\Delta T$, $\delta = 0.0$), the effect of the reconfiguration penalty on the energy savings is small compared to the influences of solution method and traffic load.

With about 20 % to 35 % for MILP and 15 % to 30 % for SA, energy savings for the Nobel-Germany and Germany50 networks with the flat power model (plotted in the annex in section B.3) are lower than for Abilene and Géant. This is primarily due to a lower energy consumption of the RS scheme: While for Abilene and Géant, singular peak values in the 14-day demand trace dominate the peak demand matrix we use for defining the static configuration, this matrix only contains the busy-hour values of the single-day profile for Nobel-Germany and Germany50. Relative to the average traffic load, we thus obtain a less overdimensioned static configuration in the latter case. This is beneficial for the energy consumption of RS, as observed earlier in this section. Moderately lower savings by SA in the Germany50 network compared to the same method for Nobel-Germany indicate a limited convergence of this solution method for the larger topology.

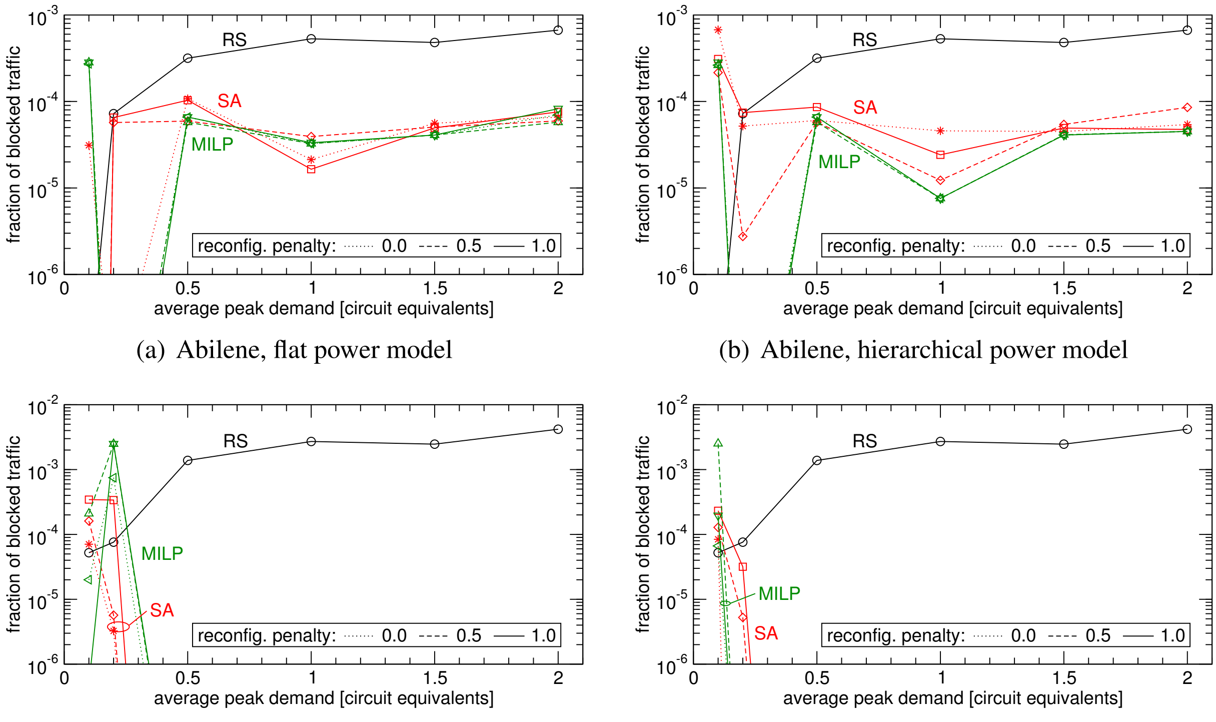
We now address the hierarchical power model, for which we observe significantly lower relative energy savings. This difference is primarily due to the quasi-static power consumption of line card chassis housing few active line cards (cf. Figure 5.4). Figure 5.6 shows savings of 4 % to 23 % for MILP and 4 % to 18 % for SA for Abilene and Géant. These savings tend to increase with the traffic load as the weight of lowly-utilized hierarchical components decreases. For Nobel-Germany, the saving figures are again slightly lower. Thanks to the higher overall traffic volumes in the larger Germany50 network, we observe higher savings than for the other topologies at low loads. The savings achieved by SA range between 10 % and 18 % in this topology.

5.5.4 Traffic Blocking

To evaluate traffic blocking, we differentiate the cases of sufficient resource dimensioning, i. e. $\sigma_{\mathrm{dim}} \in \{1.0, 2.0\}$, from those where resource shortages may prevent the static configuration determined during dimensioning from serving all traffic demands. In these latter cases of $\sigma_{\mathrm{dim}} = 0.8$, network reconfiguration may mitigate the loss of traffic due to insufficient capacity by flexibly dedicating resources to spatially varying traffic loads. While we do not systematically evaluate this benefit in this monograph centering on energy savings by network reconfiguration, the following results give some indication on the achievable effects.

Figure 5.7 gives the amount of traffic blocked on virtual links with insufficient capacity[6] during the 14-day period relative to the total offered traffic during this period. Like in the previous figures, we plot this metric for different reconfiguration methods and reconfiguration penalty values over the traffic load. For the RS scheme, the curves initially increase with the traffic load since in the case of small traffic load, the coarse granularity of installable resources leads to more spare capacity. For Abilene with $\bar{d}_{\mathrm{peak}} = 0.1$ circuit equivalents, this overdimensioning suffices to completely avoid blocking. For average peak demands of 0.5 circuit equivalents and more, the fraction of traffic blocked by RS ranges between 0.03 % and 0.07 % for Abilene and

[6] Since we sum up the traffic volume exceeding the capacity on all links, we disregard the possibility that the loss of traffic on some overloaded links mitigates overload on other links. Hence, this value may be higher than the actually lost traffic.



(a) Abilene, flat power model

(b) Abilene, hierarchical power model

(c) Géant, flat power model

(d) Géant, hierarchical power model

Figure 5.7: Average fraction of blocked traffic for $\sigma_{\text{dim}} = 0.8$ per power model for different solution methods and reconfiguration penalties over the traffic load

between 0.1 % and 0.4 % for Géant. In this load range, SA and MILP significantly reduce the amount of blocked traffic. For Abilene, they both achieve a reduction by two thirds to over one order of magnitude. For Géant, they manage to avoid blocking altogether.

For low traffic loads, our reconfiguration methods tend to perform worse than RS. For $\bar{d}_{\text{peak}} = 0.1$ circuit equivalents, disadvantageous configurations chosen during low traffic periods often render the transition to more suitable configurations impossible. This explains e. g. the high variation of the blocked traffic for SA with different reconfiguration penalties in Figure 5.7(a). At $\bar{d}_{\text{peak}} = 0.2$ circuit equivalents, traffic blocking by MILP and SA is on par with or below RS in most constellations. Géant with the flat power model depicted in Figure 5.7(c) constitutes an exception, while SA with small reconfiguration penalties proves that configurations with low blocking are possible in this scenario. Table 5.5 hints at an explanation: For $\delta \in \{0.5, 1.0\}$, we had to revert to the unrestricted MILP model in about one quarter of the reconfiguration intervals, resulting in gaps partly exceeding 100 %. This poor convergence of the solutions likely resulted in the high amount of blocked traffic. Lower blocking for $\delta = 0.0$ coincides with fewer intervals requiring the unrestricted problem.

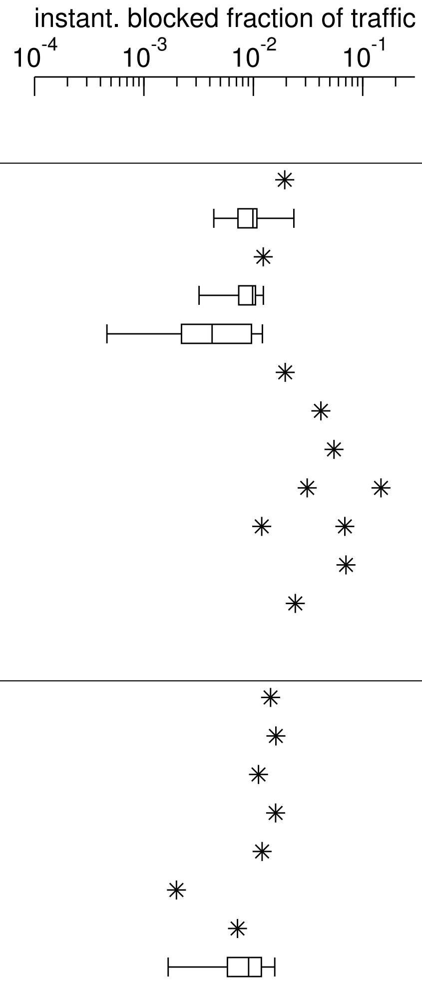
Due to tighter dimensioning, section B.4 in the annex shows the fraction of blocked traffic for RS to exceed 1 % for higher loads in the Nobel-Germany and Germany50 topologies. With one exception, blocking for MILP and SA is on par with RS for low loads in the Nobel-Germany network, whereas these solution methods reduce the fraction of blocked traffic by at least two orders of magnitude for higher loads. The latter also applies to Germany50 in all load scenarios.

If network resources are at least dimensioned for the peak demands, i. e. $\sigma_{\text{dim}} \geq 1.0$, the RS scheme guarantees network settings without traffic blocking. Blocking incurred by our reconfiguration methods due to disadvantageous previous configurations is then the price we have to pay in terms of service quality for the achieved energy savings. Fortunately, such blocking proved to occur rarely. Table 5.6 documents the occurrence of blocking events in the simulations for the Abilene and Géant networks along with the concerned traffic volume. Since these volumes are generally very small compared to the traffic offered over the simulated period, we indicate them relative to the average traffic offered within one reconfiguration interval (unlike in Figure 5.7). For up to two occurrences per 14-day period, we indicate the respective traffic fractions. In case of more concerned reconfiguration intervals, we give box plots specifying median, 25 % and 75 % quantiles, and 5 % and 95 % quantiles, respectively.

The table indicates that for Abilene with the hierarchical power model and an average peak demand of $\bar{d}_{\text{peak}} = 0.2$ circuit equivalents, MILP is particularly prone to incur traffic blocking.

Table 5.6: Occurrences of blocking for sufficient dimensioning

method	σ_{dim}	power model	δ	$\bar{d}_{\text{peak}}$	number (blocking events)
Abilene					
MILP	1.0	hierarchical	0.0	0.10	1
				0.20	7
			0.5	0.10	1
				0.20	8
			1.0	0.20	41
SA	1.0	flat	0.0	0.20	1
			0.5	0.20	1
		hierarchical	0.0	0.10	1
				0.20	2
			0.5	0.10	2
				0.20	1
			1.0	0.20	1
Géant					
MILP	1.0	hierarchical	0.0	0.20	1
				0.50	1
				1.00	1
			0.5	0.50	1
	2.0	hierarchical	0.0	1.00	1
				2.00	1
			0.5	1.00	1
SA	1.0	hierarchical	1.0	0.10	18



Further analysis of these simulations revealed that blocking occurs in the same seven temporally close reconfiguration intervals for all reconfiguration penalty values. For $\delta = 1.0$, the 41 intervals with blocking form several such groups. Gaps of 0 % in this constellation indicated in Table 5.5 prove that the blocking is unavoidable due to disadvantageous resource pre-occupation. The 18 instances of blocking in one simulation for Géant with SA are likewise temporally close. In contrast, the time instances of single blocking events in different simulations do not exhibit any correlation. They are thus randomly caused either by poor convergence of the optimization or by individual problematic previous configurations. The random nature of the occurrences of blocking is underlined by MILP finding blocking-free configurations under the same resource and traffic constraints for the flat power model. It is also worth noting that we observe no blocking in the Géant network for MILP with $\delta = 1.0$, which showed better convergence than for lower reconfiguration penalties.

One further important observation is that except for three badly converged MILP instances for Géant, all events of blocking occurred for a dimensioning factor of $\sigma_{\text{dim}} = 1.0$. Hence, providing more resources efficiently reduces the risk of traffic blocking. Otherwise, the reconfiguration methods bring about a certain risk of blocking, instantaneously concerning up to 2 % of the average offered traffic for MILP and reaching up to 10 % (or more in one case) for SA in the investigated scenarios. For Nobel-Germany, the frequency and amplitude of traffic blocking is lower for both solution methods. For Germany50, SA produced somewhat more blocking events, which however only concern minimal fractions of traffic in most cases. According to section B.4 in the annex, the maximum instantaneously blocked traffic fraction is 3.5 % there.

5.6 Discussion

5.6.1 Energy Consumption

The good convergence behavior of the exact solution method at least for certain reconfiguration penalty values allows a precise estimation of the minimum energy consumption achievable by applying our reconfiguration problem defined in section 4.3 in the considered scenarios. Generally valid quantifications of obtained energy savings are however impossible, since they strongly depend on the reference case. Relative to our baseline scheme scaling active resources according to the load in a configuration optimized for the peak demand, we obtain scenario-dependent reductions of up to 20 % by assuming that hierarchical components of state-of-the-art routers are switched on and off according to demand. These savings may double with more fine-grained power adaptation features or a leaner component hierarchy reducing the quasi-static power consumption of line card chassis.

A high reconfiguration penalty value of $\delta = 1.0$ tends to be beneficial for the energy consumption. By reducing the number of modified circuits, it limits the energy consumed by resources during activation and deactivation. In addition, such a value improves the solution quality of the exact method in case of larger networks, which proved to have a positive effect on the energy consumption of the resulting network configuration.

The more coarse-grained optimization of the network configuration by the heuristic solution method centering on the virtual topology comes at a cost in terms of energy consumption: its resulting configurations consume up to about 10 % more energy than those of the exact solution method, which additionally optimizes traffic routing.

5.6.2 Reconfiguration Effort

It is impossible to name a bound for the admissible reconfiguration effort since the dynamic operation of optical core network resources is a future paradigm unsupported by current equipment. Accordingly, no practical experience of possibly detrimental side effects exists. We therefore take the circuit activations and deactivations incurred by our reference resource scaling scheme for orientation. In three out of the four considered networks, it modifies up to 8 % of existing circuits on average depending on the load scenario. In the largest considered topology, this figure is up to 12 %.

The exact solution method completely optimizing the upper layer generally causes a similarly low reconfiguration effort for a penalty of $\delta = 1.0$. The effect of lower positive penalty values depends on the convergence of the solution method: In case of a high solution quality, the effort is only moderately higher, whereas poorly converged solutions tend to significantly increase the number of circuit modifications.

The heuristic optimization of the virtual topology brings about two to three times as many circuit changes as the exact method for $\delta = 1.0$ due to its coarser control of the network setting. Depending on the scenario, reducing the reconfiguration penalty for the heuristic method may increase the reconfiguration effort moderately or by up to a factor of 2.5. For both solution methods, a penalty of zero may lead to a significant increase of the reconfiguration effort and is therefore not advisable.

5.6.3 Traffic Blocking

The tolerable level of traffic blocking is likewise hard to specify. A wide-spread paradigm is that no traffic blocking shall occur in core networks, which are protected from traffic peaks by the limited capacity of aggregation networks. Still, blocking is not generally avoidable in case of extremal traffic load in particular situations.

In accordance with this paradigm, we do not consider the case of insufficient installed resources as a relevant scenario to evaluate the blocking behavior of our reconfiguration methods. In this case, however, we find that both of our solution methods can significantly reduce the amount of blocked traffic compared to a static configuration if a minimum quantity of installed resources allows them to be effective. In case of few available resources, in contrast, reconfiguration attempts may also aggravate traffic blocking.

Whether traffic may be blocked in practice depends on service level agreements and traffic classes. We can therefore not generally state whether the frequency and amplitude of the isolated blocking events we observed with resources dimensioned for the peak demand are tolerable. It is however instructive to note that providing resources dimensioned for twice the peak demand reduces these events in both frequency and amplitude. While the low numbers of such events prevents the generalization of this observation, it does hint that spare resources installed in core networks for resilience may help to reduce reconfiguration-related traffic blocking to acceptable levels if used to support reconfiguration during normal network operation.

6 Conclusion and Outlook

Environmentalist concerns, resulting regulatory measures, and cost pressure oblige network operators to limit the energy consumption of their transport networks in the face of increasing traffic volumes. Traditionally, the energy consumption of the access segment has been dominant due to a huge number of devices. However, new access technologies like PONs support increasing access rates and traffic volumes at constant energy consumption, moving the focus to core networks where increasing traffic results in an according increase of the energy consumption assuming current technology. One countermeasure consists in adapting the configuration of generally largely overdimensioned core network resources to the traffic load, thereby saving the energy otherwise consumed by active but unused resources. Previous research on energy-aware core network reconfiguration *either* disregarded constraints inherent to reconfiguration and optimized individual network configurations *or* incrementally modified the network configuration by means of simple heuristics. This thesis has developed and evaluated an optimization problem and according solution methods respecting realistic constraints of a reconfiguration situation.

Core networks are generally multi-layer networks featuring a circuit-switched optical WDM network as the lowest layer. Their upmost electrical layer is packet-switched. We assumed a widely-used architecture consisting of two such layers: IP/MPLS over WDM networks. Since optical circuits do not allow a fine-grained adaptation of their capacity and power consumption to the traffic load they carry, energy-aware reconfiguration of such networks requires adapting the virtual topology made up by these circuits. Significant diurnal variations of the traffic volume observed at public internet exchanges, with nightly periods with traffic as low as 25 % of the peak hour, hint at an important potential for energy savings by load-dependent core network operation.

In chapter 4, we developed our reconfiguration method. We started by deriving a model for the dynamic operation of core network node components and their corresponding power consumption. This model underlies the minimization of the network-wide power consumption and the evaluation of the energy consumption in chapter 5. Next, we addressed a technological constraint governing the reconfiguration time: The transient behavior of EDFAs requiring the gradual adaptation of the power of optical signals. This results in circuit modification times in the order of minutes. Together with a limited traffic forecast horizon – we assume a reasonable bound while disregarding the actual prediction mechanism – this circuit modification time implies a hitless transition between consecutive configurations in a single step. We devised an according one-step reconfiguration scheme and included pre-occupation constraints for optical resources in the optimization problem formulation.

In accordance with the motivation of this thesis, the main objective of our reconfiguration problem is minimizing the power consumption of the new network configuration. However, we identified a further central requirement on the reconfiguration method: limiting the impact on network operation. We quantify this impact by two metrics: First, the volume of traffic lost due to disadvantages previous network configurations. Since traffic loss is hardly tolerable in core networks due to QoS requirements, we try to avoid it at any energetic cost. Second, the number of modified circuits. While circuit reconfiguration is a prerequisite for saving energy, resources occupied by circuits under modification consume energy during their gradual set-up or teardown. The evaluation in chapter 5 showed that limiting circuit modifications is beneficial regarding overall energy consumption. In addition, limited reconfiguration is less of a breach with the traditional paradigm of static core network operation.

We opted to include all three aspects in one objective function and accordingly formulated a single optimization problem. We preferred this approach to first minimizing traffic loss in order to facilitate online application, which may require to abort the computation upon reaching a time limit and to use a preliminary sub-optimal solution. This applies in particular to the case of heuristic optimization. Since the non-trivial problem of allocating ports from line cards cannot be solved optimally without knowledge of all temporally subsequent network configurations, we excluded this aspect from our problem formulation and always assumed an optimal allocation.

Based on the optimization methodology introduced in chapter 2, we then defined two solution methods for our reconfiguration problem. We first split off the circuit routing problem, for which we proposed a set of heuristics. For the remaining problem, we established a multi-commodity flow MILP formulation, which provides solutions of defined quality by means of mathematical optimization. Alternatively, we devised a heuristic optimization procedure based on simulated annealing. For reduced complexity, the latter only optimizes the virtual topology while routing traffic along shortest paths. Afterwards, it tries to resolve two issues incurred by shortest-path routing by means of a simple heuristic.

In chapter 5, we systematically evaluated the network configurations obtained from our reconfiguration methods by metrics reflected in the objective function: the energy consumption and the operational impact in terms of circuit modifications and traffic blocking. We aimed at doing so in a scenario as realistic as possible, in particular with respect to traffic dynamics. Due to the lack of suitable dynamic models for aggregated core network traffic, we resorted to using traffic traces extracted from measurements. While this restricted our evaluation to a set of REN-based reference networks for which such traces – collected in the years 2004 and 2005 – are publicly available, we were able to maintain a reasonable variation of evaluation scenarios by scaling traffic traces and varying resource dimensioning to emulate different network loads.

Using static network operation according to the current paradigm as a reference case to quantify energy savings and the impact on network operation has several drawbacks. First, the baseline energy consumption would strongly depend on assumptions made during network dimensioning. Second, saving figures would show the combined benefit of the technological progress we assume in terms of load-adaptively operable network equipment and of the effort spent to optimize network configurations. Finally, a static reference case does not provide a useful baseline for judging the reconfiguration effort. For these reasons, we defined a different reference scheme: resource scaling, which assumes a static network configuration regarding traffic routes while exploiting our assumptions on load-dependent resource operation by deactivating idle circuits and scaling packet processing capacities.

Compared to this resource scaling scheme, our MILP-based solution method achieved significant reductions of the load-dependent energy consumption, mostly ranging between 15 % and 20 % in the considered scenarios. These figures doubled when disregarding hierarchical node components and assuming power consumption proportional to the number of active circuits. The heuristic method, which does not optimize traffic routing, incurs a penalty of about 10 %, still yielding significant savings especially in the second case.

The influence of network reconfiguration on network operation and QoS proved unproblematic by our definitions. Given a suitable reconfiguration penalty, the MILP-based method did generally not incur more circuit modifications than the reference scheme. The heuristic optimization method brought about two to three times this number due to its coarser control of the circuit configuration. Whether this is still acceptable depends on the actual QoS impact of setting up and tearing down optical circuits, which has to be evaluated as technology matures. Traffic loss occurred sporadically in the evaluation. In the majority of cases, it was resolved by a more generous dimensioning of installed resources. If resources provided for resilience are available for reconfiguration during normal operation, traffic loss should therefore not be an issue.

In conclusion, we found that network reconfiguration by our methods can save significant amounts of energy in core networks without problematic impact on network operation and QoS. The savings would further increase if network nodes exhibited improved power proportionality, i. e. a stronger dependency of their power consumption on active circuits and processed traffic volumes, while we assumed active line cards and chassis to dominate. In contrast, power proportionality of optical circuits, which may be approached with flexible line rates, would improve the energy efficiency of resource scaling and accordingly reduce the benefit of optimization-based reconfiguration.

On an algorithmic level, future research could target heuristics for the line card port allocation problem and integrate them with our reconfiguration methods. This would enable an even more realistic evaluation of the energy saving potential. In addition, research could aim at improved solution methods providing good solutions in shorter time for our reconfiguration problem. Recent developments towards more flexibility in the optical layer, such as flexible line rate and flexible grid, offer additional degrees of freedom for network reconfiguration. According reconfiguration methods should also be subject to future research. On a technological level, research towards an increased power proportionality of network nodes would enable significantly higher energy savings by load-dependent network operation. Finally, traffic prediction required for network resource operation closely following demand is an open issue.

A Fundamental Mathematical Notations

This annex defines a number of fundamental mathematical notations used throughout this monograph. Its first part addresses conventions on the use of different types of parenthesis, whereas the second one makes use of these conventions to define sets of numbers with several quantifiers.

A.1 Literals for Sets, Intervals, Tuples

In order to improve the legibility of mathematical expressions, we use different types of parenthesis according to widely applied conventions and some additional definitions. We detail these notations in the following.

We describe sets by curled brackets, denoting a set of countably many elements e_i by $\{e_1, e_2, e_3, \dots\}$. Alternatively, we use a vertical bar to separate elements in the set from conditions applying to them, e. g. $\{e_i | i = 1, 2, 3, \dots\}$. This also allows specifying sets of innumerably many elements.

Intervals comprising contiguous subsets of real numbers are special cases of sets of innumerably many elements. We use square brackets to designate closed intervals, which comprise their border values, and normal parenthesis for open intervals. We accordingly denote the respective intervals between 0.5 and 2.0 by

$$[0.5, 2.0] \equiv \{x \in \mathbb{R} | x \geq 0.5 \wedge x \leq 2.0\} \tag{A.1}$$

$$(0.5, 2.0) \equiv \{x \in \mathbb{R} | x > 0.5 \wedge x < 2.0\} \tag{A.2}$$

Herein, $\mathbb{R}$ is the set of all real numbers detailed below, and $\wedge$ denotes the conjunction of the two conditions.

Finally, we use different types of parentheses to denote ordered lists of elements. If the elements are homogeneous (i. e. belonging to the same basic set), we refer to the lists as *vectors* (or homogeneous tuples) and enclose the listing in square brackets, e. g. $[e_1, e_1, e_3, e_2, e_4]$. In the case of exactly two elements, we alternatively use the *pair* notation with normal parenthesis, particularly when referring to pairs of nodes identifying a network link or demand: (v_1, v_2). To distinguish heterogeneous tuples (comprising elements of different kinds), we enclose them in angle brackets, e. g. $\langle i_1, i_2, v_1, v_2 \rangle$ where i_i may be integer numbers and v_i may be nodes. One classic example of a heterogeneous tuple is a graph $G = \langle V, E \rangle$ consisting of a set of vertices V and a set of edges E.

A.2 Sets of Numbers

Regarding sets of numbers, we use the following sets and qualifiers. We refer to

- the set of all real numbers by $\mathbb{R}$,
- the set of all integer numbers by $\mathbb{Z}$.

To both symbols, we apply the plus sign in superscript to designate the respective subset of positive values, i. e.

$$\mathbb{R}^+ \equiv \{x \in \mathbb{R} | x > 0\} \equiv (0, +\infty) \tag{A.3}$$

$$\mathbb{Z}^+ \equiv \{x \in \mathbb{Z} | x > 0\} \tag{A.4}$$

Finally, we add the digit zero in subscript to add zero to the respective positive sets, yielding the sets of non-negative real or integer numbers, respectively:

$$\mathbb{R}_0^+ \equiv \{x \in \mathbb{R} | x \geq 0\} \equiv [0, +\infty) \tag{A.5}$$

$$\mathbb{Z}_0^+ \equiv \{x \in \mathbb{Z} | x \geq 0\} \tag{A.6}$$

B Further Results

Analogously to section 5.5 for Abilene and Géant, this annex reports evaluation results for the Nobel-Germany and the Germany50 topologies. For discussions of these results, the reader is referred to section 5.5, which either discusses effects observed for all topologies or explicitly points out differences in plots in this annex.

B.1 Convergence of Exact Solution Method

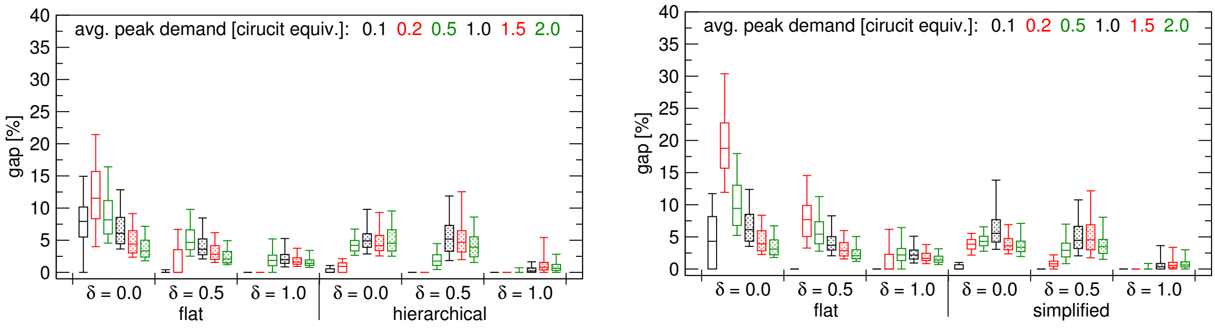
Since the exact solution method did not converge within reasonable time for the largest reference network in our evaluation, Germany50, we only report on the convergence behavior of MILP for Nobel-Germany. Table B.1 lists the settings where the restricted MILP variant required optimization time extension. While the concerned problem instances are few, it is remarkable that a primal solution has only been found after incrementing the time limit four times following the exponential scheme.

Table B.1: Occurrence of extended optimization times for Nobel-Germany

problem type			number of instances for $\bar{d}_{\text{peak}}$						extended
power model	σ_{dim}	δ	0.1	0.2	0.5	1.0	1.5	2.0	duration
flat	1.0	0.0		1					21.33 h
hierarchical	1.0	0.0		1					21.33 h

Likewise for the restricted problem, Figure B.1 gives an overview of the distribution of the percentaged gaps between the cost of the primal solution and the dual bound at the end of the basic or extended optimization time. The subfigures represent two different dimensionings, for which they give all combinations of power model and reconfiguration penalty. For each of these combinations, they further differentiate six traffic load scenarios, for which they finally visualize the distribution of gaps observed for the different reconfiguration intervals by means of box plots, where the margins of the box represent the 25 % and 75 % quantiles and the whiskers give the 5 % and 95 % quantiles, respectively.

Table B.2 lists those scenarios where we had to revert to the unrestricted problem in more than one reconfiguration interval. For each scenario, the table gives the number of concerned problem instances (out of the 192 reconfiguration intervals per simulation run for Nobel-Germany)



(a) $\sigma_{\text{dim}} = 0.8$ (b) $\sigma_{\text{dim}} = 1.0$

Figure B.1: Gap between cost of primal solution and dual bound for the restricted problem for Nobel-Germany per power model, reconfiguration cost, and traffic load

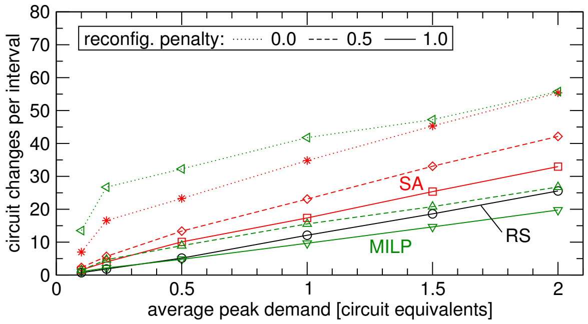
and box plots for the resulting gaps. In one further scenario, we observed a single instance of the unrestricted problem.

Table B.2: Cases of reversion to the unrestricted MILP model and resulting gaps for Nobel-Germany

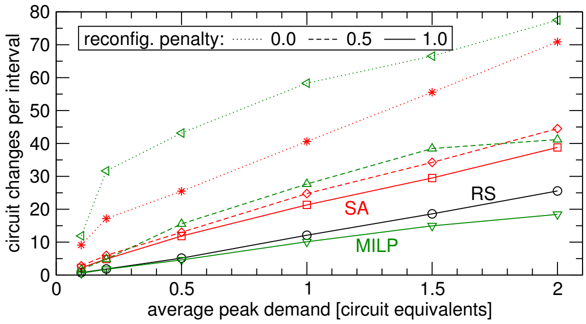
problem type				solutions of unrestricted problem	
σ_{dim}	power model	δ	$\bar{d}_{\text{peak}}$	number	gap [%] 0 25 50 75 100 125 150
0.8	hierarchical	0.5	0.10	5	
0.8	flat	1.0	0.20	28	
0.8	hierarchical	0.0	0.20	25	
0.8	hierarchical	0.5	0.20	2	

B.2 Reconfiguration Effort

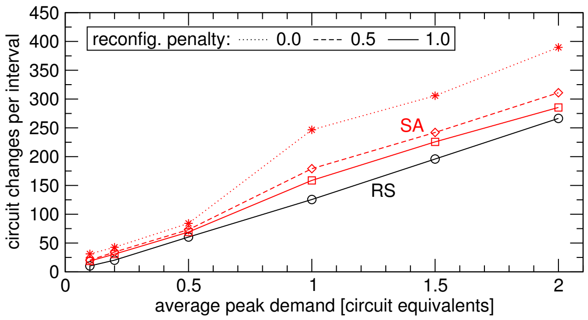
All metrics discussed in the following are available for Nobel-Germany for both solution methods and for Germany50 for SA only, since MILP did not converge within reasonable time limits. Figure B.2 plots the average number of newly established or torn-down circuits per reconfiguration step over the traffic load for different reconfiguration methods and parameterizations. To allow comparisons across topologies of different sizes, Figure B.3 plots the same numbers relative to the average number of active circuits.



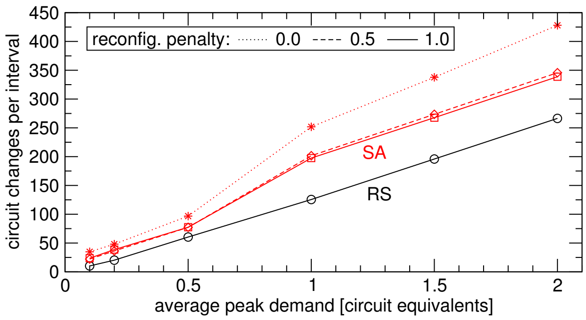
(a) Nobel-Germany, flat power model



(b) Nobel-Germany, hierarchical power model

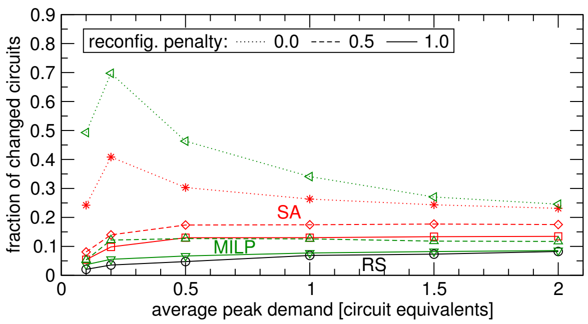


(c) Germany50, flat power model

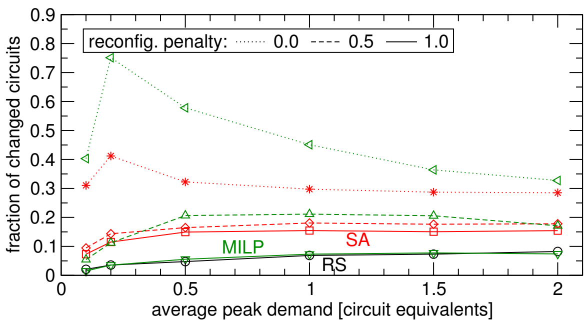


(d) Germany50, hierarchical power model

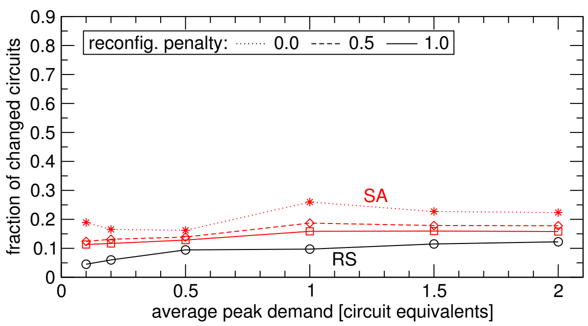
Figure B.2: Average number of circuits established or torn down during each reconfiguration step per solution method and reconfiguration penalty over the traffic load



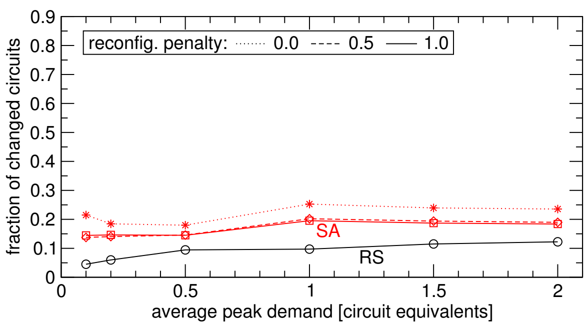
(a) Nobel-Germany, flat power model



(b) Nobel-Germany, hierarchical power model



(c) Germany50, flat power model



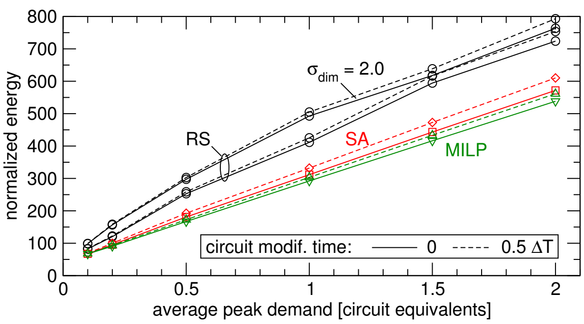
(d) Germany50, hierarchical power model

Figure B.3: Average fraction of circuits established or torn down during each reconfiguration step per solution method and reconfiguration penalty over the traffic load

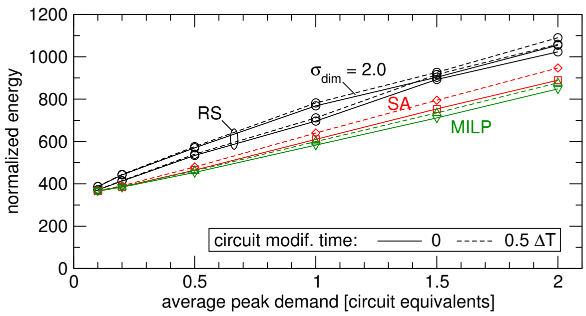
B.3 Energy Consumption

Figure B.4 plots this average power consumption relative to the consumption of one active pair of ports. In each subfigure, which represents one network topology and power model, we compare the energy consumption achieved by the different reconfiguration methods. For each method, we see results for two extremal assumptions regarding the power consumption of resources during circuit setup and teardown: Either these resources consume no power at all, or they consume as much power as active resources for the maximum time they may be in a transient state. In accordance with our assumptions in section 4.1.4, this maximum circuit modification time is $T_{\mathrm{Cm}} = 0.5\Delta T$. We denote the latter case by this circuit modification time value and refer to the former one by $T_{\mathrm{Cm}} = 0$.

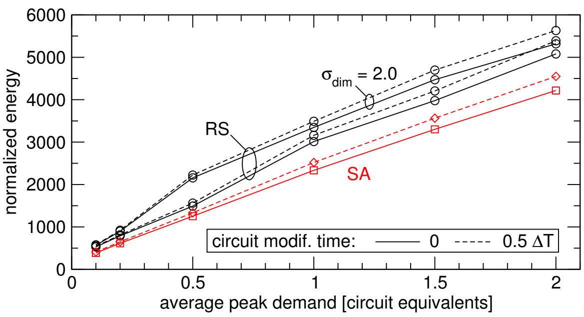
Figure B.5 and Figure B.6 plot the average energy consumption for the reconfiguration methods relative to that of RS for the flat and the hierarchical power model, respectively, over the traffic load. Unlike Figure B.4, these figures separate the cases $T_{\mathrm{Cm}} = 0$ and $T_{\mathrm{Cm}} = 0.5\Delta T$ in different subfigures and show the effect of varying the reconfiguration penalty.



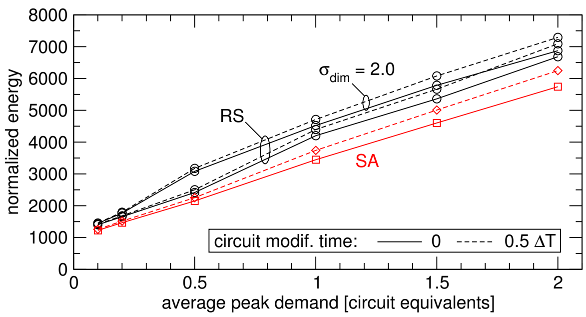
(a) Nobel-Germany, flat power model



(b) Nobel-Germany, hierarchical power model

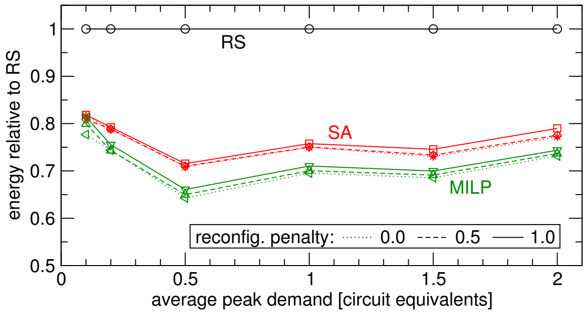


(c) Germany50, flat power model

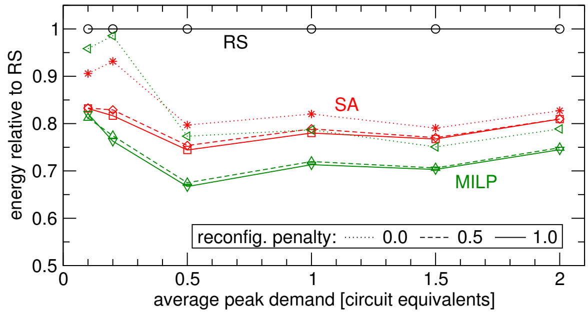


(d) Germany50, hierarchical power model

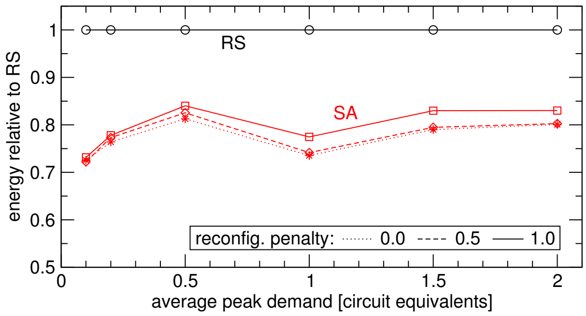
Figure B.4: Normalized energy consumption of load-adaptive resources for $\sigma_{\mathrm{dim}} = 1.0$ and $\delta = 1.0$ per solution method and circuit modification time over the traffic load



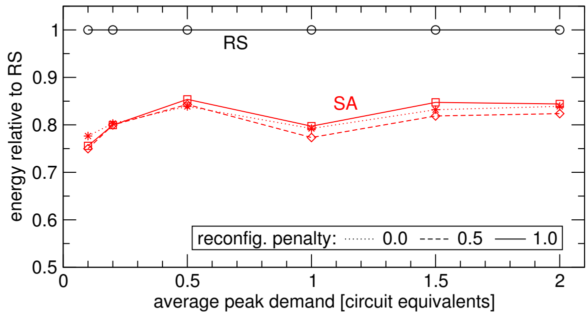
(a) Nobel-Germany, $T_{\mathrm{Cm}} = 0$



(b) Nobel-Germany, $T_{\mathrm{Cm}} = 0.5\Delta T$

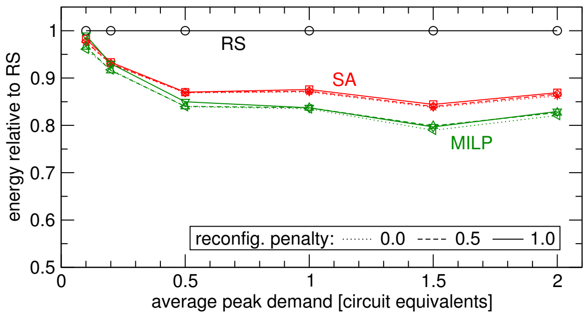


(c) Germany50, $T_{\mathrm{Cm}} = 0$

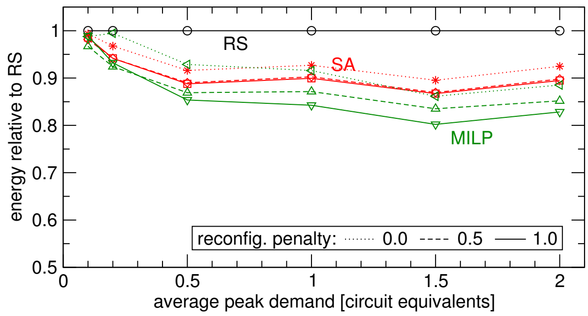


(d) Germany50, $T_{\mathrm{Cm}} = 0.5\Delta T$

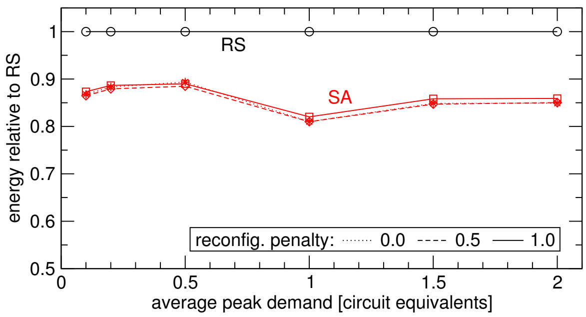
Figure B.5: Load-dependent energy consumption relative to resource scaling for the flat power model and $\sigma_{\mathrm{dim}} = 1.0$ per solution method and reconfiguration penalty over the traffic load



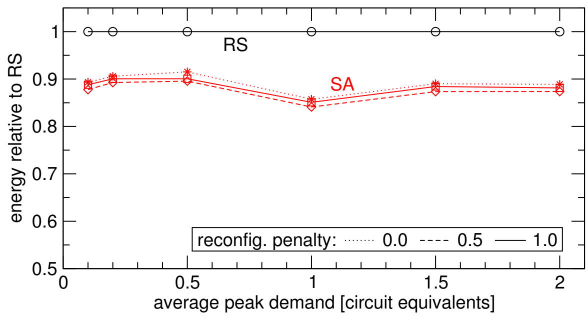
(a) Nobel-Germany, $T_{\mathrm{Cm}} = 0$



(b) Nobel-Germany, $T_{\mathrm{Cm}} = 0.5\Delta T$



(c) Germany50, $T_{\mathrm{Cm}} = 0$



(d) Germany50, $T_{\mathrm{Cm}} = 0.5\Delta T$

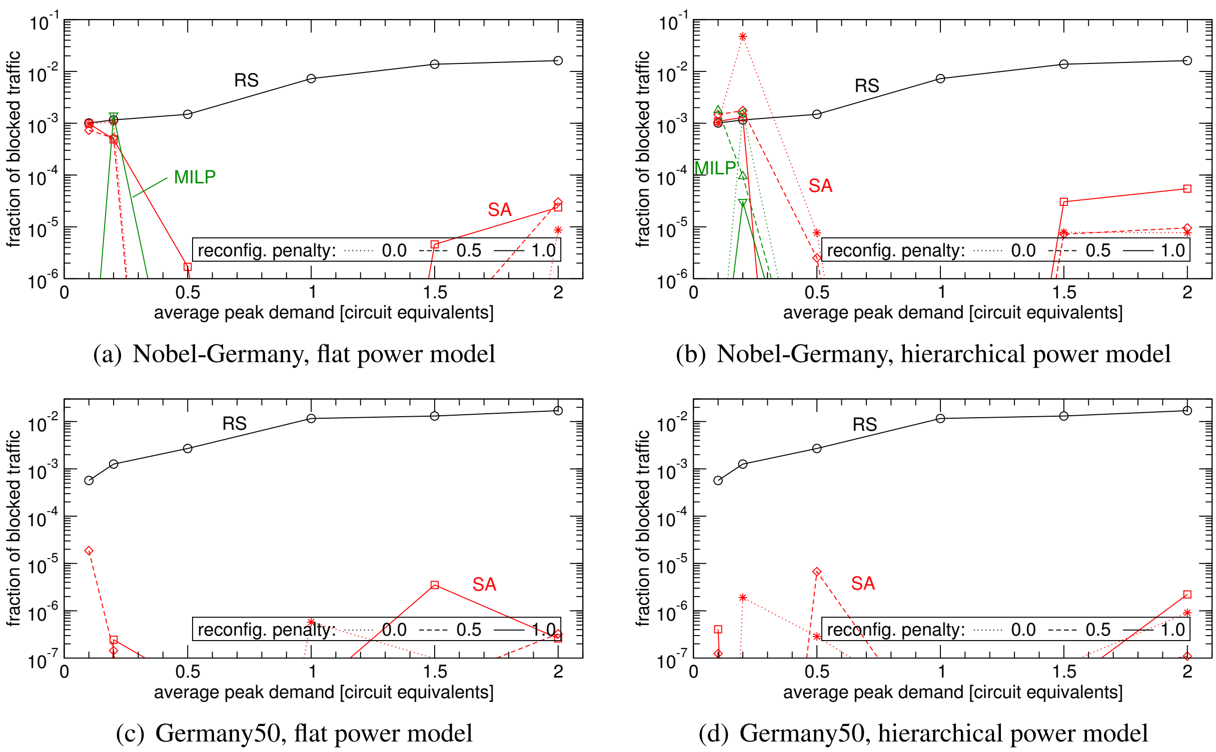
Figure B.6: Load-dependent energy consumption relative to resource scaling for the hierarchical power model and $\sigma_{\mathrm{dim}} = 1.0$ per solution method and reconfiguration penalty over the traffic load

B.4 Traffic Blocking

For resources (under-)dimensioned with $\sigma_{\text{dim}} = 0.8$, Figure B.7 gives the amount of traffic blocked on virtual links with insufficient capacity[1] during the simulated time period relative to the total offered traffic during this period.

Table B.3 and Table B.4 document the occurrence of blocking events for sufficiently dimensioned resources in the simulations for the Nobel-Germany and Germany50 networks, respectively, along with the concerned traffic volume. Since these volumes are generally very small compared to the traffic offered over the simulated period, we indicate them relative to the average traffic offered within one reconfiguration interval (unlike in Figure B.7).

[1]Since we sum up the traffic volume exceeding the capacity on all links, we disregard the possibility that the loss of traffic on some overloaded links mitigates overload on other links. Hence, this value may be higher than the actually lost traffic.



(a) Nobel-Germany, flat power model
(b) Nobel-Germany, hierarchical power model
(c) Germany50, flat power model
(d) Germany50, hierarchical power model

Figure B.7: Average fraction of blocked traffic for $\sigma_{\text{dim}} = 0.8$ per power model for different solution methods and reconfiguration penalties over the traffic load

Table B.3: Occurrences of blocking for sufficient dimensioning in Nobel-Germany network

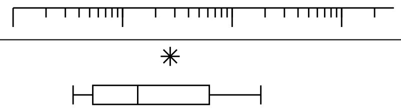
method and problem type					blocking events	
method	σ_{dim}	power model	δ	$\bar{d}_{\text{peak}}$	number	instant. blocked fraction of traffic 10^{-4} 10^{-3} 10^{-2} 10^{-1}
MILP	1.0	hierarchical	0.0	1.00	1	
SA	1.0	hierarchical	0.0	0.10	12	

Table B.4: Occurrences of blocking for sufficient dimensioning in Germany50 network

method and problem type					blocking events	
method	σ_{dim}	power model	δ	$\bar{d}_{\text{peak}}$	number	instant. blocked traffic fraction(s)
SA	1.0	flat	0.0	0.10	1	$1.23 \cdot 10^{-2}$
				1.00	1	$2.38 \cdot 10^{-2}$
				1.50	1	$8.42 \cdot 10^{-7}$
			0.5	0.50	1	$9.37 \cdot 10^{-6}$
			1.0	0.50	1	$2.14 \cdot 10^{-6}$
				1.50	1	$6.59 \cdot 10^{-7}$
		hierarchical	0.0	0.20	1	$1.15 \cdot 10^{-2}$
				0.50	2	$4.38 \cdot 10^{-5}$, $1.13 \cdot 10^{-4}$
				2.00	2	$7.81 \cdot 10^{-6}$, $3.14 \cdot 10^{-2}$
			0.5	1.50	2	$4.20 \cdot 10^{-5}$, $3.52 \cdot 10^{-2}$
			1.0	0.50	1	$5.46 \cdot 10^{-6}$
				2.00	2	$1.92 \cdot 10^{-6}$, $2.72 \cdot 10^{-5}$
	2.0	flat	0.5	0.50	1	$9.93 \cdot 10^{-6}$
			1.0	0.50	1	$6.24 \cdot 10^{-6}$
		hierarchical	0.5	0.10	1	$6.06 \cdot 10^{-4}$
				1.50	1	$2.91 \cdot 10^{-5}$
			1.0	1.50	1	$1.02 \cdot 10^{-3}$

Bibliography

[1] S. Lambert, W. V. Heddeghem, W. Vereecken, B. Lannoo, D. Colle, and M. Pickavet. Worldwide electricity consumption of communication networks. *Optics Express*, 20(26):B513–B524, December 2012. doi:10.1364/OE.20.00B513.

[2] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti. Energy efficiency in the future Internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures. *IEEE Communications Surveys & Tutorials*, 13(2):223–244, 2011. doi:10.1109/SURV.2011.071410.00073.

[3] A. Bianzino, C. Chaudet, D. Rossi, and J. Rougier. A survey of green networking research. *IEEE Communications Surveys & Tutorials*, 14(1):3–20, 2012. doi:10.1109/SURV.2011.113010.00106.

[4] K. Hinton, J. Baliga, M. Feng, R. Ayre, and R. S. Tucker. Power consumption and energy efficiency in the Internet. *IEEE Network*, 25(2):6–12, 2011. doi:10.1109/MNET.2011.5730522.

[5] J. Baliga, R. Ayre, K. Hinton, W. V. Sorin, and R. S. Tucker. Energy consumption in optical IP networks. *Journal of Lightwave Technology*, 27(13):2391–2403, July 2009.

[6] DE-CIX traffic statistics [online]. URI: http://www.de-cix.net/about/statistics/ [Retrieved on October 05, 2014].

[7] M. Channegowda, R. Nejabati, M. R. Fard, S. Peng, N. Amaya, G. Zervas, D. Simeonidou, R. Vilalta, R. Casellas, R. Martínez, R. Muñoz, L. Liu, T. Tsuritani, I. Morita, A. Autenrieth, J. Elbers, P. Kostecki, and P. Kaczmarek. Experimental demonstration of an OpenFlow based software-defined optical network employing packet, fixed and flexible DWDM grid technologies on an international multi-domain testbed. *Optics Express*, 21(5):5487–5498, March 2013. doi:10.1364/OE.21.005487.

[8] B. Mukherjee. *Optical WDM Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[9] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: an approach to high bandwidth optical WAN's. *IEEE Transactions on Communications*, 40(7):1171–1182, 1992. doi:10.1109/26.153361.

[10] G. Rizzelli, A. Morea, M. Tornatore, and O. Rival. Energy efficient traffic-aware design of on–off multi-layer translucent optical networks. *Computer Networks*, 56(10):2443–2455, 2012. doi:10.1016/j.comnet.2012.03.013.

[11] G. Agrawal and D. Medhi. Lightpath topology configuration for wavelength-routed IP/MPLS networks for time-dependent traffic. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–5, November 2006. doi:10.1109/GLOCOM.2006.35.

[12] R. Huelsermann, M. Gunkel, C. Meusburger, and D. A. Schupke. Cost modeling and evaluation of capital expenditures in optical multilayer networks. *Journal of Optical Networking*, 7(9):814–833, September 2008. doi:10.1364/JON.7.000814.

[13] F. Rambach, B. Konrad, L. Dembeck, U. Gebhard, M. Gunkel, M. Quagliotti, L. Serra, and V. López. A multilayer cost model for metro/core networks. *Journal of Optical Communications and Networking*, 5(3):210–225, Mar 2013. doi:10.1364/JOCN.5.000210.

[14] W. van Heddeghem, F. Idzikowski, W. Vereecken, D. Colle, M. Pickavet, and P. Demeester. Power consumption modeling in optical multilayer networks. *Photonic Network Communications*, 24, 2012.

[15] U. Bauknecht and F. Feller. Dynamic resource operation and power model for IP-over-WSON networks. In *Proceedings of the 19th EUNICE Workshop on Advances in Communication Networks*, Chemnitz, Germany, August 2013.

[16] K.-C. Lee and V. Li. A wavelength-convertible optical network. *Journal of Lightwave Technology*, 11(5):962–970, May 1993. doi:10.1109/50.233260.

[17] K. Littger. *Optimierung: Eine Einführung in rechnergestützte Methoden und Anwendungen*. Springer-Verlag GmbH, 1992.

[18] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[19] W. Domschke and A. Drexl. *Einführung in Operations Research*. Springer-Lehrbuch. Springer-Verlag GmbH, 2007.

[20] A. van de Liefvoort. Introduction to complexity theory and NP-completeness. In M. Pióro and D. Medhi, editors, *Routing, Flow, and Capacity Design in Communication and Computer Networks*, chapter Appendix B, pages 625–638. Morgan Kaufmann, 2004.

[21] F. Idzikowski, S. Orlowski, C. Raack, H. Woesner, and A. Wolisz. Dynamic routing at different layers in IP-over-WDM networks – maximizing energy savings. *Optical Switching and Networking*, 8(3):181–200, 2011. doi:10.1016/j.osn.2011.03.007.

[22] S. Gunreben. *Framework for Systematic Evaluation of Protocol Performance with Respect to Out-of-Sequence Packet Arrivals*. PhD thesis, Universität Stuttgart, December 2010.

[23] T. Ott, T. Bogovic, T. Carpenter, K. R. Krishnan, and D. Shallcross. Algorithms for flow allocation for multi protocol label switching. Technical report, Telcordia Technologies, Inc, Morristown, NJ, August 2000.

[24] M. Tornatore, G. Maier, and A. Pattavina. WDM network optimization by ILP based on source formulation. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 1813–1821, June 2002. doi:10.1109/INFCOM.2002.1019435.

[25] K. Zhu and B. Mukherjee. Traffic grooming in an optical WDM mesh network. *IEEE Journal on Selected Areas in Communications*, 20(1):122–133, 2002. doi:10.1109/49.974667.

[26] P. N. Tran and U. Killat. Dynamic reconfiguration of logical topology for WDM networks under traffic changes. In *Proceedings of the IEEE Network Operations and Management Symposium (NOMS)*, pages 279–286, April 2008. doi:10.1109/NOMS.2008.4575145.

[27] IBM. *IBM ILOG CPLEX V12.1 – User's Manal for CPLEX*, 2009.

[28] T. Achterberg. SCIP: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, July 2009.

[29] G. B. Dantzig. *Linear programming and its extensions*. Princeton University Press, Princeton, NJ, 1963.

[30] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.

[31] G. Leguizamón, C. Blum, and E. Alba. Evolutionary computation. In T. F. Gonzalez, editor, *Handbook of Aproximation Algorithms and Metaheuristics*, chapter 24, pages 24.1–24.15. Chapman & Hall/CRC, 2007.

[32] R. Solis-Oba. Local search. In T. F. Gonzalez, editor, *Handbook of Aproximation Algorithms and Metaheuristics*, chapter 18, pages 18.1–18.15. Chapman & Hall/CRC, 2007.

[33] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, September 2003. doi:10.1145/937503.937505.

[34] H. H. Hoos and T. Sützle. Stochastic local search. In T. F. Gonzalez, editor, *Handbook of Aproximation Algorithms and Metaheuristics*, chapter 19, pages 19.1–19.13. Chapman & Hall/CRC, 2007.

[35] F. Glover, M. Laguna, and R. Martií. Principles of tabu search. In T. F. Gonzalez, editor, *Handbook of Aproximation Algorithms and Metaheuristics*, chapter 23, pages 23.1–23.12. Chapman & Hall/CRC, 2007.

[36] E. Aarts, J. Korst, and W. Michiels. Simulated annealing. In T. F. Gonzalez, editor, *Handbook of Aproximation Algorithms and Metaheuristics*, chapter 25, pages 25.1–25.11. Chapman & Hall/CRC, 2007.

[37] M. Necker. *A Novel Algorithm for Distributed Dynamic Interference Coordination in Cellular OFDMA Networks*. PhD thesis, Universität Stuttgart, 2009.

[38] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[39] V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985. doi:10.1007/BF00940812.

[40] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. doi:10.1063/1.1699114.

[41] M. D. Huang, F. Romeo, and A. Sangiovanni-Vincentelli. An efficient general cooling schedule for simulated annealing. In *Proceedings of the IEEE International Conference on Computer-aided Design (ICCAD)*, pages 381–384, Santa Clara, California, November 1986.

[42] T. Rachlitz. Untersuchung der Leistungsfähigkeit und Parametrisierung eines auf Simulated Annealing basierenden Algorithmus für dynamische optische Bypass-Schaltung. Bachelorarbeit, Institute of Communication Networks and Computer Engineering, Universität Stuttgart, September 2013.

[43] J. Sommer. *Verfahren zur Kostenoptimierung von Topologien eingebetteter Kommunikationsnetze*. PhD thesis, Universität Stuttgart, 2011.

[44] H. Höller and S. Voß. A heuristic approach for combined equipment-planning and routing in multi-layer SDH/WDM networks. *European Journal of Operational Research*, 171(3):787–796, 2006. doi:10.1016/j.ejor.2004.09.006.

[45] S. De Maesschalck, M. Pickavet, D. Colle, and P. Demeester. Multi-layer traffic grooming in networks with an IP/MPLS layer on top of a meshed optical layer. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, volume 5, pages 2750–2754, 2003. doi:10.1109/GLOCOM.2003.1258736.

[46] R. Sabella, M. Settembre, G. Oriolo, F. Razza, F. Ferlito, and G. Conte. A multilayer solution for path provisioning in new-generation optical/MPLS networks. *Journal of Lightwave Technology*, 21(5):1141, May 2003.

[47] T. Cinkler, D. Marx, C. Larsen, and D. Fogaras. Heuristic algorithms for joint configuration of the optical and electrical layer in multi-hop wavelength routing networks. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 2, pages 1000–1009, 2000. doi:10.1109/INFCOM.2000.832275.

[48] R. Ramaswami and K. Sivarajan. Design of logical topologies for wavelength-routed optical networks. *IEEE Journal on Selected Areas in Communications*, 14(5):840–851, June 1996. doi:10.1109/49.510907.

[49] D. Banerjee and B. Mukherjee. Wavelength-routed optical networks: linear formulation, resource budgeting tradeoffs, and a reconfiguration study. *IEEE/ACM Transactions on Networking*, 8(5):598–607, October 2000. doi:10.1109/90.879346.

[50] K. Bala, G. Ellinas, M. Post, C.-C. Shen, J. Wei, and N. Antoniades. Towards hitless reconfiguration in WDM optical networks for ATM transport. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, volume 1, pages 316–320, London, UK, November 1996.

[51] N. Skorin-Kapov, P. Pavon-Marino, B. Garcia-Manrubia, and R. Aparicio-Pardo. Scheduled virtual topology design under periodic traffic in transparent optical networks. In *Proceedings of the 6th International Conference on Broadband Communications, Networks, and Systems (BROADNETS)*, pages 1–8, September 2009. doi:10.4108/ICST.BROADNETS2009.7120.

[52] N. Sengezer and E. Karasan. Multi-layer virtual topology design in optical networks under physical layer impairments and multi-hour traffic demand. *IEEE/OSA Journal of Optical Communications and Networking*, 4(2):78–91, February 2012. doi:10.1364/JOCN.4.000078.

[53] G. N. Rouskas and M. H. Ammar. Dynamic reconfiguration in multihop WDM networks. *Journal of High Speed Networks*, 4(3):221–238, 1995.

[54] A. Gençata and B. Mukherjee. Virtual-topology adaptation for WDM mesh networks under dynamic traffic. *IEEE/ACM Transactions on Networking*, 11(2):236–247, 2003. doi:10.1109/TNET.2003.810319.

[55] B. Ramamurthy and A. Ramakrishnan. Virtual topology reconfiguration of wavelength-routed optical WDM networks. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, volume 2, pages 1269–1275, 2000. doi:10.1109/GLOCOM.2000.891340.

[56] P. Tran and U. Killat. Distributed algorithm for dynamic logical topology reconfiguration in IP over WDM networks. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*, pages 748–756, July 2009. doi:10.1109/ISCC.2009.5202289.

[57] P. Iovanna, R. Sabella, and M. Settembre. A traffic engineering system for multilayer networks based on the GMPLS paradigm. *IEEE Network*, 17(2):28–37, March 2003. doi:10.1109/MNET.2003.1188284.

[58] F. Farahmand, X. Huang, and J. Iue. Efficient online traffic grooming algorithms in WDM mesh networks with drop-and-continue node architecture. In *Proceedings of the 1st International Conference on Broadband Networks (BroadNets)*, pages 180–189, 2004. doi:10.1109/BROADNETS.2004.37.

[59] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee. Dynamic traffic grooming in WDM mesh networks using a novel graph model. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, volume 3, pages 2681–2685, 2002. doi:10.1109/GLOCOM.2002.1189116.

[60] K. Zhu and B. Mukherjee. On-line approaches for provisioning connections of different bandwidth granularities in WDM mesh networks. In *Proceedings of the Optical Fiber Communication Conference and Exhibit (OFC)*, pages 549–551, 2002. doi:10.1109/OFC.2002.1036548.

[61] M. Necker, C. Gauger, and S. Bodamer. A new efficient integrated routing scheme for SDH/SONET-WDM multilayer networks. In *Proceedings of the Optical Fiber Communications Conference (OFC)*, volume 2, pages 487–488, March 2003. doi:10.1109/OFC.2003.1248357.

[62] M. Kodialam and T. V. Lakshman. Integrated dynamic IP and wavelength routing in IP over WDM networks. In *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 1, pages 358–366, 2001. doi:10.1109/INFCOM.2001.916718.

[63] E. Yetginer and G. Rouskas. Power efficient traffic grooming in optical WDM networks. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–6, 2009. doi:10.1109/GLOCOM.2009.5425886.

[64] G. Shen and R. Tucker. Energy-minimized design for IP over WDM networks. *IEEE/OSA Journal of Optical Communications and Networking*, 1(1):176–186, June 2009. doi:10.1364/JOCN.1.000176.

[65] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. Power awareness in network design and routing. In *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM)*, pages 457–465, April 2008. doi:10.1109/INFOCOM.2008.93.

[66] Y. Wu, L. Chiaraviglio, M. Mellia, and F. Neri. Power-aware routing and wavelength assignment in optical networks. In *Proceedings of the 35th European Conference on Optical Communication (ECOC)*, pages 1–2, September 2009.

[67] S. Huang, D. Seshadri, and R. Dutta. Traffic grooming: A changing role in green optical networks. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–6, 2009. doi:10.1109/GLOCOM.2009.5425630.

[68] W. Fisher, M. Suchara, and J. Rexford. Greening backbone networks: reducing energy consumption by shutting off cables in bundled links. In *Proceedings of the first ACM SIGCOMM Workshop on Green Networking*, pages 29–34, New York, NY, USA, 2010. doi:10.1145/1851290.1851297.

[69] M. Zhang, C. Yi, B. Liu, and B. Zhang. GreenTE: Power-aware traffic engineering. In *Proceedings of the 18th IEEE International Conference on Network Protocols (ICNP)*, pages 21–30, October 2010. doi:10.1109/ICNP.2010.5762751.

[70] L. Chiaraviglio, M. Mellia, and F. Neri. Reducing power consumption in backbone networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 1–6, June 2009. doi:10.1109/ICC.2009.5199404.

[71] L. Chiaraviglio, M. Mellia, and F. Neri. Minimizing ISP network energy cost: Formulation and solutions. *IEEE/ACM Transactions on Networking*, 20(2):463–476, April 2012. doi:10.1109/TNET.2011.2161487.

[72] A. Bianzino, C. Chaudet, F. Larroca, D. Rossi, and J. Rougier. Energy-aware routing: A reality check. In *Proceedings of Workshops at the IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1422–1427, December 2010. doi:10.1109/GLOCOMW.2010.5700172.

[73] N. Vasić and D. Kostić. Energy-aware traffic engineering. In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking (e-Energy)*, pages 169–178, New York, NY, USA, 2010. ACM. doi:10.1145/1791314.1791341.

[74] A. A. Kist and A. Aldraho. Dynamic topologies for sustainable and energy efficient traffic routing. *Computer Networks*, 55(9):2271–2288, 2011. doi:10.1016/j.comnet.2011.03.008.

[75] J. Cardona Restrepo, C. Gruber, and C. Machuca. Energy profile aware routing. In *Proceedings of Workshops at the IEEE International Conference on Communications (ICC)*, pages 1–5, June 2009. doi:10.1109/ICCW.2009.5208041.

[76] B. Puype, W. Vereecken, D. Colle, M. Pickavet, and P. Demeester. Power reduction techniques in multilayer traffic engineering. In *Proceedings of the 11th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4, July 2009. doi:10.1109/ICTON.2009.5184981.

[77] E. Amaldi, A. Capone, and L. Gianoli. Energy-aware IP traffic engineering with shortest path routing. *Computer Networks*, 57(6):1503–1517, 2013. doi:10.1016/j.comnet.2013.02.006.

[78] Y. Zhang, M. Tornatore, P. Chowdhury, and B. Mukherjee. Energy optimization in IP-over-WDM networks. *Optical Switching and Networking*, 8(3):171–180, 2011. doi:10.1016/j.osn.2011.03.006.

[79] B. Addis, A. Capone, G. Carello, L. Gianoli, and B. Sanso. Energy-aware multiperiod traffic engineering with flow-based routing. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 5957–5961, June 2012. doi:10.1109/ICC.2012.6364824.

[80] A. P. Bianzino, L. Chiaraviglio, M. Mellia, and J.-L. Rougier. GRiDA: Green distributed algorithm for energy-efficient IP backbone networks. *Computer Networks*, 56(14):3219–3232, 2012. doi:10.1016/j.comnet.2012.06.011.

[81] A. Coiro, M. Listanti, A. Valenti, and F. Matera. Energy-aware traffic engineering: A routing-based distributed solution for connection-oriented IP networks. *Computer Networks*, 57(9):2004–2020, June 2013. doi:10.1016/j.comnet.2013.03.017.

[82] J. Scharf. Efficiency analysis of distributed dynamic optical bypassing heuristics. In *Proceedings of the Workshop on Green Communications and Networking at the IEEE International Conference on Communications (ICC)*, June 2012.

[83] F. Idzikowski, L. Chiaraviglio, and E. Bonetto. EWA: An adaptive algorithm for energy saving in IP-over-WDM networks. In *Proceedings of the 17th European Conference on Networks and Optical Communications (NOC)*, pages 1–6, June 2012. doi:10.1109/NOC.2012.6249922.

[84] E. Bonetto, L. Chiaraviglio, F. Idzikowski, and E. L. Rouzic. Algorithms for the multi-period power-aware logical topology design with reconfiguration costs. *Journal of Optical Communications and Networking*, 5(5):394–410, May 2013. doi:10.1364/JOCN.5.000394.

[85] K. Zhu and B. Mukherjee. A review of traffic grooming in WDM optical networks: Architectures and challenges. *Optical Networks Magazine*, 4(2):55–64, 2003.

[86] J. Bannister, L. Fratta, and M. Gerla. Topological design of the wavelength-division optical network. In *Proceedings of the 9th Annual Joint Conference of the IEEE Computer and Communication Societies: The Multiple Facets of Integration (INFOCOM)*, volume 3, pages 1005–1013, 1990. doi:10.1109/INFCOM.1990.91351.

[87] J.-F. Labourdette and A. Acampora. Logically rearrangeable multihop lightwave networks. *IEEE Transactions on Communications*, 39(8):1223–1230, 1991. doi:10.1109/26.134012.

[88] S. Huang and R. Dutta. Dynamic traffic grooming: the changing role of traffic grooming. *IEEE Communications Surveys & Tutorials*, 9(1):32–50, 2007. doi:10.1109/COMST.2007.358970.

[89] R. Dutta and G. N. Rouskas. A survey of virtual topology design algorithms for wavelength routed optical networks. *Optical Networks*, 1:73–89, 2000.

[90] L. Sahasrabuddhe and B. Mukherjee. Light trees: optical multicasting for improved performance in wavelength routed networks. *IEEE Communications Magazine*, 37(2):67–73, February 1999. doi:10.1109/35.747251.

[91] R. Dutta and G. Rouskas. Traffic grooming in WDM networks: past and future. *IEEE Network*, 16(6):46–56, 2002. doi:10.1109/MNET.2002.1081765.

[92] M. Köhn, S. Bodamer, C. Gauger, S. Gunreben, G. Hu, and D. Sass. Comparison of IP/WDM transport network architectures for dynamic data traffic. In *Proceedings of the 11th European Conference on Networks and Optical Communications (NOC)*, July 2006.

[93] M. Pickavet, P. Demeester, D. Colle, D. Staessens, B. Puype, L. Depré, and I. Lievens. Recovery in multilayer optical networks. *Journal of Lightwave Technology*, 24(1):122, January 2006.

[94] W. Golab and R. Boutaba. Policy-driven automated reconfiguration for performance management in WDM optical networks. *IEEE Communications Magazine*, 42(1):44–51, January 2004.

[95] B. G. Józsa and M. Makai. On the solution of reroute sequence planning problem in MPLS networks. *Computer Networks*, 42(2):199–210, 2003. doi:10.1016/S1389-1286(03)00189-0.

[96] A. Farrel and I. Bryskin. *GMPLS: Architecture and Applications*. Morgan Kaufmann series in networking. Elsevier, 2006.

[97] Open Networking Foundation. Software-defined networking: The new norm for networks. White paper, Open Networking Foundation, Palo Alto, CA, USA, April 2012. URI: http://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf.

[98] H. Zang, J. P. Jue, and B. Mukherjee. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *Optical Networks Magazine*, 1:47–60, 2000.

[99] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[100] M. Gupta and S. Singh. Greening of the Internet. In *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, pages 19–26, New York, NY, USA, 2003. doi:10.1145/863955.863959.

[101] Y. Zhang, P. Chowdhury, M. Tornatore, and B. Mukherjee. Energy efficiency in telecom optical networks. *IEEE Communications Surveys & Tutorials*, 12(4):441–458, 2010. doi:10.1109/SURV.2011.062410.00034.

[102] E. Palkopoulou, D. Schupke, and T. Bauschert. Energy efficiency and CAPEX minimization for backbone network planning: Is there a tradeoff? In *Proceedings of the IEEE 3rd International Symposium on Advanced Networks and Telecommunication Systems (ANTS)*, pages 1–3, December 2009. doi:10.1109/ANTS.2009.5409867.

[103] L. Roberts. A radical new router. *IEEE Spectrum*, 46(7):34–39, July 2009. doi:10.1109/MSPEC.2009.5109450.

[104] M. Baldi and Y. Ofek. Time for a "greener" Internet. In *Proceedings of Workshops at the IEEE International Conference on Communications (ICC)*, pages 1–6, June 2009. doi:10.1109/ICCW.2009.5208036.

[105] Y. Chen, T. Wang, and R. Katz. Energy efficient Ethernet encodings. In *Proceedings of the 33rd IEEE Conference on Local Computer Networks (LCN)*, pages 122–129, October 2008. doi:10.1109/LCN.2008.4664160.

[106] The ICT STRONGEST Project [online]. URI: http://www.ict-strongest.eu [Retrieved on June 04, 2014].

[107] The TREND Network of Excellence [online]. URI: http://www.fp7-trend.eu [Retrieved on June 04, 2014].

[108] P. J. Kühn and M. E. Mashaly. Automatic energy efficiency management of data center resources by load-dependent server activation and sleep modes. *Ad Hoc Networks*, in press, 2014. doi:10.1016/j.adhoc.2014.11.013.

[109] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing router buffers. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, SIGCOMM '04, pages 281–292, New York, NY, USA, 2004. ACM. doi:10.1145/1015467.1015499.

[110] A. Bononi and L. Rusch. Doped-fiber amplifier dynamics: a system perspective. *Journal of Lightwave Technology*, 16(5):945–956, May 1998.

[111] F. Feller. Evaluation of a centralized method for one-step multi-layer network reconfiguration. In *Proceedings of the 24th Tyrrhenian International Workshop on Digital Communications (TIWDC)*, Genoa, Italy, September 2013.

[112] F. Feller. An optimization-heuristic approach to dynamic optical bypassing. In *Proceedings of the 13th ITG Symposium on Photonic Networks*, Leipzig, Germany, May 2012.

[113] F. Feller. Evaluation of centralized solution methods for the dynamic optical bypassing problem. In *Proceedings of the 17th International Conference on Optical Network Design & Modeling (ONDM)*, Brest, France, April 2013.

[114] K. Schöck. Untersuchung von MILP-basierten Lösungsstrategien für die dynamische Multilayer-Netz-Optimierung. Diplomarbeit, Institute of Communication Networks and Computer Engineering, Universität Stuttgart, December 2013.

[115] F. J. Briones Rodríguez. Genetic algorithm based optimization strategy for dynamic optical bypassing. Master's thesis, Institute of Communication Networks and Computer Engineering, Universität Stuttgart, November 2012.

[116] J. Sommer, E. A. Doumith, and Q. Duval. On link harness optimization of embedded Ethernet networks. In *Proceedings of the IEEE Symposium on Industrial Embedded Systems (SIES)*, July 2009.

[117] M. Ruffini, D. O'Mahony, and L. Doyle. Optical IP switching: A flow-based approach to distributed cross-layer provisioning. *IEEE/OSA Journal of Optical Communications and Networking*, 2(8):609–624, August 2010. doi:10.1364/JOCN.2.000609.

[118] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly. SNDlib 1.0–Survivable Network Design Library. *Networks*, 55(3):276–286, 2010. doi:10.1002/net.20371.

[119] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon. Providing public intradomain traffic matrices to the research community. *ACM SIGCOMM Computer Communication Review*, 36:83–86, January 2006. doi:10.1145/1111322.1111341.

[120] P. Lothberg. TeraStream – a simplified IP network service delivery model. Presentation at RIPE 67, Athens, Greece, October 2013. URI: https://ripe67.ripe.net/presentations/131-ripe2-2.pdf.

[121] M. Barisch, C. Gauger, M. Necker, K. Dolzer, S. Bodamer, J. Sommer, I. Kauffmann, and J. Scharf. *IKR Simulation Library 2.7 Reference Guide*. Institut für Kommunikationsnetze und Rechnersysteme, Universität Stuttgart, March 2010.

[122] IKR Simulation Library [online]. URI: http://www.ikr.uni-stuttgart.de/Content/IKRSimLib [Retrieved on June 04, 2014].

[123] JGraphT Library [online]. URI: http://jgrapht.org [Retrieved on June 04, 2014].

[802.1ay] IEEE Computer Society. 802.1ay: Draft Standard for Local and Metropolitan Area Networks–Virtual Bridged Local Area Networks, Amendment: Provider Backbone Bridge–Traffic Engineering, December 2007.

[RFC 2991] D. Thaler and C. Hopps. Multipath Issues in Unicast and Multicast Next-Hop Selection. RFC 2991, IETF, November 2000.

[RFC 3031] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031, IETF, January 2001.

[RFC 3410] J. Case, R. Mundy, D. Partain, and B. Stewart. Introduction and Applicability Statements for Internet-Standard Management Framework. RFC 3410, IETF, December 2002.

[RFC 3630] D. Katz, K. Kompella, and D. Yeung. Traffic Engineering (TE) Extensions to OSPF Version 2. RFC 3630, IETF, September 2003.

[RFC 3945] E. Mannie (Editor). Generalized Multi-Protocol Label Switching (GMPLS) Architecture. RFC 3945, IETF, October 2004.

[G.694.1] ITU-T. Spectral grids for WDM applications: DWDM frequency grid. Rec. G.694.1, ITU-T, June 2002.

[G.800] ITU-T. Unified functional architecture of transport networks. Rec. G.800, ITU-T, February 2012.

[G.805] ITU-T. Generic functional architecture of transport networks. Rec. G.805, ITU-T, March 2000.

[G.8010/Y.1306] ITU-T. Architecture of Ethernet layer networks. Rec. G.8010/Y.1306, ITU-T, February 2004.

[G.8080/Y.1304] ITU-T. Architecture for the automatically switched optical network (ASON). Rec. G.8080/Y.1304, ITU-T, November 2001.

[G.8110/Y.1370] ITU-T. MPLS layer network architecture. Rec. G.8110/Y.1370, ITU-T, January 2005.

Acknowledgments

This thesis concludes more than six years of great experiences, challenges fostering my personal development, hard work, and fun in a friendly and inspiring environment at the Institute of Communication Networks and Computer Engineering (IKR). I appreciate the confidence the heads of the institute put into me in granting much freedom for the tasks in research projects and teaching they entrusted me with. Striving to attain the high quality standards of the institute, I was able to enhance my methodological and scientific skills thanks to the continuous open and constructive feedback by my superiors and colleagues. I am likewise grateful for the opportunity to develop my interpersonal skills by supervising students and interacting with many partners in large research projects.

First of all, I would like to thank Prof. em. Dr.-Ing. Dr. h. c. mult. Paul J. Kühn for giving me this great opportunity by employing me as a research staff member, and for his guidance and advice especially during the last phase of work on this thesis. I appreciated the insight he gave me into his work when I assisted him in the lecture on teletraffic theory. I am also grateful for an earlier intercultural experience he enabled by establishing and admitting me to the double-degree diploma program with the French engineering school Télécom ParisTech.

I am grateful to Prof. Dr.-Ing. Andreas Kirstädter, the current head of the institute, for maintaining the spirit of freedom and trust that enabled many valuable experiences in project and teaching work. I appreciated assisting him in revising and giving his computer engineering lecture and I would like to thank him for evaluating this thesis. I would also like to thank Prof. Dr.-Ing. Thomas Bauschert, whom I met at several conferences, for the interesting discussions we had and for his readiness to provide his expertise in the evaluation of this thesis.

I would further like to thank Ulrich Gemkow, the deputy head of the IKR, for his continuous valuable feedback, his support, and for ensuring the administrative framework which allowed me to make diverse experiences and to complete this thesis. I am particularly indebted to him for his advice in confining and refining the topic of this thesis.

The topic of this thesis has evolved in thorough and fruitful discussions with different people, and its creation has been fostered by the supportive, congenial, open-minded, and social atmosphere at the IKR. I am very grateful to all active and former colleagues for creating this environment. In particular, I would like to thank the members of the former high speed networks group and its successor, the fixed networks group, for many formative discussions. In chronological order: Ulrich Gemkow, Martin Köhn, Detlef Saß, Sebastian Gunreben, Simon Hauger, Arthur Mutter, Jochen Kögel, Joachim Scharf, Marc Barisch, Elisabeth Georgieva, Mirja Kühlewind, Sebastian Meier, Domenic Teuchert, David Wagner, Alexander Vensmer, and Uwe Bauknecht. I am grateful to Joachim Scharf for mentoring me during the first years and for introducing me to the topic of optical bypassing. I am particularly indebted to Sebastian Scholz, Uwe Bauknecht, David Wagner, and Magnus Proebster, who reviewed parts of my manuscript and provided valuable feedback. I would further like to thank Magnus Proebster, with whom I shared the office for more than four years, David Wagner, and Christian Blankenhorn for personal and technical advice.

I am grateful to a number of students who contributed to this thesis – knowingly or not – by discussions, preliminary investigations performed, and pieces of software created within their bachelor, master, or diploma theses projects. In particular, I would like to thank Francisco Javier Briones Rodríguez, Thilo Rachlitz, and Karsten Schöck.

The topic of this thesis originated in the STRONGEST project and it has been refined in the SASER project. I am grateful to all people involved, whom I cannot enumerate individually here, for the good cooperation, discussions, and feedback at diverse project meetings. I would particularly like to thank our partners at former Alcatel-Lucent Bell Labs, with whom I have worked closely on several topics: Gert Eilenberger, Lars Dembeck, Wolfram Lautenschläger, Jens Milbrandt, and Ulrich Gebhard.

Pursuing, and in particular, writing a thesis requires – among others – patience. I am grateful to my relatives and friends who encouraged and supported me during this time, and I want to apologize for neglecting some relations. I would particularly like to thank Laura for supporting me – and bearing my moods – during our common time. Last but not least, I am deeply indebted to my late parents for providing me an inspiring and save environment to grow up, fostering a diverse range of interests, backing all my decisions, and giving me any possible support for my studies and my subsequent work in the academic field, which led to this thesis.