Network state advertisement and $p$-cycle protection for reliable connections

PhD Thesis
2011

János Szigeti
PhD candidate
szigeti@tmit.bme.hu

Supervisor:
Tibor Cinkler, PhD
Dept. Telecommunications and Media Informatics, BME
Contents

1 Introduction ........................................... 1
  1.1 Multilayer optical networks .......................... 2
     1.1.1 Optical switching ............................... 4
     1.1.2 Optical switching devices ....................... 6
     1.1.3 Provisioning Label Switched Paths (LSPs) ........ 8
     1.1.4 Controlling of multilayer optical networks ........ 9
     1.1.5 Routing and Wavelength Assignment (RWA) supported
          by Wavelength Graph ............................. 10
     1.1.6 Network model for topology advertisement ........ 12
     1.1.7 Inter-domain routing ............................. 15
  1.2 Quality-assured routing ............................... 17
     1.2.1 Estimating the link availability metric ........... 18
     1.2.2 Resilience schemes ............................... 18
     1.2.3 Metrics of connection availability ................ 19
  1.3 $p$-Cycle Protection ................................ 21
     1.3.1 Resource efficiency of $p$-cycles ................. 21
     1.3.2 $p$-Cycles and multiple failures .................. 22
     1.3.3 Failure handling priority ........................ 23
     1.3.4 Selecting candidate $p$-cycles in the network ..... 25
     1.3.5 $p$-Cycle assignment to protect connections ........ 26

2 Enhancing topology advertisement ..................... 28
  2.1 Topology aggregation on additive link metrics ......... 28
  2.2 Rule-Based Topology Advertisement (RBTA) ............... 29
     2.2.1 Problem of Wavelength Graph representation ....... 30
     2.2.2 Principle of RBTA ............................... 31
     2.2.3 Description of the messages ...................... 32
     2.2.4 Constructing the WG ............................. 33
  2.3 Generalized Protection Formula (GPF) ................... 34
     2.3.1 Scope of GPF ..................................... 35
     2.3.2 Transforming the protection schemes to fit into GPF .. 35
     2.3.3 Definition of GPF ................................ 36
     2.3.4 Simple GPF example .............................. 38
2.4 Topology description supporting protection (RBTA-P) . . . . 39
  2.4.1 Advertised topology information . . . . . . . . . . . . 39
  2.4.2 Structure of the stored data . . . . . . . . . . . . . . 40
  2.4.3 Path computation . . . . . . . . . . . . . . . . . . . . 41
  2.4.4 Novel failure handling paradigm . . . . . . . . . . . . 41
2.5 Complexity and scalability of RBTA . . . . . . . . . . . . . 41
  2.5.1 Initial topology advertisement . . . . . . . . . . . . . 43
  2.5.2 Dynamic link state information . . . . . . . . . . . . . 43
  2.5.3 TED size . . . . . . . . . . . . . . . . . . . . . . . . . 44
  2.5.4 Path computation complexity . . . . . . . . . . . . . . 45
  2.5.5 SRG identification and failure notification . . . . . . . 46
3 Inter-domain \(p\)-cycle protection 49
  3.1 Planning reliable inter-domain connections . . . . . . . . . 50
    3.1.1 Two-level hierarchical network model . . . . . . . . . 50
    3.1.2 Defining the set of candidate MDPCs . . . . . . . . . 51
    3.1.3 Assigning candidate MDPCs . . . . . . . . . . . . . . 51
    3.1.4 Intra-domain part of inter-domain \(p\)-cycle . . . . . 51
    3.1.5 Optimization goal . . . . . . . . . . . . . . . . . . . 53
    3.1.6 Achieving further reliability . . . . . . . . . . . . . . 54
  3.2 Performance of inter-domain \(p\)-cycle protection . . . . . 57
    3.2.1 Simulation setup . . . . . . . . . . . . . . . . . . . . 57
    3.2.2 Evaluated performance metrics . . . . . . . . . . . . . 60
    3.2.3 Unavailability reduction . . . . . . . . . . . . . . . . 61
    3.2.4 Resource consumption . . . . . . . . . . . . . . . . . 63
    3.2.5 Getting the desired ratio of connection with predefined
                           availability . . . . . . . . . . . . . . . . . . . . . . . 64
    3.2.6 Performance in overloaded networks . . . . . . . . . . 66
4 Improving connection availability by \(p\)-cycles 68
  4.1 The Exhaustive Grow candidate cycle search algorithm . . . 69
    4.1.1 Comparative results . . . . . . . . . . . . . . . . . . . 70
  4.2 Incremental availability calculation method for \(p\)-cycle-protected
                   connections . . . . . . . . . . . . . . . . . . . . . . . 72
    4.2.1 Scope of the model . . . . . . . . . . . . . . . . . . . 73
    4.2.2 Notations . . . . . . . . . . . . . . . . . . . . . . . . 74
    4.2.3 Mathematical formulation . . . . . . . . . . . . . . . . 76
    4.2.4 Calculation in practice . . . . . . . . . . . . . . . . . 79
    4.2.5 Calculation complexity . . . . . . . . . . . . . . . . . 81
  4.3 The approximation error of the Serial-Parallel calculation model 82
    4.3.1 Single link overlap . . . . . . . . . . . . . . . . . . . 84
    4.3.2 Approximation error of the Serial-Parallel calculation
                   model in case of two overlapping links . . . . . . . . 89
4.3.3 Approximation error in case of more than two overlapping links ........................................ 94
4.3.4 Scenarios illustrating approximation error ................................................................. 101
4.3.5 Summary of results ..................................................................................................... 104

5 Conclusion .................................................................................................................... 107
Chapter 1

Introduction

Telecommunication networks keep growing and evolving even today. At the access part of the networks, from an average end-users’ perspective we see that within the last 15–20 years we have reached a bit rate of 100 Mbps or even higher compared to the 2.4 Kbps speed of the old modem times. We can also state that telecommunication connections have become more reliable and they have a higher availability. Altogether, we can say that – trying to fulfill the growing demands of the consumers – the quality of the telecommunication services offered by the network operators is increasing rapidly. And this means also a steady evolution of the underlying telecommunication technology.

If we take a look at the core of the current networks, we will see that the backbone transfers much more traffic than a few years ago, however, the network operators have to deal not only with this increased amount of traffic but also with the increased quality requirements of the clients. Today, the traffic in the core of the networks is basically carried on optics, since fiber is economically the most suitable medium to transmit large amount of data over long (and also over short) distances. Compared to the conventional electrical transmission, due to its physical attributes and the high transmission speed, optical transmission requires special handling in many aspects.

Concerning the optical networking, there are many research subareas that have been examined recently, most of all, the physical impairments [1], different optical switching techniques, optical burst and packet switching [2][3][4], etc. In this thesis work we focus on problems regarding the network controlling and management:

- “How can the network operators provide connections to the network end-users with the desired level of availability” is an issue where large research efforts are made. There are well known resilience mechanisms to enhance the availability of the connections, however, in inter-domain context, where different networks and network operators have to co-
operate with each other, novel resilience mechanisms are needed.

- If the operators are aware of basic network device availability metrics, they need to estimate the connection availability fast and/or accurately, since, on the one hand, overestimation may lead to unforeseen service downtime and unrest among the end-users, on the other hand, underestimation may lead to overprotection, implying higher resource consumption and – this way – a lower network throughput.

- In case of distributed controlling the state changes of the network devices have to be advertised. The network state changes happen due to resource allocations and deallocations. How can the process of connection provisioning be supported by efficient topology and link state advertisement so that it scales well with the evolution of the network in the dimensions of network size, used wavelengths and number of requests.

These are the main questions this thesis work focuses on. The thesis work is based on our results regarding information advertisement [5][6][7][8][9] (Chapt. 2), inter-domain protection [10][11][12] (Chapt. 3) and availability estimation [13][14] (Chapt. 4). Before presenting the answers on these questions, however, to get familiar within this research topic, to have a general overview, we must take a closer look at the backgrounds. First, we will give an introduction on the optical networking, next on the availability theory and finally on the so-called p-cycle resilience scheme which became very popular during the last decade.

1.1 Multilayer optical networks

The first event when two network devices succeeded in communicating through fiber connection happened already in early 1970’s and since that time we have been witnessing of a permanent evolution in optical telecommunication. The biggest break-through of optical technology was the 1990’s when public network operators began in their core network to replace the copper cables by fiber. That time the optical technology was used only as a transporter medium, which means that the electronic signal carrying the information was transformed into optical impulse at the head end of the fiber and a light detector converted this impulse back into electronic sign at the tail end of the fiber. Routing and traffic switching, however, still happened in the electronic domain. Accordingly this solution was still expensive. This way, the next important milestone of optical networking was the appearance of optical switching devices. Today, the communication networks still keep growing and are getting more and more meshed, therefore – to keep the operational expenses (OPEX) at a moderate level – operators tend to employ optical switching devices instead of electronic ones. However, currently,
there are many switching cases when the optical domain cannot substitute
the electronic domain, thus usually a combination of electronic and optical
switching devices is used (see Sect. 1.1.2).

The optical domain, as fibers offer very low attenuation for wavelengths
within the wide range of 1270 nm < λ < 1610 nm, is suitable to carry
multiple channels on different wavelengths simultaneously. The WDM
(Wavelength Division Multiplexing), offering 40–100 Gb/s, or even DWDM
(Dense WDM) systems also became popular and they are displacing single-
wavelength optical transmission (with 10 Gb/s).

Separated by responsibility and functionality, in general, we can differenti-ate three network planes in current telecommunication networks:
• The Data Plane (DP) (or Forwarding Plane) is responsible to carry
  the traffic of the end-users.
• The Control Plane (CP) has to control the connections (e.g., connection
  setup, release, failure handling: backup switching, etc.)
• The general network management functions are delegated to the Management Plane (MP).

To sum up the trends of the last decades of optical networking, there is a
simultaneous growth of multiple factors. This implies that the operators
of the current core and metro networks, and – since connections become
rather automatically switched by the Control Plane (CP) than permanently
set up by the Management Plane (MP) – most of all the Control Plane
of the networks, must cope simultaneously with an increasing number of
network devices, fibers, wavelengths and connections. All that leads to an
increasing number of control messages calling for controlling channels with
higher capacity.

Recently, two major standardization bodies have made efforts towards
the standardization of the control plane in optical networks: ITU-T has
defined ASTN/ASON [15] while IETF has created the GMPLS framework
[16].

These two actors face the problem of standardization from different as-
pects. The ASON framework of ITU-T is a top-down approach, which de-
fines the elements, the building blocks of the network on different planes
(transport, control and management planes), the responsibilities of the
planes, the interfaces between the elements, etc., but – intentionally – there
are no specific protocols described. Instead, ITU-T forces the manufactur-
ers to standardize protocols fitting into the ASON framework. Indeed, the
Optical Internetworking Forum (OIF) [17] does the last but nevertheless
very difficult step of developing and agreeing in standard protocols from the
recommendations.

Contrarily, the IETF follows a bottom-up method of standard specifi-
cation. There is a maturity process of the appearing standards from the
“proposed standard” entry level evolving to “draft standard” and finally “standard” (Request for Comment, RFC) [18]. About a decade ago, all the optical-related issues were assigned to the GMPLS. Meantime the Common Control and Measurement Plane Working Group (CCAMP) was born addressed to control-related problems of switching (Wavelength Switched Optical Networks, WSON) and routing (Routing and Wavelength Assignment, RWA), later also the Path Computation Element (PCE) Working Group, dealing with label switched routing, also became stand-alone.

To sum up, in the world of standardization of the optical networking, ITU-T, IETF and OIF are of main importance, and in the current work we rely on the terminology used in ASON framework, but also refer to entities defined by IETF.

1.1.1 Optical switching

We can classify the type of the communication by the number of communicating entities. If there is a conversation between two persons, we call it point-to-point communication. Another possibility is the point-to-multipoint arrangement, which is usually an asymmetric form of communication, i.e., either there is one speaker and multiple listeners, or there is a single listener which collects information from multiple speakers. Finally, there are cases when all the entities share a common communication channel, and everybody receives the messages sent by anybody (e.g., Ethernet). Even if every endpoint in the network is –indirectly– connected to every other endpoint, on higher levels of communication, as a communication originator, usually we want to select one ending or a set of endings to which we should be connected and that should receive our messages. There are different switching techniques available and used to support the connection of the communication terminals.

Background of switching

In the beginning, the early telephone networks were built of pairs of telephones connected each to the other by a pair of copper wires. This way, the end-user that wanted to have the possibility to communicate to each one in the network of \( N \) users, needed \( N - 1 \) communication terminals and \( N - 1 \) wire pairs. Then the multiple telephone sets were replaced by a single one with some built-in switching capability, however, the user still had \( N - 1 \) pairs of wire connected to its terminal equipment. Practically, the network formed a full mesh between the end-user terminals with \( N(N - 1)/2 \) links (Fig. 1.1(a)) which was a waste of resources.

As the number of end-users and the size of the network increased, this networking approach became too complex and inefficient. The task of switching was moved to a dedicated switching center. Each terminal was
Switching at the terminals

(b) Switching in a dedicated center

Figure 1.1: Introduction of switching in telephone networks

connected only to this central node (Fig. 1.1(b)). This switching center had the task to connect the wires of two selected terminals to each other whenever such a switching was requested (on demand) by one of the two end-users.

Generally spoken, switching devices can conjugate their input and output lines (ports). The simplest switching devices are the 1x2 and 2x1 switching elements, being able to link one input line to one of the two output lines, selectively, or vice versa. Using multiple instances of these basic elements, even more complex switching devices (e.g., 2x2, 4x4 etc.) can be fabricated. However, due to physical limitations and the complex structure, large non-blocking switching matrices have to be built in other way. A preferred solution is the crossbar switching [19], which means in the electronic domain that the $N$ input and $M$ output lines are arranged into $N$ horizontal and $M$ vertical metal bars in a grid with $N \cdot M$ crosspoints, and an input to output switch is realized by connecting the corresponding horizontal and vertical bars at their crosspoint.

Both the basic 1x2 element and the crossbar switch device have well-known and commonly used implementation in the optical domain (e.g., couplers, liquid crystals, MEMS mirrors, etc.).

Optical specialities

There are many tasks of signal processing that are difficult or cannot be performed in the optical domain.

Although fibers are suitable to transmit signals over long distances (0.2 dB/km attenuation), sections longer than 100–200 km needmediate amplifiers [20]. Due to signal dispersion and other impairment, sometimes even signal reshaping is also needed. Signal amplification can be carried out in the optical domain, however, currently, signal reshaping requires O/E
In optical networking there is another difference compared to the conventional transmission: the usage of wavelength division multiplexing instead of TDM (Time Division Multiplexing). It means that a fiber can carry multiple data channels, each of them identified by the nominal central wavelength of the transmission. If we want to multiplex two data channels into the same fiber, these data channel have to be transmitted on different wavelengths. Analogously, in TDM, the channels need to occupy different time slots, and channel collision can be resolved using signal delays. In WDM these collisions can be resolved by means of wavelength conversion. However, compared to the signal delays applied in TDM, converting a wavelength into a different wavelength is difficult.

The most usual and general solution for wavelength conversion is converting the optical signal into electronic, and then converting it back to optical signal on a different wavelength. In fact, the switching can take place even between the O/E and the E/O conversion, i.e., in the electronic domain. Recently there were also all-optical converters presented which are based on frequency coherent effects or cross-modulation [22]. The advantages of the latter one are the relative lower costs and the less power consumption, whereas the opto-electronic conversion has the advantage, that in the electronic domain not only the physical, but also higher communication layers (e.g., the STS/STM frames) may be switched, offering the possibility of multilayer switching. To reduce costs – taking also into consideration that wavelength conversion is not necessary in each switching case – switching devices with limited conversion capability are very popular.

### 1.1.2 Optical switching devices

Concerning the switching devices in optical networks, there is a large variety available on the market. One can balance between the price and the switching capabilities.

As already denoted, the optical transmission uses the wavelength range $\lambda = 1270 – 1610$ nm where optical fibers have low attenuation (specified in [23]). If we emit a signal on a definite wavelength at the head end of the fiber, the arriving light at the tail end – due to dispersion effects – will spread into a small spectral range around the emitted wavelength. However, compared to the width of the whole low-attenuation spectrum, this spread range is considerably narrow. This way, using high precision transmitters, the range 1270 – 1610 nm is wide enough to carry multiple channels on different wavelengths by means of wavelength division multiplexing (WDM) [21].

In the following we will present some of the most popular WDM devices. Although not all of them are able to perform full cross connecting and some of them have extended capability, for the sake of simplicity we call them
commonly Optical Cross-Connects (OXC).

Note that even though single wavelength devices are also current, here we do not present any of them since they can be classified as a subset of WDM devices, having the number of employed wavelengths reduced to 1.

**Optical Add/Drop Multiplexer (OADM)**

The cheapest way of putting the electronic signal on a fiber bearer is using Optical Add/Drop Multiplexer (OADM) [19] (page 202). Its concept is to demultiplex one wavelength channel from the incoming fiber (drop), and to multiplex the same wavelength to the rest of the passing through traffic (add). We can convert the dropped wavelength into electronic signal at the Rx (receiver) port, and vice versa, the incoming electronic signal into optical signal at Tx (transmitter) port.

If the network topology forms a ring, then there may be established point-to-point communication between OADMs acting on the same wavelength.

**Reconfigurable OADM (ROADM)**

More flexible solution can be achieved if the device can drop multiple wavelengths from the passing through fiber, and if we can select which wavelengths to drop/add. The reconfigurable OADMs (ROADMs) provide this functionality [24].

**Wavelength-selective OXC (WS-OXC)**

In this device, at each input port the fiber is demultiplexed into wavelength channels, and at each wavelength the input channels are cross-connected to the output channels, finally, for each output port the different wavelength channels are multiplexed into a single fiber. This way, for each separate wavelength full cross-connect capability is provided, but there is no possibility to change the wavelength of the input channel to a different wavelength on the output.

**Wavelength-interchangeable OXC (WI-OXC)**

Unlike the WS-OXC, this type of optical cross-connect allows to switch any ingress wavelength to any egress wavelength channel on any output port, since right before the optical multiplexers at the output ports wavelength converters convert any input signal to the appropriate wavelength [25].

This type of OXC is rather only a theoretical one, since the complete rearrangement of the wavelengths is almost never required. Instead, practically, only a reduced set of wavelength channels at the output ports are
equipped with all-optical converters. This cost-efficient type of WI-OXC is referred to as *all-optical OXC with limited wavelength conversion capability*.

**OXC with electronic core**

This kind of optical device is similar to the WS-OXC, but the demultiplexed wavelength channels are immediately converted by O/E converters into electronic signals and the cross-connect switching takes place in the electronic domain.

As already mentioned, the *wavelength conversion*, i.e., switching the data from an ingress wavelength channel to an egress channel of a different wavelength, is not supported by the WS-OXC. However, it can be easily carried out using this type of OXC. Furthermore, if the electronic core is a *time slot interchange* (TSI) device, even *traffic grooming* (higher layer / sub-lambda granular / multilayer switch) [26] is possible.

Due to its enhanced strength in switching, this type of OXC has also higher installation and maintenance costs.

**OXC with limited electronic processing capability**

There are one or more dedicated ports or wavelength channels of the WS-OXC that are interconnected via O/E and E/O converters with an electronic switch. Compared to the OXC with electronic core, this device offers reduced wavelength conversion capability – but also reduced costs. The capacity of wavelength conversion is limited by the number of OE/EO converting ports.

**Further switching possibilities**

There are also switching solutions that treat the spectrum of the wavelengths as a set of *wavebands*. One waveband contains multiple neighboring wavelength channels. From the point of view of the wavelength switching, this waveband switching OXC introduces switching constraints, however, it also reduces the cost of switching devices [27].

There are several other switching solutions, especially in the world of small devices (e.g., $2 \times 2$ switches [28]). Taking just a short overview, we do not enumerate them here.

**1.1.3 Provisioning Label Switched Paths (LSPs)**

If we talk about connections, we mean a bi-directional virtual circuit that is established between two network equipment on the Data Plane of the network.

Label switching is a routing technique first introduced in Frame Relay [29][30], next in ATM (Asynchronous Transfer Method), and currently
employed in MPLS (Multiprotocol Label Switching) [31] and GMPLS (Generalized MPLS) [16]. The main advantage of label switching is the ability of path pre-definition. Unlike in the traditional hop-by-hop routing (e.g., IPv4), when using label switching it is possible to define the path of connection in advance, that makes a more versatile traffic engineering possible. Similarly to circuit switch, in this architecture the intermediate nodes along the predefined path do not have to make any routing decision whenever a labeled packet/frame arrives, they just switch/forward the packets to a destination associated with the label of the packet. However, as there are no real communication circuits, we call it rather virtual circuit (VC) switch.

The label ↔ destination association for a given label in a given node is set up during the connection provisioning procedure. The connection provisioning is the task of the Control Plane (CP) and of the Management Plane (MP).

To emphasize the importance of the connection provisioning compared to other tasks, in GMPLS controller devices (which is called Optical Connection Controller, OCC in ASON terminology), the task of routing decision making is separated from other general control responsibilities and delegated to the Path Computation Element (PCE) [32].

1.1.4 Controlling of multilayer optical networks

Multilayer networks consist of multiple networking technologies and techniques stacked one over the other, e.g., IP/MPLS/OTN or IP/Ethernet/ngSDH/OTN (Next-generation SDH (Synchronous Digital Hierarchy) over Optical Transport Network).

Nowadays, in a multilayer network, all the lower (i.e., the optical) layers are statically configured either manually or via the MP routines, while the uppermost layer is switched via the CP. This kind of heterogeneous (MP and CP based) configuration is typical for both, IP networks and PSTNs today. However, to reduce the OPEX and to speed up provisioning of new services, in other words, to get a more flexible solution, an extended and smart controlling is needed at the lower layers as well.

Different controlling models

There are multiple ways known of controlling the different layers in multilayer networks. The main question is whether the controlling should take place separately for each layer or jointly.

The simplest solution is when all the layers have their own CP, they are separated from each other, still there is an interaction between them: the upper (client) layer can ask the lower (server) layer to perform specified tasks, provide pre-defined services to the client, but the client cannot modify the state of the server directly, and whenever the state of the lower layer
changes the client layer has to adapt to these changes. This is referred to as *Overlay Model*.

In overlay model the controlling interface between the layers is rather thin, which makes it, on the one hand, easy to implement and realize. On the other hand, the interaction between the layers is quite unidirectional, since the lower layer cannot request services from the upper layer.

If there is advanced information exchange between the CPs of the layers and the CPs are interfaced much denser to each other, then they can make routing decision jointly. This is referred to as *Peer Model* since the layers have analogous capabilities and they are rather in peer than server-client relation.

Besides the Overlay and Peer Interconnection models discussed before, there has been introduced the so-called *vertically integrated* model where the network layers are typically run by the same operator, and instead of having a separated CP per layer, there is a single integrated unified CP for multiple layers. This is the most flexible controlling solution and the most complex as well.

The difference between the enumerated controlling schemes and the strength of the vertically integrated model can be demonstrated spectacularly on scenarios of provisioning a connection with *protection* [33] or routing sub-lambda granular traffic [34][35].

### 1.1.5 Routing and Wavelength Assignment (RWA) supported by Wavelength Graph

In conventional networks, the route for a connection was defined by the sequence of network nodes between the two end nodes. The traffic could be transmitted between two adjacent nodes, if the transmission channel had enough free capacity.

In optical networks, however, each wavelength is a unique transmission channel. Merely enumerating the network nodes along the path is not sufficient to define the route of a connection. Also a wavelength has to be assigned for each transmission link. If the wavelength conversion capability is limited in the network (i.e., not all of the switching devices support full wavelength conversion), the tasks of the routing and the wavelength assignment cannot take place sequentially – they have to be performed jointly. In other words, the routing algorithm must also assign wavelength for the selected links. This way, the task of routing is rather called *Routing and Wavelength Assignment* (RWA) in optical networks.

**Wavelength Graph (WG)**

For the RWA process, compared to the conventional routing, it is not enough to know how many free capacity is available on the network links, but we
must be aware of it per wavelength (channel). Moreover, it is also important to know about the switching capability of the optical cross-connects.

Whenever routing algorithms try to find paths between the source and the destination nodes, the network is typically represented by a graph. In optical networks, to support RWA, i.e., to assign also wavelength channels to the selected path, the routing is performed in a so-called Wavelength Graph (WG)[36][37]. In the WG a single fiber connection between two network nodes is represented by as many edges as the number of wavelengths and the optical cross-connects are directed subgraphs of the WG where the edges are denoting switching and conversion capability. This way, the path that the routing algorithm finds between source and destination nodes implicitly also tells which wavelength to use. Hence the WG representation of the network is very favorable for RWA purposes, but its main drawback comes from its relatively large size.

The WG topology becomes more complex in vertically interconnected multilayer networks, where the WG has to represent both the different optical and the electronic capability of the network nodes (λ (wavelength) usability, λ- and OE/EO-conversion, grooming, etc.) along with the node adjacency.

(a) Simple 4-port OXC  (b) 2-port OXC with 3 E/O ports (not tunable)  (c) 2-port OXC with 2 E/O ports (tunable)

Figure 1.2: Different OXC models represented by a subgraph

Figure 1.2 depicts the subgraph of different OXC models, detailed subgraph representation principles can be read in [9][35][37].

Note, however, that flooding all this information of all layers to the PCEs loads the Control Channel significantly. Although the vertically integrated model offers more detailed topology information to the PCE and there are no interaction limitations between the optical and the electronic (e.g., MPLS) layers, there are some problems with the conventional WG which need to be solved.
(a) Failure risks at the network links
(b) Shared Risk Groups in the Wavelength Graph

Figure 1.3: Grouping WG edges into SRGs

Shared risk link groups (SRLG)

The WG represents an optical fiber by a set of edges. When assigning a protection, typically, the routing algorithms are searching for a path (or paths) in the graph that is node- or edge-disjoint with the default path. The problem is, that using the WG for route assignment, it can happen that even though the default and the protection paths are node and edge disjoint in the WG, they pass through the same fiber. And if that certain fiber fails (e.g., the fiber is cut), then both the default and the protection path will go down.

To overcome this problem, link groups must be defined for those links that share the same risk (see Fig. 1.3). This is the basic idea of Shared Risk (Link) Groups (SRLG,SRG)[38]. The routing algorithms then have to take into account the additional criterion that the default and backup paths are not allowed to share the same risk, i.e., their links cannot be members of the same SRLGs. Note that efficient dual path search algorithms become more complex or even unusable because of this extra criterion [39].

1.1.6 Network model for topology advertisement

Conforming to the ASON model of ITU-T and to the GMPLS model of IETF, we assume a network model (depicted in Fig. 1.4) with the following items: Optical cross-connects (OXC) switch traffic at the Data Plane (DP). Each OXC is controlled by an optical connection controller (OCC). Note that there may be solutions where – for the sake of reliability – there are multiple OCCs assigned to control the same OXC [40], however, in our model we do not emphasize this facility. The OCCs are aware of the underlying data switching architecture, i.e., the switching possibilities and the actual state of the controlled OXCs. Note that the OXC and the OCC frequently resides in the same physical device. The ASON model, however, separates
the data transmitting part and the controller part of an optical switch or router. Some of the controllers – but even each of them may – host PCEs that are making the routing decisions. The PCEs maintain the network topology information in the Traffic Engineering Database (TED). The OCC can act in the role of a Path Computation Client (PCC) as well as in the role of a Data Provider when interfacing the PCE. The end-users send their data traffic to the OXCs whereas they are also directly connected to the OCCs.

Now, all the information necessary for routing must be spread to all network components that are responsible for making routing decisions. Us-

Figure 1.4: Logical network model

Figure 1.5: Control message exchange during connection setup
ing the IETF terminology, these components are the aforementioned PCEs. They maintain their TED for storing topology and link state information. Based on the TED, the PCE can calculate QoS-aware, so-called Traffic Engineering Label Switched Paths (TE LSPs) whenever a client (Path Computation Client, PCC) requests a TE LSP from the PCE. Usually, in case a traffic demand arrives in the network, the ingress node acts in the role of a PCC and requests a TE LSP.

The information necessary for routing is the state of the network devices. Note that the term “state” involves both invariant attributes (e.g., adjacency of the devices, physical attributes) and dynamic information (e.g., free capacity/bandwidths, failure state). Spreading the information is referred to as Link State Advertisement (LSA), that is performed by the Interior Gateway Protocol (IGP).

Figure 1.5 demonstrates the control message flow in the network during a connection setup. The connection setup request arrives in one of the OCCs, practically in the controller of the source node (Step 1). This OCC, acting as a PCC asks the PCE to define a route that fulfills certain Service Level Agreements (SLA) requirements (Step 2). The PCE, based on TED data, constructs or updates the Wavelength Graph and returns the computed path – or, in case of protected connections, paths – to the PCC (Step 3). The originator OCC allocates the resources along the path(s) by sending signaling messages to the corresponding OCCs (Step 4). These OCCs configure the switches of the controlled OXCs (Step 5) and finally, in the role of Data Provider, each of them sends information to the TED(s) stored in PCE(s) about the new state of the controlled OXC (Step 6).

Whenever provisioning a connection in the network we have two major expectations:

- The connection provisioning should not be blocked.
- The established connection should satisfy the QoS requirements.

To achieve these two goals the network unit that is responsible for making routing decisions (i.e., the PCE in IETF terminology) must be aware of sufficient topological and link state information.

At this point, however, there arise scalability issues. As the network grows and the number of devices, fibers or wavelengths increases, there are getting more link states to be advertised on the control channel and to be stored in the TEDs. Furthermore, the increasing number of connection request – which is the premise or the consequence of network grow – is another factor that implies a higher number of LSA messages. To handle this problem, the first step is to expand the capacity of the TED. As the second step, concerning the control channel throughput, we have two possibilities. One way is to expand also the capacity of the control channel by square or
higher polynomial as the network grows, while the other, the preferred, way is to reduce the amount of information to be advertised and processed.

Reducing the topology advertisement overhead

As the number of provisioned connections increases, on the one hand, the signaling overhead – to establish and release routes – increases, on the other hand, there are more link state changes. If we want to keep the data of the TEDs of the PCEs accurate, each link state change must be advertised immediately. This immediate link state advertisement results in high load on the control channels. Furthermore, usually it is unnecessary, since if the inaccuracy of the topology information stored in the TEDs is moderate, the quality of the routing (e.g., the blocking probability or the resource usage compared to the optimal solution) is still acceptable. The information update can be reduced by applying a certain triggering policy [41]. The trigger can be time-, class- or threshold-based.

The time-based triggering does not adapt to the demand arrival intensity and long update periods may lead to high level of inaccuracy accompanied by high rate of blocking [42].

Other policies, like class- or threshold-based ones, trigger information update only in case when the relative spare capacity change on the link exceeds a certain limit. That way these policies give an upper bound to the rate of inaccuracy, keeping the blocking probability at a low level whereas reducing the amount of flooded link state messages. The difference between these two policies is that the class-based policy uses predefined absolute triggering threshold levels, whereas in the threshold-based approach these limits are relative to the last advertised values.

The biggest disadvantage of these triggering policies is that in WDM networks even a marginal inaccuracy of topology information stored in the TED may lead to blocking the connection setup (see Sect. 2.2.1). Therefore, various routing algorithms are proposed [42], [43], [44] to assist the probability of route acceptance.

1.1.7 Inter-domain routing

Due to physical, protocol related and legal limitations the networks cannot grow beyond any size. Another reason to keep the size of the networks relatively small is the maintainability. Furthermore, replacing devices, wires and protocols is easier to carry out in a small network.

The networks are, however, interconnected somehow, since there has been a high demand to establish connections even between end nodes belonging to different networks. In other words, to make internetworking. In such interconnected environment the original networks are called Autonomous Systems (AS) or domains. And the interconnected environment
is referred to as **inter-domain** or **multidomain network**. Within the domain we call the dedicated network devices that are connected to foreign domains **border nodes**. Compared to other network devices, the border nodes have additional control functionalities. In the conventional IP networks the autonomous systems have gateways as border nodes. The gateways use the **Border Gateway Protocol** (BGP4) [45] to communicate with each other and to make inter-domain routing decisions.

Unlike in conventional IP, in optical networks we want to establish virtual circuits – with predefined QoS parameters – instead of forwarding individual data packets. Provisioning inter-domain connection is a rather complex problem, and has multiple aspects. The standardization body ITU-T focuses on the interfaces. Figure 1.6 depicts that the ASON model emphasizes that the OCCs of the neighboring domains may have different capabilities, but they must agree in the communication interface, which is the **Exterior Network-Network Interface** (E-NNI). Official E-NNI implementation agreements are available online at OIF’s webpage [46]. The PCE–PCC concept of the IETF presents what kind of information has to be exchanged between domains and how.

The proposed models implicitly suggest that some topology information has to be advertised even to foreign domains to support inter-domain routing. Of course, this information is allowed to hide the details of the domain, since the full internal topology is the private information of the operator, who tries not to share its confidential information, furthermore, most of the internal topology information is unnecessary for inter-domain routing. Thus,

![Figure 1.6: ASON multidomain architecture and interfaces](image-url)
operators share a virtual topology about their domain, containing aggregated routing and connectivity information.

The main questions are, what should such an aggregated topology information contain; and, based on that information, how is it possible to provision QoS-aware – i.e., protected – routes. There are many solutions on aggregation. The simplest solution is to hide all the internal topology and state of the domain. In this case, at the inter-domain level, the domain will be represented by a single node. This kind of aggregation is used in Private Network-Node Interface (PNNI) [47]. In our research we have been using aggregation schemes that provide more detailed information of the internal topology of the domain [7][8][9] in optical networks.

Related researches have been carried out by UPC (Universidad Politécnica de Cataluña). They faced the problem of multidomain routing primarily from a practical, protocol oriented point of view, and proposed general [48][49] and optical specific [50] solutions. In Chapt. 3 we also examine the problem of multidomain resilience and propose a solution based on p-cycles (Sect. 1.3).

1.2 Quality-assured routing

The end-users of the networks request for connections that have a definite quality. The Quality of Service (QoS) [51][52] parameters of the connection are defined in the contract, Service Level Agreement (SLA) made between the service provider and the end-user.

Besides the provided bandwidth, the transmission delay and the bit error ratio, the connection availability is one of the most important QoS attributes of a connection.

In the SLA the minimally expected availability is described usually as the allowed daily, weekly and/or yearly outage (e.g., seconds/day, minutes/week, hours/year). To provide the desired grade of availability [53], the operator of the network has to estimate the availability of the connections in advance and, if necessary, some resilience mechanism (protection or restoration) must be provided for the connection in order to enhance its availability. Applying different resilience schemes, however, also requires such a network architecture (physical equipment, controllers, routing/restoration protocols, etc.) that supports the selected scheme.

In the rest of this section we will give a short overview about the link availability estimation methods (Sect. 1.2.1), about the resilience schemes (Sect. 1.2.2) and how the availability of a protected connection can be estimated (Sect. 1.2.3).
1.2.1 Estimating the link availability metric

To estimate the availability of the optical cables we take two basic invariants, namely \( MTTR \) (Mean Time To Repair [h]) and \( CC \) (Cable-Cuts [km] – the average cable length suffering 1 cut a year). Having a cable of length \( l \) [km], its \( MTBF \) (Mean Time Between Failures) value is given in hours [h] as

\[
MTBF = \frac{CC \cdot 365 \cdot 24}{l},
\]

(1.1)

next, the unavailability ratio is calculated as

\[
U = \frac{MTTR}{MTBF} = \frac{MTTR}{CC \cdot 365 \cdot 24} \cdot l,
\]

(1.2)

and finally, the availability metric \( A \) of a network element we get as the complementary of unavailability: \( A = 1 - U \).

In our simulations (Sect. 3.2) we use a special invariant, the Link Failure Coefficient [km\(^{-1}\)] defined as

\[
LFC = \frac{MTTR}{CC \cdot 365 \cdot 24h}.
\]

(1.3)

It takes its value from the range \([3 \cdot 10^{-7}, 3 \cdot 10^{-4}]\) which range covers the optimistic, nominal and conservative values of \( MTTR \) and \( CC \) listed in [54].

1.2.2 Resilience schemes

Network failures can be handled in several ways. Common of the failure handling strategies is that they intend to provide a backup path for the traffic that is going through the broken/failed part (either cable or switching equipment) of the network. The difference of various resilience mechanisms, as partially presented in [55], can be viewed from many aspects and classified along multiple dimensions:

1. the responsibility of activation:
   
   - do the end nodes of the connection (or some immediate nodes along the connection) need to be notified about the failure event to activate the protection (connection oriented),
   - or is the backup path activation performed locally?

2. the scope:
   
   - is the whole working path protected by an alternate, disjoint backup path (end-to-end protection),
   - are there segments of the working path (i.e., two or more spans) that are protected separately[56][57],

18
• or the *spans* between two adjacent nodes (e.g., *p*-cycles [58] or RPR (Resilient Packet Ring) [59]) are the subject of the protection?

3. the manner (timing) of backup path planning:
   - is the backup path defined in advance (*protection*) or
   - after the failure is detected (*restoration*)?

4. the shareability and the activation of the backup resources (only in case of protection):
   - 1+1 dedicated protection, meaning that the data is sent simultaneously on the default and the backup channel, and in case of failure on the default path, the receiver switches to listen to the backup channel;
   - 1:1 or, generalized, N:M protection, when all the default and backup paths are allocated in advance, however, the backup channels carry traffic only in case of failure;
   - shared protection, when the backup channels are allocated only in case of failures on the default path. Otherwise they are not allocated, thus these resources may be used for protection in shared manner.

5. the quality (does the backup path provide the same service level as the original one, what is the protection preemption strategy of the scheme, etc.) and the quantity of the backup connection:
   - is the *whole* amount of the traffic protected or
   - only a certain *ratio* of it, e.g., the 2:1 protection can protect only 50% of the traffic, if both of the default paths fail.

There may be other aspects taken into consideration as well.

Note that the first two dimensions are not independent as span and segment protections are usually handled locally whereas path protections are controlled by the connection ends.

### 1.2.3 Metrics of connection availability

Whilst the most works in the field of optimization and algorithm development have focused on achieving near-optimal (i.e., minimal) resource usage, the availability, or more precisely said the availability evaluation, has been a marginal issue of the optimization methods.

To assure the availability of a connection, many different methods can be used. The easiest way is to prove *single failure survivability*. The formerly mentioned protection schemes protect against single failures.
However, as the networks grow and the applications demand higher availability, assuring single failure survivability is not enough to decide the fulfillment of connection availability requirements. The straightforward extension of single failure survivability is to define dual-failure survivability, etc. In the literature this trend is referred to as \( n \)-failure restorability, and the quality metric for the connection / network is given as \( R_1, R_2, R_3 \), etc. defining the ratio, how many connections can survive 1, 2, 3... failures [60].

If we want to take into account the fact that different network equipment have different availability, more complex methods are needed which count with multiple simultaneous failures. These methods derive connection availability from link availabilities. There are two basic cases of accumulated availabilities of multiple links:

- **ALL** the links have to be available, e.g., connecting links in series, resulting in accumulated availability of

  \[
  \overline{A} = \prod_i A_i; 
  \]

- **ANY** the links have to be available, e.g., connecting links in parallel, resulting in accumulated availability of

  \[
  \overline{A} = 1 - \prod_i (1 - A_i). 
  \]

This manner, the availability of the dedicated 1+1 end-to-end protected connection can be evaluated by modeling the connection as parallel switched series of devices [61]. However, if the resources are shared, e.g., in SBPP (Shared Backup Path Protection), the availability modeling becomes complex. Either extensions have to be applied to this Serial-Parallel (S-P) method or other methods (heuristics) must be used. The authors of [62] trace back connection availability on link availabilities in networks without resource sharing, [63] suggests a method for estimating SBPP availability, [64] estimates connection availability by evaluating each combination of \( N \) failures in the network and giving an upper-bound for the estimation deviation, [65] investigates dual failures and [66] analyzes multiple failures in \( p \)-cycles. Another possibility is using heuristics, e.g., Monte Carlo simulation, Tabu Search, stratified sampling [67] etc. for estimating connection availability.

Although the \( p \)-cycle protection scheme, discussed in Sect. 1.3, is a kind of shared protection, provides a special way resource sharing. By exploiting this speciality, the accurate link availability based availability for \( p \)-cycle-protected connections can be evaluated relatively fast. The method for that evaluation will be presented in Sect. 4.2.
1.3 \( p \)-Cycle Protection

The \( p \)-cycle protection scheme combines the advantages of rings and mesh: it realizes ring-like speed while retaining the capacity efficiency of the mesh-based methods [58].

The \( p \)-cycle is a cyclic, pre-calculated, pre-assigned, closed path of spare capacity. It provides protection for any link that has both end nodes on the cycle (Fig. 1.7). These links are either on-cycle or straddling links. In the case of an on-cycle link failure, the protection path, similar to the BLSR (Bi-directional Line Switched Ring), passes on the remaining part of the cycle (Fig. 1.7(a)). In contrast to BLSR rings, however, \( p \)-cycles are able to protect straddling links. This feature results in better capacity-efficiency of a \( p \)-cycle than a similarly sized ring [68]. In the case of a straddling link failure, the \( p \)-cycle can protect two units of capacity by providing two alternate backup paths around the cycle, as Fig. 1.7(b) and 1.7(c) shows.

The \( p \)-cycle can also protect against node failures, and more generally it can protect any working path segment that has its both ends on the cycle [69].

Further, detailed description about \( p \)-cycles is given in [70].

1.3.1 Resource efficiency of \( p \)-cycles

One of the most important advantages of \( p \)-cycles is that their relative capacity requirement is low. This kind of resource efficiency basically depends on two factors: first, among the set of candidate cycles, we have to prefer those cycles, which can share resources efficiently (a priori efficiency), second, the protection has to be assigned to the connections in an optimal (or nearly optimal) way.

The a priori efficiency of \( p \)-cycle \( p \) (\( AE(p) \)) [71] is defined as the ratio of the total amount of working capacity that \( p \) can protect and the cost of \( p \). Denoting the on-cycle spans by \( S_{C,p} \) and the straddling spans by \( S_{S,p} \) and
assuming that the cost of 1 unit of capacity is 1 on each span, we get that

\[ AE(p) = \frac{2 \cdot |S_{Sp}| + |S_{Cp}|}{|S_{Cp}|}, \quad (1.4) \]

since – as mentioned before – the \( p \)-cycle can protect two units of capacity on each straddling span, where one unit on the on-cycle spans.

If we cannot assume that the cost of capacity is uniform in the network, we have to use the following formula instead of Eq. (1.4):

\[ AE(p) = \frac{2 \cdot |S_{Sp}| + |S_{Cp}|}{\sum_{e \in S_{Cp}} c_e}, \quad (1.5) \]

where \( c_e \) is the cost of 1 unit of capacity on span \( e \).

This definition implies that \( p \)-cycles with more straddling spans (relative to on-cycle spans) are more effective than those which have fewer straddling spans. Hence, the candidate cycle search algorithms (see Sect. 1.3.4) aim to find in the network cycles with many straddling spans.

The objective of \( p \)-cycle protection assignment is usually to minimize total cost of spare capacity usage while providing protection for each working capacity. We denote the amount of working capacity on span \( e \) by \( w_e \), the amount of spare capacity assigned to \( p \)-cycles by \( s_e \) and the total amount of working capacity on span \( e \) that can be protected with the assigned \( p \)-cycles, by \( b_e \). Now we can define the efficiency ratio [72] as

\[ r_{eff} = \frac{\sum_{e} s_e}{\sum_{e} w_e} \quad (1.6) \]

This ratio can be kept low, if we employ \( p \)-cycles with high a priori efficiency. However, usually we cannot guarantee that the protection covers the working capacity exactly, thus \( b_e \) will be higher than \( w_e \) for some spans, meaning that there is more protection assigned to a span than needed. This overprotection deteriorates the capacity efficiency resulting in higher efficiency ratio.

### 1.3.2 \( p \)-Cycles and multiple failures

Since two simultaneous failures of on-cycle spans make the protection of both affected spans impossible, the \( p \)-cycle is suited for a single on-cycle failure only. That implies that the on-cycle spans can share their protection path without worrying about concurrent resource usage requests and failure handling priorities. That way, on the one hand, there is resource sharing, which makes resource usage efficient, on the other hand, there is no competition on the shared resources, since whenever – due to multiple on-cycle
failures – there would be more candidates requesting the resources, none of them could setup the required protection path.

From failure handling perspective, however, straddling spans are not that simple. In the case of two straddling span failures, opposed to the dual on-cycle span failures, (at least) one of the two straddling span failures could be protected by the still active on-cycle spans. According to this, in \( p \)-cycles, it is worthwhile dealing with multiple failures. The two basic approaches are First Come – First Served (FCFS) and Priority-based failure handling. Regarding the FCFS approach, [73] enumerates and analyzes the possible dual-failure sequences. The main drawback of the FCFS approach is that arranging simultaneous failures into a sequence requires a dynamic model also dealing with setup, notification and other controlling delays, which leads to further complexity, and that is out of the scope of our study. When examining the connection availability and the performance of \( p \)-cycles, in Chapter 4, we will consider only priority-based failure handling.

1.3.3 Failure handling priority

Priority-based failure handling means that in the case of simultaneous on-cycle and straddling span failures the behavior of the \( p \)-cycle is pre-defined regardless of the failure occurrence order. On straddling spans two units of working capacity can be protected and if there is only one unit of capacity to protect, only the half part of the cycle is needed (to be available) for providing a backup path. On-cycle spans, however, do not require any straddling span to be operational for protection.

Since both possibilities, i.e., giving priority either to on-cycle spans or to straddling spans, have advantages, we have defined more strategies for failure handling priorities. Figures 1.8,1.9,1.10 and 1.11 demonstrate which spans must be operational to protect one unit of capacity of an on-cycle span and of a straddling span, respectively.

Spans marked with an ‘X’ sign are the failed spans. Spans in bold denote those spans which must be operational as they provide the backup path for the failed span. Those spans which are overwritten with an ‘!’ mark, are not involved in forming the backup path, still they are required to be available and the spans with a ‘?’ mark over them are in don’t care state: at current cycle configuration they may even fail, but in case they fail, they are not protected. The 4 presented strategies are the following:

**Strategy I:** All spans must be available except the failed one (Fig. 1.8).

This simple strategy does not handle priorities and the protection will fail in any case of two or more simultaneous failures of both on-cycle and straddling spans.

**Strategy II:** Protect on-cycle spans first (Fig. 1.9).
This strategy gives priority to the on-cycle spans against the straddling spans. Furthermore, there is no priority defined among straddling spans: the $p$-cycle protection of a straddling span will fail in the case of simultaneous straddling span failures.

**Strategy III:** Protect straddling spans first, with priority handling (Fig. 1.10).

In this case, the straddling spans have higher protection priority than the on-cycle spans, however, as all the on-cycle spans must be operational whenever a straddling span fails, we do not gain much in availability compared to **Strategy I** except for the fact that some straddling spans (those with lower priority) may fail whenever we protect a high-priority straddling span.
**Strategy IV:** Protect straddling spans first, with priority handling, non-backup-path on-cycle spans may fail (Fig. 1.11).

![Diagram](image)

Figure 1.11: \(p\)-Cycles allowing non-backup-path spans to fail

This strategy requires the least spans operational in case of straddling span failures. But it also penalizes on-cycle failures: each span except for the failed one must be up to provide protection against on-cycle failures. Dual-failure survivable “Platinum” class connections [73, 74] would prefer this type of failure priority handling since they are conducted on straddling spans.

Note, that there may be other strategies defined as well, whilst even Strategy IV does not allow concurrent straddling span failure protection even if resources for backup paths are available.

### 1.3.4 Selecting candidate \(p\)-cycles in the network

If we want to apply \(p\)-cycle protection in a network, first, we have to collect a set of potential cycles. There are multiple objectives that can be considered when defining such a set:

- **The size of the cycle set should be small.** Basically, there are two reasons in the background of this objective: First, the enumerated cycles need to be stored in a database, and a larger set of cycles requires more storage space. Second, the algorithms that assign \(p\)-cycle protection to connections (e.g., CIDA or an ILP solution) run faster if they have to work with a reduced set of candidate cycles.

- **The cycles should cover the spans within the network in a diversified way.** This objective aims at the efficient resource usage. Having more candidate cycles available we have more choices in protection assignment, hence we can find a protection layout that is closer to the optimal solution.

- **The cycles should be short.** Controlling short cycles is simpler, furthermore, in short cycles the probability of multiple failures is low.
The cycles should contain multiple straddling spans. Since the a priori efficiency of the cycles depends on the relative number of straddling spans (see Sect. 1.3.1).

The algorithm that produces the set of candidate cycles should be simple and fast. This is a general expectation against algorithms. However, in our case – as the candidate cycle search procedure is performed in advance – this objective is of less importance.

It is obvious that some of the objectives are in conflict, and it is clear that we cannot find a single algorithm that fulfills all our expectations. Therefore, there were multiple candidate $p$-cycle searching algorithms proposed resulting in different cycle sets. The most important among them are the SLA [75], Expand and Grow algorithms [76], however, enumerating each cycle in the network is also a viable solution.

The SLA (Straddling Link Algorithm) tries to find for each span two disjoint bypass paths. After merging these two bypassing paths we get a cycle, in which the bypassed span is a straddling link.

The Expand algorithm takes the result of the SLA algorithm as an initial set, and for each cycle it tries to substitute each on-cycle span with a bypass path. Thus the original on-cycle spans will become straddling spans in the new cycle.

The Grow algorithm differs from the Expand in that sense, that after finding a bypass path for an on-cycle span, instead of processing the next on-cycle span, the algorithm begins to iterate through the spans of the bypass path, searching for further on-cycle span $\rightarrow$ bypass path replacement.

Common of the briefly presented algorithms is that they use Dijkstra’s shortest path algorithm [77] when searching for a bypass path. Even if there are multiple bypasses for a span, they will find only the shortest one, and in usually they will not enumerate all the cycles in the network. If we want to define a candidate $p$-cycle set that contains all the cycles in the network, we can apply exhaustive search algorithm, e.g., [78].

1.3.5 $p$-Cycle assignment to protect connections

Having a candidate cycle set available, the next step of the $p$-cycle protection procedure is to assign the cycles to protect working capacity. Finding the optimal (most resource efficient) protection layout is a hard problem. However, we can formulate the problem as an Integer Linear Programming problem and obtain the optimal solution by ILP. There are proposals on how to find the optimum in case the working paths of the connections are predefined [72] both in WS and WI optical networks. Joint optimization of working paths and protection is an even more complex problem, but can lead to less total capacity requirement.
There are also heuristics available that can design the protection layout much faster than the ILP does. In the simulations in Chapt. 3 we use the Capacitated Iterative Design Algorithm (CIDA) [71] which we shortly present here.

The CIDA algorithm examines in each iteration the unprotected working capacities ($w'_e$, initialized by $w'_e = w_e$) in the network. For each candidate cycle its actual efficiency is tested. The definition of the actual efficiency – corresponding to the a priori efficiency – is

$$E_w(p) = \frac{\sum_{e \in S_{S,p}} 2 \cdot w_e + \sum_{e \in S_{C,p}} w'_e}{\sum_{e \in S_{C,p}} c_e}.$$  \hspace{1cm} (1.7)

After these tests are performed, the most efficient cycle (cycle $p$ with the highest $E_w(p)$ value) is added to the protection, and the values of unprotected working capacities are updated. This iterative step is repeated until there are no more unprotected working capacities in the network.

The calculation complexity of CIDA is proportional to the accumulated number of the on-cycle and straddling spans of all the cycles in the candidate cycle set and to the amount of total working capacity.
Chapter 2

Enhancing topology advertisement

We examine the possibilities of making the topology advertisement efficient in this chapter. The scope of the examination is limited to the single domain networks. Multidomain networks and inter-domain topology advertisement are presented in papers [7][8][9]. The common property of methods presented is that in contrast to different triggering policies, they try to preserve the accuracy of the advertised network state even with a reduced information advertisement. The key idea is to analyze and to exploit the regularity of the internal switching logic of optical devices.

First, a wavelength graph simplification method is presented (Sect. 2.1). This method reduces the number of edges in the WG while preserving the value of additive metrics between the ingress and egress nodes. Next, in Sect. 2.2, the so-called Rule-Based Topology Advertisement (RBTA) is proposed for advertising the link state changes in a compact manner. Finally, the Rule-Based Topology Advertisement is extended to support different protection schemes (Sect. 2.4). However, to support different protection schemes, we need to be able to handle arbitrary protection schemes in a uniform way in our environment. For this purpose the Generalized Protection Formula is introduced (Sect. 2.3).

2.1 Topology aggregation on additive link metrics

In the RWA problem, while constructing the Wavelength Graph, we realize that there is a kind of regularity in the topology (subgraphs) representing the OXC network devices: they are paired (sub)graphs consisting of ingress and egress vertices and their edges, connecting these vertices, can be bounded. For example, in a simple OXC without wavelength conversion capability, at each wavelength the ingress and egress vertices are connected in a complete bipartite graph ($K_{n,n}$ in case of an OXC with $n$ ports). If we can assume
that the only valuable metric that the edges carry is a simple additive metric (e.g., virtual reservation cost), and this metric is uniform in the subgraph, then by adding virtual nodes to the subgraph we can reduce the number of edges, and thus the number of links to advertise, as Fig. 2.1 shows.

In Fig. 2.1(a) we can see the original subgraph, having cost = 50 everywhere on its edges. The shown OXC has 4 ports and employs 3 wavelengths. This results in $3 \cdot 4 \cdot 4 = 48$ edges, which is the double of edges in the aggregated subgraph (Fig. 2.1(b)).

Based on this concept, we have proposed aggregations on numerous OXCs [9] in the first phase of our WDM topology advertisement research. The concept was then refined, extended to min/max metrics and allowing constrained inaccuracy for the sake of better aggregation. The master work of Zoltán Török summarizes the results and also proposes algorithms for aggregation [79].

Although the described aggregation model is a significant achievement, it suffers from certain shortcomings. For example, whenever the metrics on some edges has been changed, it breaks the regularity of the graph, where each edge had the same metric. This leads to deteriorated performance of aggregation. Another problem is that this kind of graph aggregation works only for a single edge metric, and even though it can be applied in inter-domain environment, still it is not a general solution. Furthermore, when protection is required, we face serious modeling problems to solve [80].

Therefore, we have proposed another general and at the same time efficient way of topology advertisement in WDM networks. The following sections will present the results of this research.

2.2 Rule-Based Topology Advertisement (RBTA)

Although by applying the formerly described solutions we can reduce significantly the controlling overhead generated by the topology advertisement,
the most remarkable reduction can be achieved if we recognize that the link state changes of an optical device are not independent. Exploiting the fact that there is a certain correlation within the LSA messages, we can base the topology advertisement on rules describing this correlation of link state changes, and this way we can avoid the exchange of redundant information.

2.2.1 Problem of Wavelength Graph representation

In Sect. 1.1.5 we have seen what are the benefits of the WG representation of the network and how does the WG representation support the RWA process. We have also seen that the WG does not scale well as the number of wavelengths increases. Moreover, unlike the basic graph representation of the network, in the WG representation the basic event of allocating capacity on a single link may influence the state of other links as well. We can find the reason for this behavior in the architecture of the cross-connects and in the fact that graphs cannot express this architecture in a compact manner.

![Figure 2.2: Cross-connect state: a) initially, b) after switching i2 to o1](image)

![Figure 2.3: Wavelength graph state: a) initially, b) after switching i2 to o1](image)

Figures 2.2 and 2.3 illustrate this problem showing a cross-connect that has 4 input and 4 output ports, all the ports connecting channels of capacity of 192 bandwidth units. For better understanding the figures show only a single wavelength. Figure 2.2 shows two states of the 4 × 4 cross-connect: an initial one and another one after provisioning a channel of 12 bandwidth units from i2 to o1. Figure 2.3.a and 2.3.b show the subgraphs of the cross-connect in the WG that correspond to Fig. 2.2.a and b, respectively.

Initially all the 16 switches are in the same state: either of them is allowed to be switched on. The WG equivalent for this initial state is, as shown in
Fig. 2.3.a, that the 4 vertices at the left hand side, that represent the input ports, are fully connected to the 4 output ports by edges with a capacity of 192 bandwidth units. Evidently, after the allocation we still have 180 out of 192 free bandwidth units between \( i_2 \) and \( o_1 \). This is denoted by the bold line in Fig. 2.3.b. However, the cross-connect at the optical layer cannot split the wavelength channel and as \( i_2 \) and \( o_1 \) become connected, none of the other ports are allowed to connect to these two ports (Fig. 2.2.b). As a consequence, those switching possibilities that the cross-connect does not allow will appear in the WG as edges without any free capacity (marked with dotted lines in Fig. 2.3.b).

By this example we intended to highlight the basic problem of the WG representation of optical networks: provisioning resources along the selected path may affect also edges that are not part of the path. In terms of topology advertisement this means multiple LSA messages for provisioning a single resource inside a network node. Further problem is that these additionally affected edges suffer a state change (capacity reduction) of 100\% (the free capacity metric becomes 0 on these edges). This may explain why triggering policies show a relatively bad performance when applied in WDM networks using conventional routing algorithms.

However, the set of additionally affected edges and the originally provisioned edge are related, i.e., the LSA messages to be advertised have a certain redundancy. This redundancy can be described by *switching rules*. The principle of RBTA is to reduce the topology advertisement overhead based on these switching rules.

### 2.2.2 Principle of RBTA

Conventionally, the link state advertisement (LSA) procedure, that takes place between the Data Provider and the TED (see Sect. 1.1.6), advertises the state of the WG edges. Each LSA message holds information about a single WG edge. If we have multiple WG edge state changes, multiple LSA messages will appear on the control channel. As discussed before, within an OXC, each time a connection is set up or released, there may be multiple WG edges suffering state change, hence multiple LSA messages are needed to describe these state changes.

The OCC acts as Data Provider whenever resource allocation or deallocation happens to the controlled OXC due to connection setup or release. Conventionally, in these cases the OCC takes three types of information to calculate the modified state of the WG subgraph of the controlled OXC. These sources are:

1. the original state of the WG subgraph before the state change (what should be modified?),
2. the switching rules describing which edges become unreachable or configurable after a SETUP or RELEASE call (how should be modified?),

3. the received signaling message about resource allocation or release (what event triggers the modification?).

If the logic deciding edge reachability (2nd type of information) is known by the PCE, and the PCE is aware of the initial network topology, then the PCE itself can maintain its TED merely by receiving information about resource allocation/deallocation. In this case the PCE does not need to get explicit link state information from the Data Providers. This is the basic idea of the RBTA. This way, for each modified OXC, the whole LSA procedure can be replaced by a simple notification sent to the PCE. For compatibility reason, however, we encapsulate this notification into a single LSA message.

2.2.3 Description of the messages

In the Rule-Based Topology Advertisement we separate static and dynamic topology information: The adjacency of the vertices of the WG is invariant, it is static information, whereas the attributes of the edges of the WG change dynamically due to resource allocation/deallocation.

Static data

Static invariant data, i.e., the topology and the switching rules – as they are coherent – can be advertised together and initially. The static data consists of blocks. Each block describes a port, represented by a vertex in the WG, and its attributes. The attributes of the WG vertex are the TE attributes of the represented port, the identifiers of incoming and outgoing links, and the following additional information. For both groups of incoming and of outgoing links a discrete value \( k \) \((0 < k \leq d; d \text{ (nodal degree) is the size of the set)}\) denotes how many links of the group can be switched simultaneously. Note that usually only \( k = 1 \) (a single link is allowed to be switched, e.g., simple OXC, discussed in Sect. 2.2.1) or \( k = d \) (no switching restriction is given) are defined, but other \( k \) values may also be defined.

Formally, the attributes of the static data are the following:

- \text{id} identifies the port.

- \text{TE\_attr} is the compound of various TE attributes. For example, \text{TE\_attr.total\_cap} tells the amount of free capacity of the port. Note that this data is basically dynamic information, still we include it in the set of static information for initialization purposes.

- \text{in\_links} is the set of link identifiers that are connected to the given port at the ingress side (traffic may come from these links).
\texttt{in\_max} is the maximal number of links simultaneously switched at the ingress of the port.

\texttt{out\_links} is the set of link identifiers that are connected to the given port at the egress side (traffic may go to these links).

\texttt{out\_max} is the maximal number of links simultaneously switched at the egress of the port.

In this message specification we exploit the fact that the TE attributes of a link are determined by the ports at its endings. Defining the switching rules that way, the topology of the network, i.e., the adjacency of the vertices, is also given (see Sect. 2.2.4), as the incidence list of the WG can be derived from the \texttt{out\_links} sets (or from the \texttt{in\_links} sets).

Note that the RBTA assumes that the topology of the network is invariant. Topology changes (e.g., installation of a new OXC in the network) are not handled by the RBTA internally. This implies, that in case of topology change some kind of \textit{re-initialization} is needed. This re-initialization may be realized by advertising the static RBTA data (partially or entirely) whenever the topology of the network is modified.

\textbf{Dynamic data}

The simplest realization of the dynamic data, i.e., the link state change notification, is an LSA message identifying the concerned link and denoting the amount of the capacity change. The TED evaluates the actual sum of the allocated bandwidth for each link ($\texttt{sum\_cap}[e]$) and if it equals 0, the PCE assumes that the given link is not switched, otherwise the link is considered as being switched. In the latter case, the number of simultaneously switched links must be checked in the ports at the head and at the tail of the link to decide whether other links connected to the port are allowed to be switched or not (see the definition of the \texttt{free\_cap}(e) function in Sect. 2.2.4).

\textbf{2.2.4 Constructing the WG}

The algorithm that constructs the WG topology from the static data is simple. First, for each data block a vertex (identified by the \texttt{id} attribute of the data block) must be added to the WG. We also process the sets \texttt{in\_links} and \texttt{out\_links} to define the invariant associative arrays of

- \texttt{head}[\textit{edge}] as the head node of the edge,
- \texttt{tail}[\textit{edge}] as the tail node of the edge,
- \texttt{in\_edge}[\textit{node}] as the set of ingress edges,
- \texttt{out\_edge}[\textit{node}] as the set of egress edges.
Next, as the head and the tail of each edge are collected in the arrays head and tail we add each edge to the WG. Finally, we derive from the static data the invariant associative array of

\[ \text{total_cap}[\text{edge}] \]

as the minimum of \text{TE.attr.total_cap} of the two end nodes of the edge,

and initialize the variable array of

\[ \text{sum_cap}[\text{edge}] \]

for each edge \[\text{sum_cap}[\text{edge}] = 0\] denoting that initially there are no resources allocated in the network.

Based on these arrays we define the \text{free_cap(e)} function to determine the free capacity (and reachability) of link \(e\). The pseudo-code of the function is as follows:

\begin{verbatim}
function free_cap(e)
    if (sum_cap[e] > 0) // already switched
        return total_cap[e] - sum_cap[e]
    out_n=0
    in_n=0
    for_each (i in out_edge[head[e]])
        if (i != e and sum_cap[i] > 0)
            out_n++
    for_each (i in in_edge[tail[e]])
        if (i != e and sum_cap[i] > 0)
            in_n++
    if (out_max[head[e]] <= out_n or
        in_max[tail[e]] <= in_n)
        return 0
    return total_cap[e]
\end{verbatim}

Note that the first two lines of the function calculate the free capacity of the link, whereas the remaining part of the function is responsible to determine whether the given link is allowed to be switched (is reachable) or not. Unreachability of link \(e\) is denoted by the value \text{free_cap(e)} = 0. Thus, the routing algorithms will avoid these unreachable links (due to not enough capacity).

The complexity of the function is discussed – along with the performance of the RBTA model – in Sect. 2.5.4.

2.3 Generalized Protection Formula (GPF)

Before introducing the protection-aware extension of RBTA (RBTA-P) we give a generalized description for protections.
The goal of Generalized Protection Formula (GPF) is to have a general formula that is capable of describing all the protection schemes in a unified manner. The GPF uses ideas of Partial Path Protection (PPP) [81], and of sub-graph routing [82][83] and provides a flexible, compact way of description.

2.3.1 Scope of GPF

We gave a classification of the resilience schemes in Sect. 1.2.2. As its name suggests, with GPF we can formulate only protection schemes, still except for the dimension of the backup path planning manner it covers all other dimensions of the domain of resilience schemes.

The protection dimensions of scope and shareability are directly supported by GPF, i.e., we can describe 1+1, 1:1 and shared end-to-end, segment and link protections, whereas other dimensions of protection (e.g., quality, multifaile restorability) can be transformed and eliminated (see Sect. 2.3.2). Finally, for the responsibility of activation, RBTA-P proposes even a new failure handling paradigm in Sect. 2.4.4.

2.3.2 Transforming the protection schemes to fit into GPF

There are some dimensions of the protection schemes (the dimensions are enumerated in Sect. 1.2.2) which need to be transformed to fit into the GPF, since the GPF originally can describe neither multifaile survivable protection schemes nor protection schemes that do not provide protection for the whole amount of the traffic.

Partial traffic protection

The amount of protection is a dimension, which can be eliminated in the following manner: if the protection path can carry only $x\%$ ($x < 100$) of the traffic, two traffic flows are routed instead of one: the one carries $x\%$ of the original traffic with protection, the other one carries the remaining part $(100 - x\%)$ of the traffic without any protection.

Multifailure survivability

Originally, the GPF describes one-failure-survivable schemes. However, the failure survivability, expressing how many network devices can simultaneously fail without breaking the connection, is also a dimension that can be transformed into single-failure-survivability and other constraints: By introducing the concept of Shared Risk (Link) Group (SRLG, SRG) (Sect. 1.1.5), a set of network devices that seem to be independent may form a risk group. This way node or dual link failure scenarios can be transformed into single SRG failure scenarios. And these single failure scenarios already
can be expressed in GPF. Note, however, that in case of \( n \) network devices there are \( n \cdot (n-1)/2 \) pairs and in case of forming triplets – to assure triple-failure-survivability – there are altogether \( n \cdot (n-1) \cdot (n-2)/6 \) triplets. This is the price we have to pay if we want to describe multifailure survivable protection schemes by means of GPF.

### 2.3.3 Definition of GPF

The network is represented by a graph \( G(V, E) \). \( C \) is the set of connections in the network. \( F \) is the set of SRGs for those risks that are considered to cause failure.

The GPF describes from which ingress edge is the traffic switched to which egress edge. It describes this for each connection for each considered failure scenario and for each node. To give such a description of the network state, however, some of the given basic sets (\( V, E, C \) and \( F \)) have to be extended.

First, we have to extend the set of edges (\( E \)). We want to describe the switching state of a node by means of the set of edges. If a connection goes through a port, it is switched from an ingress edge onto an egress edge of the node representing the given port. In this case the switching of the given node can be described by the pair \((e^{in}, e^{out})\), so that \( e^{in}, e^{out} \in E \) are denoting the switched ingress and egress edges, respectively. However, we also have to describe the switching state of those ports that the connection avoids. In these cases the ports remain unconfigured, i.e., it is not defined from which edge and to which edge do the ports switch. To support these not defined switchings we introduce a special virtual edge, the indefinite edge \((e^{u})\). To cover each switching possibility of the nodes, (including the not defined switching), the GPF uses the extended \( E^{*} \) set instead of \( E \):

\[
e^{in}, e^{out} \in E^{*} = E \cup \{e^{u}\}.
\]

Using the extended set \( E^{*} \) we can describe not only the configured ports but also the unconfigured ports (they switch \( e^{a} \) to \( e^{u} \)) of the connection.

Second, we have to extend the set of considered SRGs (\( F \)) to obtain the set of failure scenarios (\( F^{*} \)). The members of \( F \) cover all the possible failure states that may happen to the network. However, the network also may be in the state of no failures at all. This \( f_{0} \) state can be denoted also by an SRG. This SRG is special link group since is does not contain any link (\( SRG_{0} = \emptyset \)). The extension of \( F \) is

\[
F^{*} = F \cup \{f_{0}\}, \text{ where} \quad f_{0} = SRG_{0} = \emptyset.
\]

Assuming that \( c \in C, n \in V \) and \( f \in F^{*} \), the GPF definition of a network configuration is given by the function:

\[
GPF(c, n, f) = (e^{in}, e^{out})
\]
realized in the dimensions of
\[ C \times V \times F^* \rightarrow E^* \times E^*. \] (2.5)

The formula Eq. (2.4) expresses that for each connection \( c \) in each node \( n \) for each failure (and no failure) state \( f \) there must be an input \( (e^{in}) \) and an output \( (e^{out}) \) edge assigned.

The concept of sub-graph routing [83] serves the idea that there may be multiple protections for a default path, and the protections should be assigned to failures. The network configuration in case of failure \( f_j \in F^* \) is given by the restriction
\[
GPF_j^F(c, n) = GPF_{\mathcal{C} \times V \times f_j}(c, n, f) = GPF(c, n, f_j) \] (2.6)

However, there are differences between the idea of sub-graph routing and the idea of GPF. The sub-graph routing defines a kind of shared protection that takes into account each failure scenario, whereas using the GPF we want to model different protection schemes assigned to different connections simultaneously. Using the GPF we can describe not only shared protection, but also dedicated or even cases when there is no protection, etc.

The GPF also covers the Partial Path Protection [81] allowing also, besides segment- and link-level protection, that some parts of the default path remain unprotected. Given the connection \( c_x \), its default and the protection paths (\( \text{Routes}_x \)) can be derived from the restriction
\[
\text{Routes}_x(n, f) = GPF_{|c_x \times V \times F^*}(c, n, f) = GPF(c_x, n, f) \] (2.7)

By applying further restriction onto \( \text{Routes}_x \) we get the node configuration for the “\( j \)th” backup path \( (BP_j) \) meaning that it is protecting against \( f_j \) along with the special case \( j = 0 \) defining the default path \( (DP = BP^0) \):
\[
DP_x(n) = \text{Routes}_x|_{V \times f_0}(n, f) = GPF(c_x, n, f_0) \] (2.8)
\[
BP^j_x(n) = \text{Routes}_x|_{V \times f_j}(n, f) = GPF(c_x, n, f_j). \] (2.9)

Note that the failures that do not affect the default path are not handled by any protection, thus in many \( f_j \) cases \( BP^j \) is identical to \( DP \).

After having given restrictions on the dimensions of failure states and connections, let us examine the third restriction of GPF concerning the dimension of nodes/ports \( (V) \). The function \( SwR_i \) describes the Switching Rules in node \( n_i \):
\[
SwR_i(c, f) = GPF_{|c \times n_i \times F^*}(c, n, f) = GPF(c, n_i, f). \] (2.10)
The codomain of $SwR_i(c,f)$ can be narrowed if we define the subsets $E_i^{in} \subseteq E$ as the ingress and $E_i^{out} \subseteq E$ as the egress edges of node $n_i$:

\[ e^{in} \in E_i^{in} \cup \{e^u\} \quad (2.11) \]

and

\[ e^{out} \in E_i^{out} \cup \{e^u\}. \quad (2.12) \]

Note that the formula is redundant as the switching of an egress edge at a node immediately defines the switching in the next node at the head of the edge. Thus either value of the pair $(e^{in}, e^{out})$ can be omitted, however, for better readability, we use both values.

### 2.3.4 Simple GPF example

Figure 2.4 shows a network with 4 nodes and 4 links represented as 4 vertices and 8 directed edges. Assuming single link failures each directed edge pair running between the vertices forms an SRG. The unidirectional connection $c_1$ is routed from $n_1$ to $n_4$ on the edges $e_{1,2}$ and $e_{2,4}$. This connection is partially protected: If $e_{2,4}$ fails, i.e., in case of $SRG_4$ failure a backup path is provided through $n_3$.

The GPF function for $c_1$ ($Routes_1(n,f)$) can be summarized in a matrix as shown in Table 2.1. The cell in row $n_i$, column $f_j$ contains the value of $Routes_1(n_i,f_j)$.
The first column (column of \( f^0 \)) describes the default edge-switching (\( DP_1 \)). As backup path is provided only in case of \( SRG_4 \) failure (\( BP_{41} \)), the other columns (\( SRG_1, SRG_2, SRG_3 \)) are identical to the first column (\( BP_{11} = BP_{21} = BP_{31} = DP_1 \)). The differences in \( BP_{41} \) are marked with boldface.

2.4 Topology description supporting protection (RBTA-P)

The mere RBTA concept, described in Sect. 2.2 does not have direct support for resilience. This means that failure-dependent switching, i.e., resource allocation or, more precisely said, network configurations that are failure-conditional (activated only in case of failures) cannot be expressed by means of RBTA. This way the link states cannot be described entirely in case of shared (link, segment or end-to-end path) protections and in case of the 1:1 dedicated ones. Of course, there are workarounds, e.g., all the shareability information has to be sent to the PCE (TED) by an external advertisement mechanism, independent of RBTA. Restoration and 1+1 kind of protections, as there are no conditional allocations, are feasible with RBTA. To sum up, although by means of RBTA the amount of transmitted and stored topology information can be dramatically reduced, and by applying external methods, also protection is supported, the lack of internal support of diverse resilience schemes is a serious shortcoming of the mere RBTA.

In the current section we aim to eliminate this shortcoming by proposing the Rule-Based Topology Advertisement and Maintenance with Protection (RBTA-P), an extension of RBTA to support various protection schemes.

Taking into account that RBTA was intended to be quite general to describe any kind of opto/electronic cross-connect, concerning the resilience, RBTA-P also tries to be as general as possible being based on the concept of Generalized Protection Formula.

Being an extension, the RBTA-P method inherits the most properties of the RBTA. That way our description of RBTA-P focuses only on the difference compared to RBTA.

2.4.1 Advertised topology information

The structure of the initially advertised invariant static data in RBTA and RBTA-P are identical. As the static data does not contain any allocation information, neither does it contain conditional allocation information that would require any special handling. Thus, the static data advertisement of RBTA is adopted by RBTA-P as-is.

The dynamic data must have a structure that is capable of expressing the network configuration changes initiated by the setup or release of a
connection protected by any kind of protection scheme. The default and backup paths of connection $c_x$ are described by the matrix $Routes_x (V \times F^* \rightarrow E^* \times E^*)$. Usually the connection does not pass through each switching point (node) in the network in each failure state, this way, the matrix will contain many $(e^u, e^u)$ pairs, expressing that the corresponding switching node in the corresponding failure state is unused. However, those members of the matrix which differ from $(e^u, e^u)$ denote altogether a configuration change which has to be advertised.

A straightforward extension of the RBTA dynamic message structure is that after identifying the edge and giving the amount of capacity change we enumerate the failure cases when the allocated capacity is used.

Most of the protection paths are defined for only a single (link-protection) or just a few (segment and end-to-end protection) SRG failures. All the other failure scenarios are covered by the default behavior. If the number of these failure scenarios is high, the LSA message that describes the default behavior (the default LSA) gets long, since all these failure scenarios have to be enumerated in the default LSA message. However, the length of the default LSA message can be reduced significantly, if, instead of enumerating all the corresponding failure scenarios, an Inverted SRG set is provided. This inverted SRG set enumerates the failure states when any protection path is active, and the whole set is marked as inverted. This set is as long as the number of failure states covered by the backup paths. Whenever receiving an LSA message with an inverted set, the TED will invert the received SRG set back to get the original SRG set.

Usually, the number of covered failure states is significantly less than the number of the considered failure scenarios in the network. That is why using the proposed SRG set inversion we can reduce the length of the default LSA message significantly.

### 2.4.2 Structure of the stored data

The TED stores only as much data as necessary for the path computation process. What information is needed to compute paths? Some of the paths, basically the default paths of the connections, are computed assuming that any protection of any other connection may be activated in the network, whereas other resources, mainly backup paths, are configured to be activated only in a given failure state or failure states.

Thus the TED must keep link state information for each failure state. That means that the $\text{sum\_cap}(e)$ association of the original RBTA must be extended and evaluated for each $f \in F^*$: $\text{sum\_cap}(e)[f]$ so that the $\text{free\_cap}(e,f)$ function can be retrieved for any $e \in E$ and any $f \in F^*$. 
2.4.3 Path computation

The construction of the WG used for path computation is inherited from RBTA as the initially advertised invariant static data in RBTA and RBTA-P are identical (see Sect. 2.4.1). Only the way of the free capacity (and the reachability information) calculation makes RBTA-P different from RBTA. Section 2.2.4 has already described an algorithm how to define this metric for an edge \( e \). This algorithm is extended now to \( \text{free\_cap}(e, f) \) that determines the actual free capacity for a given \( f \in F^* \). However, the default and the backup paths are configured not only for a single but also for multiple failure cases. If this set of failure cases is denoted by \( F_x \) (\( F_x \subseteq F^* \)) then the free capacity on link \( e \) in case of \( F_x \) is \( \text{Free\_Cap}(e, F_x) = \min_{f \in F_x} \text{free\_cap}(e, f) \).

2.4.4 Novel failure handling paradigm

Conventionally, the mechanism of activating shared backup protection is briefly is follows:

1. The failure is detected.
2. If the node that detects the failure is not the head-end of the protection path, failure notification message is sent to this head node.
3. The head node signals the activation of the backup resources (setup).

The release (deactivation) process happens similarly.

Now, in RBTA-P, taking advantage of the feature that the controllers are aware of the failure-dependent switching rules, Steps 2 and 3 can be replaced by a simple failure notification message \textit{flooded} to each controller. This failure handling mechanism provides faster failure recovery for shared protections with less control messages. The price of this improvement is the SRG failure identification, i.e., the detected link and node failures must be converted into SRG failures. This problem requires further research on the topic of \textit{failure localization} [84] and \textit{monitoring trail assignment} [85]. Numerical calculations that confirm the advantages of this failure notification mechanism are provided in Sect. 2.5.5.

Besides the control message reduction a further advantage of this failure handling paradigm, particularly in case of \textit{“in-band”} signaling (when a link at the Data Plane and the corresponding link at the Control Plane share the same risk) is that the information flooding is less failure sensitive against control plane failures, than the conventional backup path signaling.

2.5 Complexity and scalability of RBTA

We compare the performance of RBTA-P to the mere RBTA and to the conventional link state advertisement model. The performance analysis as-
sumes that the network topology is given, and the connections are routed on the same paths for each examined advertisement model. Basically, we have the following invariants:

\[ |V| \] is the number of vertices in the WG (i.e., the number of ports in the network),

\[ |E| \] is the number of edges in the WG,

\[ |F^*| \] is the number of failure cases, including no-failure case,

\[ |C| \] is the number of connections,

\[ |C_A| \] is the total number of connections served in the network. Note the difference between \( |C_A| \) and \( |C| \): \( |C| \) is the actual number of connections in the network, whereas \( |C_A| \) counts also the already released connections.

\( W_n \) is the length of the node identifier (in bytes),

\( W_e \) is the length of the edge identifier (in bytes),

\( W_a \) is the length of the TE attributes of an edge (in bytes),

\( W_f \) is the length of a failure identifier (in bytes),

\( d^{\text{max}} \) is the maximal number of node degree in the WG.

Additionally, we will use some derived invariants:

\[ \overline{d^{\text{in}}} = \overline{d^{\text{out}}} = \overline{d} \] is the average ingress, egress node degree:

\[ \overline{d} = \frac{|E|}{|V|}. \] (2.13)

\( \overline{wlen} \) is the average default path length. In case of RBTA-P we can derive it from the GPF function:

\[ \overline{wlen} = \frac{1}{|C|} \sum_{c \in C} \sum_{n \in V} I_{E^* \times E}(GPF(c, n, f_0)). \] (2.14)

\( \overline{blen} \) is the average length of backup paths:

\[ \overline{blen} = \frac{1}{|C| \cdot (|F^*| - 1)} \sum_{c \in C} \sum_{n \in V} \sum_{f \in F} I_{E^* \times E}(GPF(c, n, f)). \] (2.15)

Note that Eq. (2.15) counts to the backup paths also those cases when there is a failure somewhere in the network which does not affect the connection and the default path carries the traffic.
\(\overline{\text{clen}}\) is the average of the total number of edges that are affected by a connection:

\[
\overline{\text{clen}} = \frac{1}{|C|} \sum_{c \in C} \sum_{n \in V} \mathbf{1}(\exists f \in F^*: GPF(c, n, f) \in (E^* \times E))
\]  

(2.16)

\(\eta\) is the average port usage ratio:

\[
\eta = \frac{1}{|V|} \sum_{n \in V} \mathbf{1}(\exists f \in F^*, \exists c \in C : GPF(c, n, f) \neq (e^u, e^u))
\]  

(2.17)

In the following we define the total length of static \((M_{\text{stat}})\) and dynamic \((M_{\text{dyn}})\) link state messages (in bytes), the size of the database \((D)\) for storing the TED (in bytes) and the order of computational steps \((P)\) to calculate routes in the network.

Each metric is given for the conventional, original \((^0)\), for the RBTA \((^*)\) and for the RBTA-P \((^{**})\) control schemes.

### 2.5.1 Initial topology advertisement

In the conventional way, the WG is built by simple LSA messages whereas the RBTA uses special initial messages. In RBTA-P the WG is initialized by the same messages as in RBTA.

The conventional approach deals with LSA messages identified by a unique edge identifier, naming the nodes that the link connects and carrying TE attributes. That makes an LSA message length of \(W_e + 2 \cdot W_n + W_a\). This information is sent for each edge which results in total of

\[M_{\text{stat}}^0 = |E| \cdot (W_e + 2 \cdot W_n + W_a).
\]  

(2.18)

The initial messages of RBTA enumerate the ingress and the egress links (both of them are on average \(d\)) for each node. The TE attributes are assigned also to the nodes and there are two discrete values that denote the maximal simultaneous switching directions (as described in Sect 2.2.3). All these result in

\[
M_{\text{stat}}^* = M_{\text{stat}}^{**} = |V| \cdot (W_n + 2 \cdot d \cdot W_e + 2 + W_a).
\]  

(2.19)

The order of magnitude of the \(M_{\text{stat}}\) metrics are equal, since \(|E| = |V| \cdot d\).

### 2.5.2 Dynamic link state information

We can observe the most significant information reduction in the dynamic topology information exchange. Conventionally, in an optical switch, the resource allocation on a certain link has also affected the state of other
links (which became unreachable, see Fig. 2.3). The number of additionally affected links can be approximated by \(2 \cdot (d - 1)\) in an empty network. However, an operational network is not empty. In a maximally loaded network the number of additionally affected links is 0, but usually an operational network is not maximally loaded.

If this port usage ratio \((\eta)\) is uniformly distributed in the network, the number of affected links is approximately \(2 \cdot (d - 1) \cdot (1 - \eta)\), since out of the \(2 \cdot (d - 1)\) links that may be affected, \(2 \cdot (d - 1) \cdot \eta\) are already unreachable, they do not suffer any state change.

Altogether we have \(2 \cdot C_A\) allocations/deallocations in the network, which results in
\[
M^d_{\text{dyn}} \approx 2 \cdot C_A \cdot \text{clen} \cdot (2 \cdot (d - 1) \cdot (1 - \eta)) \cdot (W_n + W_a). \tag{2.21}
\]

In RBTA only one state change message is sent by each affected node, informing the TED about the place (identified by 2 node ids) and the amount (given in TE attribute format) of the change:
\[
M^*_{\text{dyn}} = 2 \cdot C_A \cdot \text{clen} \cdot (2 \cdot W_n + W_a). \tag{2.22}
\]

In RBTA-P the sent message is longer as it must also contain at most \(\text{wlen}\) SRG labels:
\[
M^{**}_{\text{dyn}} \leq 2 \cdot C_A \cdot \text{clen} \cdot (2 \cdot W_n + W_a + \text{wlen} \cdot W_f). \tag{2.23}
\]

The most important message of Eq. (2.22) and Eq. (2.23) is that neither \(M^*_{\text{dyn}}\), nor \(M^{**}_{\text{dyn}}\), depends on \(d\). In case of RBTA, \(M^*_{\text{dyn}}\) definitely scales better than \(M^0_{\text{dyn}}\), whereas in case of RBTA-P the amount of control channel load reduction depends on the ratio \(\text{wlen} / ((d - 1) \cdot (1 - \eta))\). That makes the deployment of RBTA and RBTA-P favorable especially in networks where cross-connects support – full or partial – wavelength conversion resulting in high \(d\).

2.5.3 TED size

Conventionally, the TED stores the WG as an array of edges. The rows of the array are \(2 \cdot W_n + W_e + W_a\) byte wide. Having \(|E|\) edges in the graph means a total amount of
\[
D^0 = |E| \times (2 \cdot W_n + W_e + W_a). \tag{2.24}
\]

In RBTA and RBTA-P the edge attributes are assigned to nodes. The invariant part of the stored data is given by the initial topology advertisement which is \(M^*_{\text{stat}} = M^{**}_{\text{stat}} = |V| \times (W_n + 2 \cdot (1 + d_{\text{max}} \cdot W_e) + W_a\) (see Eq. (2.22) and Eq. (2.23)).
Besides of this information the sum_cap array is also stored. In RBTA
the sum_cap is a simple edge \( \rightarrow \) capacity association, whereas RBTA-P
has a two-dimensional sum_cap as edge \( \times \) failure \( \rightarrow \) capacity association.
Altogether that makes

\[
D^* = M^*_{stat} + |E| \cdot W_a \quad \text{and} \quad (2.25)
\]
\[
D^{**} = M^{**}_{stat} + |E| \cdot |F| \cdot W_a. \quad (2.26)
\]

Summarized, \( D^* = O(D^o) \), whereas \( D^{**} = |F| \cdot O(D^o) \), meaning that
RBTA-P requires much larger storage space than the conventional scheme or
even RBTA. However, RBTA-P has internal support for protection schemes
and, conforming to Sect. 2.4.4, in case of failures, when protections become
activated and the connections are switched to backup paths, the database
of RBTA-P does not need to be modified, thus the control channel is not
loaded with additional LSA messages.

2.5.4 Path computation complexity

We can split the path calculation process into three phases: pre-process
phase, calculation phase and post-process phase. The number of procedural
steps we denote by \( P^\text{pre} \), \( P^\text{calc} \), and \( P^\text{post} \), respectively. The calculation phase
is usually based on Dijkstra’s shortest path algorithm [77], thus the task of
the pre-process phase is deriving WG attributes from the TED, whereas the
post-process phase is responsible for manifesting the connection provision in
LSA messages.

The complexity of path calculation is \( P^\text{calc} = O(|E| \cdot \log(|V|)) \). In the
conventional model, the TED stores the WG attributes directly, therefore
\( P^\text{pre} = O(1) \). However, the calculated path does not refer to all the affected
WG edges directly, and these indirectly affected edges need to be identified
after the route is found. That requires \( P^\text{post}_o = O(clen \cdot d) \) post-process steps.
On the contrary, in RBTA and RBTA-P, the post-process phase is simple,
the LSA messages are directly defined by the assigned path, result
\( P^\text{pre} = O(|E| \cdot d^\text{max}) \)

and \( P^{** \text{pre}} = O(|E| \cdot |F| \cdot d^\text{max}) \).

Putting it together, to run a shortest path algorithm on an accurate
WG, the models need as many calculation steps as follows:

\[
P^o = O(1 + |E| \cdot \log(|V|) + clen \cdot d) = O(|E| \cdot \log(|V|))
\]
\[
P^* = O(|E| \cdot d^\text{max} + |E| \cdot \log(|V|)) = O(|E| \cdot \log(|V|))
\]
\[
P^{**} = O(|E| \cdot |F| \cdot d^\text{max} + |E| \cdot \log(|V|)) = O(|E| \cdot |F| \cdot d^\text{max})
\]
Figure 2.5: Conventional controlling after failure

Note that the values $P^*$ and $P^{**}$ can be lowered if the computations are evaluated on-demand. For example, the free_cap() function should not be evaluated for each edge from the scratch, but only updated for those edges (or edge/failure pairs) that have suffered any modification since the last calculation, reducing the number of examined edges from $|E|$ to $clen \cdot \overline{d}$.

2.5.5 SRG identification and failure notification

To calculate the number of failure notification messages, we need the average default/backup path lengths: $\overline{wlen}$, $\overline{blen}$, $\overline{clen}$.

The conventional way of failure handling mechanism of shared protection, which has the most complicated procedure among the conventional protection schemes, is depicted in Fig. 2.5. First, the failure is detected. There may be (1) local detection, i.e. the ends of the link detect the failure, (2) distributed detection and localization [86][87] using monitoring cycles, however, a widespread solution is (3) that the end nodes of connection directly get aware of the failure by recognizing the Loss of Light. If there is no such kind of direct detection, a failure notification message is sent to the head end of the connection (Step 1). That requires on average $\overline{wlen}/2$ control messages. In Step 2, the head node performs the allocation of the backup resources. The amount of control messages that this procedure requires – ignoring Step 1 – is as much as the length of the backup path ($\leq \overline{clen}$).

Using RBTA-P, the failure notification flooding produces $|E|/2$ instances of control messages. (The message is flooded on every link in the network, except for the reverse links, i.e., those links that are oppositely directed compared to the flooding direction. In fact, every link means here every physical link, thus counting with $|E|/2$ instances is a rough overestimation by a factor of the number of wavelengths.)

Comparing these values, we get that $|E|$ is considerably higher than $\overline{wlen}/2 + \overline{clen}$ which was an upper bound estimation.

Note, however, that this example was about a failure that affected a
Figure 2.6: Increasing number of signaling messages for multiple connections

single connection. If there are more connections affected by the failure, as Fig. 2.6 depicts, the failure notification flooding mechanism of RBTA-P becomes more profitable: the number of connections that are affected by the failure \( f \) on the SRG consisting of edges \( E_f \) is

\[
|C_f| = \sum_{c \in C} I(\exists n(GPF(c, n, f, 0) \in (E^* \times E_f))) \quad (2.27)
\]

To approximate \( |C_f| \) we assume that the probability that an edge is member of the default path of a given connection is \( \frac{wlen}{|E|} \), whereas the probability than an edge is affected by a given failure is \( \frac{|E_f|}{|E|} \). We also assume that these probabilities are independent. The chance of a failure to affect a certain connection is the complementary probability of the event that a failure does not affect a certain connection: all the edges that the failure affects are not part of the connection. This probability is in case of low \( |E_f|/|E| \) approximately:

\[
1 - \left(1 - \frac{wlen}{|E|}\right)^{|E_f|} \approx |E_f| \cdot \frac{wlen}{|E|},
\]

thus \( |C_f| \) can be approximated as:

\[
|C_f| \approx |C| \cdot \frac{wlen \cdot |E_f|}{|E|}. \quad (2.28)
\]

To answer the question “Over how many connections is the proposed new failure handling paradigm favorable?” we assume that conventionally more than \( wlen \) notification messages are needed for each backup path activation and in this manner we express \( |C| \) from

\[
|C| \cdot \frac{wlen \cdot |E_f|}{|E|} \cdot \frac{wlen}{wlen} > \frac{|E|}{2}
\]

\[
|C| > \frac{|E|^2}{2 \cdot wlen \cdot |E_f| \cdot wlen}
\]

\[
|C| > \frac{1}{2} \cdot |E_f| \cdot \left(\frac{|E|}{wlen}\right)^2
\]

47
For example, we examine the COST266BT reference network [88], operating on a single wavelength. In this single wavelength network the 28 routers are represented as vertices, there are 41 pairs of directed edges ($|E| = 82$) and $\text{wlen} = 3.8$. We assume simple SRGs (each edge pair defines an SRG: $|E_f| = 2$). In this network, according to this rough approximation, if the number of connections is higher than 120, the proposed failure notification paradigm outperforms the conventional one.
Chapter 3

Inter-domain $p$-cycle protection

In a multidomain environment the expectation of the end-users is that they get the same or near the same reliability for long inter-domain connections as for the short intra-domain connections (distance fairness [89]). This expectation is, however, set back by several difficulties: a) physically longer connections may fail with higher probability and hence when using merely those traditional protection schemes which were satisfactory in intra-domain case, we do not get the desired grade of availability; and b) setting up protection in a multidomain environment, where multiple service providers are present, also raises common management and control issues.

Although the $p$-cycle protection scheme applied in multidomain environment consumes more spare capacity than other protection schemes, e.g., end-to-end dedicated protection (see Sect. 3.2.4), it has more advantages. It inherits the fast reacting time property of $p$-cycles, it does not require detailed intra-domain topology information (Sect. 3.1) and it is easy to deploy in GMPLS controlled networks [11]. It also provides higher availability for the connections and this availability, compared to the conventional protection schemes, is predictable:

For example, an end-to-end protected connection may fail with higher probability than expected (pre-calculated as common outage of the two disjoint paths) in that unfortunate case when failures – or more generally, SRLGs – in foreign, competitor domains are not independent. Typically, disasters cause problems in many networks of different operators at the same place.

If we consider the feasibility of a protection scheme also in this manner and want to eliminate such hidden dependencies, then topologically hierarchical networks with inter-operator contracts and protection schemes based on such contracts gain importance. However, this realization raises also the requirement of a (hierarchically) higher-level Management Plane where
these inter-operator contracts are agreed. And at this point \( p \)-cycle or any other not connection-oriented protection scheme become important as they do not require an individual contract for each connection.

3.1 Planning reliable inter-domain connections

Both for privacy and for scalability reasons, the interior topology of the domain is not advertised outside the domain. However, either by using an EGP or by communicating to a PCE, it is possible to find out what domain does a certain node belong to, moreover, the domain paths that lead to it and the list of traversed border nodes are also given. We assume that protecting the intra-domain parts of the inter-domain connections is the responsibility of the given domain, and here we describe a solution how to protect the inter-domain parts (i.e., the inter-domain links) of connections spanning over multiple domains.

3.1.1 Two-level hierarchical network model

We assume that the multidomain network is modeled by a two-level topology as shown in Fig. 3.1. The network is modeled by an undirected graph, the network nodes are the vertices of the graph, whereas the links between the network nodes are represented as edges. At the higher level only the border nodes and the inter-domain links are known definitely. The details of the internal topology of the domains, i.e., the real nodes and links, are hidden at the higher level, only virtual links between the border nodes denote the connectivity. These virtual links may carry aggregated information about bandwidth, delay, hop count, etc. \([90][91]\).

At the lower level, each domain is aware of its own internal topology, however none of them has any information about the network outside the domain.

![Multi-domain network model](image.png)

Figure 3.1: Multi-domain network model
3.1.2 Defining the set of candidate MDPCs

In order to apply inter-domain $p$-cycle protection in the network we have to define the candidate MDPC set. In conventional, intra-domain context there are several fast and efficient cycle-search algorithms [71][10], e.g., the Exhaustive Grow algorithm (see Sect. 1.3.4 and Sect. 4.1) that may be applied to get the (higher level) inter-domain $p$-cycle candidate set. If we want to apply these algorithms in inter-domain context, we have to model our network as a graph where the domains are represented by the nodes of the graph whereas the inter-domain links are the edges of the graph. Having the higher level network topology available at the common management plane, this graph can be constructed easily. Selecting the candidate $p$-cycle set, i.e., applying one of the cycle search algorithms on the constructed graph, is the task of the aforementioned common management plane.

3.1.3 Assigning candidate MDPCs

There are multiple strategies, how to assign candidate $p$-cycles to protect inter-domain links. The first strategy is to use rather short cycles, which involve only two or three domains and have a low number of straddling links. The advantage of this strategy is that contracts are made very easily on these short cycles (only 2 or 3 operators have to make an agreement). However, as the number of straddling links is low, the resource efficiency of the solution is bad. The second, opposite strategy is to use long inter-domain $p$-cycles having multiple straddling links to enhance resource efficiency. In this case, however, the intra-domain parts of the MDPCs, connecting on-cycle and straddling links of the MDPCs within the domains, have to be resolved (see Sect. 3.1.4), and having the ends of multiple straddling links in the same domain requires a more complex solution (see Sect. 3.1.5 and Sect. 3.1.6).

In our simulations in Sect. 3.2 we apply the CIDA algorithm (Sect. 1.3.5) for $p$-cycle assignment to protect inter-domain links.

3.1.4 Intra-domain part of inter-domain $p$-cycle

In Sect. 3.1.2 we have discussed that by modeling the domains as nodes at the higher hierarchy level we can select the set of candidate inter-domain $p$-cycles. However, these selected inter-domain $p$-cycles are incomplete since they were found in a graph that modeled the domain as nodes. In fact these domain internally have to connect the on-cycle and straddling inter-domain links of the $p$-cycle. This task gets complex if multiple straddling inter-domain links of a $p$-cycle end in the same domain.

The examples in figures 3.2 and 3.3 show which inter-domain links and intra-domain connections between border nodes (intra-domain parts) must be activated in the case of on-cycle and straddling link failures, respectively. Note that there are two types of border nodes in the figure: CBN is the
ending of an on-cycle link (on-Cycle link Border Node) and SBN is the ending of a straddling link (Straddling link Border Node).

Figure 3.2: Handling an on-cycle link failure by the higher level p-cycle

Figure 3.3: Handling a straddling link failure by the higher level p-cycle

The figures also show that, depending on the failed inter-domain link, the intra-domain part of the cycle may be configured to connect different border nodes. Normally, in case an on-cycle link fails or in case a straddling link not connected to the given domain fails, the backup traffic passes through the shortest path between the two CBNs. However, in case of a straddling link failure, in both domains linked by the straddling link the SBN must be connected to both CBNs internally. This behavior is implied by the p-cycle concept: the p-cycle can protect two units of traffic of a straddling link as the operational on-cycle links provide two disjoint backup paths for the failed straddling link (Fig. 3.3).

As long as there are no straddling links connected to the domain, the intra-domain planning task (finding a route between two gateways) is easy and self-evident: the gateways should be connected via the most suitable route. Suitability is meant here related to a specific optimization goal (see Sect. 3.1.5). However, if there are one or more straddling links connected to the domain and they end in a border node which is not a CBN at the same time, the planning of intra-domain connection becomes more complex.

Figure 3.4(a) shows the logical connections between border nodes that must be preconfigured inside the domain. These logical connections must be further refined into real paths, i.e., we have to define which intra-domain
links realize the logical connections. This can be realized in different ways as we present it in the next section. Each of these ways provides a different global availability for the protections and requires a different amount of resources.

3.1.5 Optimization goal

At this stage of network resolution, from global optimization perspectives, basically there are two opposite goals: either we can focus on the expenses and get the connections with the least cost (LC); or we can target optimization on reliability providing the highest availability for the connection (Most Reliable – MR).

Most Reliable inter-domain \( p \)-cycle intra-domain parts

If we take the latter choice, first, we must select the most reliable paths separately between each SBN–CBN pair and also between the two CBNs (Fig. 3.4(b)). After the paths are found we take the union of the links of the selected paths. The link capacity assignment algorithm is easy: Initially we allocate 1 unit of capacity on each link that is selected, i.e., the link is part of at least one path. Next, we examine the SBNs node-by-node: In each SBN two paths are originated (and connected to the two CBNs). On those links that are common in these two paths, we must allocate 2 units (dashed lines in the figure) of capacity instead of 1 unit (solid lines), as they have to carry both units of capacity of the protected straddling link.

Least Cost inter-domain \( p \)-cycle intra-domain parts

Finding the cheapest, i.e., the least resource-consuming connection between border nodes inside the domain (Fig. 3.4(c)) is a more complex optimization problem which may be solved by Integer Linear Programming (ILP) or other general solver tools.

The LC-finder algorithm that we propose and use is a greedy heuristic that builds a spanning tree, which is known to be suboptimal, however, the algorithm is fast, and if the cost metrics of the edges are close to each other, it results in a good approximation:

1. Initially, we register each border node into the set \( S \) as subgraphs containing only a single node.

2. For each subgraph \( s_i \in S \) we search for its nearest \( s_j \in S \) neighbor and compute their distance \( d(s_i, s_j) \).

3. We select the closest \((s_i^*, s_j^*)\) pair with minimal \( d(s_i^*, s_j^*) \). Let \( R^* \) denote the shortest path between \( s_i^* \) and \( s_j^* \). Delete \( s_i^* \) and \( s_j^* \) from the set \( S \) and insert \( s_x = s_i^* \cup s_j^* \cup R^* \) into \( S \) instead of them.
4. If there is still more than one member in $S$, go to step 2.

5. As a result, $S$ will contain only one subgraph $s_{res}$ which we were looking for.

The calculation’s complexity is polynomial. Having $|V|$ nodes, $|V_B|$ border nodes and $|E|$ edges in the domain, in the first iteration we perform $|V_B|$ times Dijkstra’s algorithm, totally in $O(|V_B| \cdot |E| \cdot \log(|V|))$ steps. The algorithm has $|V_B| - 1$ iterations resulting in an overall complexity of $O(|V_B|^2 \cdot |E| \cdot \log(|V|))$.

The capacity assignment in this simple heuristic is the same as described formerly for the MR case: on each link of $s_{res}$ we allocate one unit of capacity and, additionally, another unit of capacity is allocated on those links which are common in the path pair leading from an SBN to the two CBNs.

![Diagram](a) Internal connections to realize

(b) Most reliable internal connections (MR)  
(c) Least cost internal connections (LC)  
(d) Ring-based internal connections (RB)

Figure 3.4: Logical internal $p$-cycle connections and alternate resolutions

From protocol related and feasibility aspect, in both approaches (MR and LC) we have assumed that the network supports LSP aggregation (this may require specific network devices) and that in case of straddling link failure occurrence the SBN can notify either of the CBNs to modify the intra-domain track of the MDPC in order to connect the SBN to the MDPC. This notification is needed since the default intra-domain track of the MDPC connects the two CBNs. These issues are discussed in [11].

### 3.1.6 Achieving further reliability

Compared to the intra-domain $p$-cycles the inter-domain cycles may be and usually are also much longer and also less reliable. That makes reasonable to improve their availability and protect also the inter-domain cycles or at
least some part of it. One possibility is to assign two separate $p$-cycles to the inter-domain links, which have only the protected link in common. This way the first protection is protected by another $p$-cycle. This choice raises many questions, partially addressed to the general – not only inter-domain – simple $p$-cycles, and topologically is not always feasible. Similarly, routing the default path on a straddling link of a cycle, the cycle can serve as two disjoint backup paths for the link (see $p$-Cycle Multi-Restorability Capacity Placement [92]). Now we do not deal with these options.

We propose to assign protection to those parts of the long loop which are easy (from topological and protocol related aspects) to protect. In our case these easy-to-protect parts are the intra-domain parts of the long inter-domain cycles.

We get each internal logical connection realized if we find a cycle that passes through each affected border node (Fig. 3.4(d)). Additionally, in this case there are two link-disjoint paths between each border node pair. We call this MDPC intra-domain resolution Ring Based (RB) solution, since the connection of border nodes is based on a proper ring. We have to deal, however, with the following problems:

- There are no proper cycles containing all the border nodes. In that case we may find the cycle that is “the richest” in border nodes, i.e., the cycle which contains most CBNs and SBNs. Afterwards, we can connect the remaining border nodes to this cycle. Of course, the latter connected nodes will not benefit from the advantage of the RB solution.

- There may be more than one cycle that contains each border node. Which one should we choose among them? If we take capacity-efficiency also into account, a reasonable choice is that cycle, or the shortest of those cycles, which have a direct – SBN-free – connection between the two CBNs. These cycles require less capacity on the direct connection. Discussing the resource requirement of RB we will see why.

It is clear that, compared to LC and MR, the RB intra-domain connection realization of inter-domain $p$-cycles requires the most resources as we define (at least) two paths for each CBN–CBN or SBN–CBN connection. Of course, these connections share the resources among themselves but the capacity requirement is still high.

Figure 3.5 illustrates the capacity requirement of Ring-Based intra-domain solution. The normal case is when the protection is routed between the two CBNs. As Fig. 3.5(a) shows, the internal cycle provides two alternate routes, we use the shortest one by default and whenever a link failure makes that route unavailable, we can still route the protection via the longer path. This scenario requires 1 unit of capacity on each link.
The next figures, Fig. 3.5(b)-3.5(d) show scenarios where there is an inter-domain straddling link failure. In Fig. 3.5(b) we see the default case where there is no failure in the internal cycle at all or the failure affects neither route\textsubscript{1} nor route\textsubscript{2}. Figures 3.5(c) and 3.5(d) illustrate scenarios when either route\textsubscript{1} or route\textsubscript{2} fails. In these cases the two routes have common parts on some links where 2 units of capacity must be allocated.

Generally for RB solution, to protect the inter-domain p-cycle against intra-domain failures, instead of 1 unit, 2 units of capacity are required on those links which route the protection from the SBN directly to the CBNs. Consequently, if there are SBNs on both half-cycles between the two CBNs, 2 units of capacity must be allocated everywhere on the cycle. Formally, the accumulated resource requirement of an intra-domain RB ring is $|E_c|+2|E_s|$, where $E_c$ is the set of links that form the direct ring segments between the two CBNs, $E_s$ the set of links forming the rest of the RB ring, and the cost of each edge is 1 unit.
3.2 Performance of inter-domain $p$-cycle protection

In this section we analyze the performance of different MDPC solutions. We compare the $RB$ (Ring-Based) intra-domain cycle resolution scheme to the $LC$ (Least Cost) and $MR$ (Most Reliable) solutions and examine them from the aspect of resource requirement and provided availability. The figures of this section also show curves corresponding to $DP$ (Dedicated End-to-End Protection) and $NP$ (No Protection), however, these are used only as references in a global network without any domain boundaries and any topology aggregation.

3.2.1 Simulation setup

Numerically we have investigated the topic of inter-domain $p$-cycles in 5 dimensions:

1. the network connectivity (described in Sect. 3.2.1, Test networks);
2. the value of the Link Failure Coefficient (defined in Sect. 1.2.1);
3. what intra-domain cycle resolution we use (LC, MR or RB, see Sect. 3.1.5 and 3.1.6);
4. what intra-domain protection do we apply (described in Sect. 3.2.1, Protection on intra-domain part of working paths) and
5. how much spare capacity we provide for resilience (results in Sect. 3.2.6).

Test networks

Choosing appropriate topologies for the simulations is of high importance since special topology attributes can significantly influence the results. One basic expectation against test topologies is that they should be natural in sense that they should be real networks or at least they should have attributes (e.g., domain size, distances, node and domain connectivity) similar to real networks.

Unlike in case of single domain networks, like NSF Net or COST266 topologies [88], it is hard to find real inter-domain networks. For that reason we examined the protection schemes on 3 different network topologies, a realistic but hypothetical one and two “artificial” ones:

- E1Net [93], a realistic European multidomain network consisting of 17 national domains (Fig. 3.6).
• Xnet, a fairly regular grid network, organized into 9 grid groups each of 16 nodes (Fig. 3.7).

• Tnet, which gives a fair compromise between realistic and regular / artificial networks, with 7 domains and an average nodal degree of 3 (Fig. 3.8). This network was used also in [10] for simulations.

Whilst Xnet is well suited for testing the dependence between resource consumption and provided availability, Tnet provides a good testbed to examine availability improvement and with the E1Net we can make observations on realistic, heterogeneous cases.
For the sake of simplicity we have assumed, that the cost metric of the links is uniform within the networks. This assumptions makes the cost-based shortest path routing simple, furthermore, the calculation of resource usage also becomes easier as it does not have to weigh the amount of allocated capacity with the cost of the link.

Traffic pattern
In each test network 3000 traffic requests were generated. First, the end node pairs of the requests were selected at random. Each node had the same probability to be selected as an end node. If both end nodes belonged to the same domain, the end node pair assignment was repeated for the given request.

Then the bandwidth demand was defined for each request so that the values were uniformly distributed in the integer interval of \([1, 6]\) bandwidth units.

After the traffic requests were generated, the default path of the connection was defined for each of them in two steps: first, at the higher hierarchy level, the shortest path was found between the domains of the end nodes, next, at the lower hierarchy level, the internal connections within the assigned domains were defined.

Applied protection combinations
In our simulations we have employed three basic types of protection schemes and their combination for inter-domain and intra-domain parts of the connections: no protection, dedicated 1+1 protection and \(p\)-cycle protection. The protection assignment is hierarchical, i.e., first we protect the inter-domain part of the connections, next the intra-domain parts are protected.

In case of no protection, the connection is routed on the shortest path between the end nodes of the connection. In case of dedicated protection, first the shortest path is found and assigned to the connection as the default path, next another shortest path – disjoint with the default path – is found and
assigned to the connection as the backup path. In case of $p$-cycle protection, first, the shortest paths are found for all the connections, and then $p$-cycles are assigned to the working capacities using the CIDA algorithm (Sect. 1.3.5).

In multidomain networks these protection schemes are applied and combined as follows. To protect the inter-domain links of the connections, we can apply three different types of inter-domain $p$-cycles that differ from each other only in the way of their intra-domain part resolution (LC, MR or RB). Each inter-domain $p$-cycle protection can be combined, however, with a different intra-domain protection scheme protecting the intra-domain segments of the default path. The intra-domain protection is either:

- $p$-cycle protection (and the combination is referred to as $CIDA$),
- dedicated protection ($CIDED$) or
- no protection at all ($CIDA0$).

Note that we refer to the first combination as $CIDA$, although $CIDA$ already names the algorithm presented in Sect. 1.3.5. However, the inter-domain $CIDA$ behaves in the same way as the original CIDA algorithm. The only difference is that it gives priority to the inter-domain links: the intra-domain network parts are processed only after the inter-domain link are protected.

If we took into account each possible inter-domain $p$-cycle scheme $\times$ intra-domain protection combinations, that would result in $3 \cdot 3 = 9$ different schemes. The performance of all these schemes would be hard to represent in one figure. Knowing, however, that $CIDA0$ solutions (without any intra-domain protection) show low performance of availability (see Fig. 3.13) in the global comparisons we have left out the results of $CIDA0$.

Furthermore, as references, the simulation results of end-to-end dedicated protection and no protection cases are also evaluated. Note, however, that these evaluations are done on topologies that assume that the network is not divided into domains but the whole network forms a single domain. In these cases the originally inter-domain connections become intra-domain connections and we do not have to deal with problems regarding inter-domain routing (e.g., topology aggregation, etc.).

### 3.2.2 Evaluated performance metrics

The performance of the examined protection scheme can be evaluated by many properties. We have chosen the followings:

**Unavailability:** it is the average unavailability of the connections in the network. This metric is derived from the average Availability ($Unavailability = 1 - Availability$). For the sake of simple and fast calculation, for the $p$-cycle protected connections, we have evaluated
the Serial-Parallel approximation of the availability instead of deriving the exact value. This simplification introduces approximation error, however, in Sect. 4.3 we will prove that this error is negligible in most cases.

**Relative resource consumption:** it is the quotient of the total amount of resources required by the protected connections (default and backup paths) and the amount of resources required by the unprotected connections (only default path).

**Relative unavailability reduction:** it is the quotient of the *unprotected* unavailability and the unavailability of the examined protection scheme.

**Relative thrift:** it is the quotient of the *relative unavailability reduction* and the *relative resource consumption*. It expresses how “economical” is the protection scheme. Higher relative thrift values can denote higher availability enhancement with the same resource requirement, or lower resource requirement for the same availability enhancement.

**Tail behavior:** the tail behavior of the connection availability expresses for any *availability lower bound* \(x\) what is the ratio of connections that have higher availability than \(x\).

### 3.2.3 Unavailability reduction

Our basic aim with planning different \(p\)-cycle protection schemes was reducing the unavailability of connections. Now we compare the LC and the MR strategies to RB and examine how they perform when applied in CIDA and CIDED schemes.

Figure 3.9 shows the average unavailabilities of connections protected by different protection schemes. The simulations were carried out on a wide range of the LFC value (Sect. 1.2.1). Figure 3.9(a) shows that in Tnet CIDED-RB results in the least unavailability. We can observe the same in Xnet (Fig. 3.9(b)). If we order the resilience scheme by provided availability, we can state that the performance of each algorithm is close to the theoretical *Dedicated End-to-End* protection; CIDED algorithms outperform their CIDA pairs; and RB algorithms come before MR and LC algorithms. In order of magnitude, however, there is no significant difference between the connection unavailabilities provided by the different protection schemes. This can be explained by the fact, that both in Tnet and in Xnet, each of the examined protection schemes can protect against single failures, they differ only in the protection efficiency of multiple simultaneous failures.

It may be a surprising result that CIDED algorithms, from the availability point of view, perform better than CIDA do. The reason for this
behavior is that the diameter of the domains is relatively small, hence the intra-domain connection segments are short. To protect these short segments Dedicated E2E protection fits better than \( p \)-cycle protection assigned by the CIDA algorithm (which focuses on optimization of resource consumption instead of high availability). For example, we have a 1-hop long intra-domain connection segment. This can be protected using Dedicated E2E with a 2-hop long backup path, whereas the CIDA algorithm takes it as an on-cycle span and assigns to it a \( p \)-cycle which is 6-hop long. In latter case the backup path is 5-hop long. This longer backup path results usually in lower availability enhancement than the shorter one (assigned by Dedicated E2E). That is why CIDED algorithms provide higher availability than their CIDA pairs.

On the contrary, in E1Net (Fig. 3.9(c)), the topology of the network is suitable to find and to set up two disjoint E2E paths for inter-domain connection, however, within the domains, to set up intra-domain connections between the border nodes, the greedy algorithm that finds two disjoint paths is likely to fail. That is why the CIDED schemes perform worse than the CIDA schemes.
### 3.2.4 Resource consumption

We have compared the protection schemes to the no protection case by means of resource consumption. As expected, the strategies that result in higher availability demand more additional resources. Figure 3.10 points out this behavior. Dedicated protection requires roughly 2.5 times as much capacity as no protection (i.e., the backup routes are on average 1.5 times longer than the working paths), and each cycle-based protection scheme requires even more: CIDED-RB approximately 4 times as much as no protection. The intra-domain links employed by higher level \( p \)-cycles are wasted in sense that their resources are allocated, however, in contrast to the inter-domain links, the higher level \( p \)-cycle does not offer protection for their traffic. This explains the relative high resource consumption of MDPCs.

Another reason for relative high resource consumption of the \( p \)-cycle-based schemes is that default path and the protection assignment are two separate tasks in our algorithms (as the CIDA algorithm assumes that the default paths of the connections are already given). This separation results in suboptimal resource usage.

Both the provided availability enhancement (unavailability reduction) and the resource requirement are important metrics of protection schemes. It is obvious that by sacrificing more spare capacity we can gain higher availability for the connections. But how does the amount of availability improvement relate to the amount of spare capacity consumption? To answer this question we have introduced the metric of relative thrift (Sect. 3.2.2). Figure 3.11 illustrates the relative thrift of the examined protection schemes. The figures show that having a low Link failure coefficient (lower than \( 10^{-6} \)) the unavailability reduction is 10 – 100 times more than the relative additional resource requirement of the schemes. Moreover, it can be seen that it is worth to invest into RB based schemes because their relative thrift is the highest (CIDED-RB in Tnet and Xnet, see Figures 3.11(a) and 3.11(b); and CIDA-RB in E1Net, as Fig. 3.11(c) depicts).
3.2.5 Getting the desired ratio of connection with predefined availability

The measurement and analysis of availability enhancement and resource consumption presented in Sections 3.2.3 and 3.2.4 are important since, conventionally, these two attributes are the most relevant performance descriptors of the protection schemes. However, as they are aggregated values, they are not expressive enough to really understand the difference between the different protection schemes. In this section we compare these protection schemes on different topologies by means of the tail behavior of the connection availability. The figures here and in the following section (Sect. 3.2.6) show results for $LFC = 3 \cdot 10^{-6}$ (conforming to the nominal values of $CC = 300[km]$ and $MTTR = 8[h]$).

Figure 3.12 shows the rate of connections having higher availability than a given ($x$) lower limit. Figure 3.12(b) illustrates the strength of the CIDED schemes and RB intra-domain part resolutions. on the regular topology: $CIDED-RB$, $CIDED-MR$ and $CIDA-RB$ satisfy the most connections with high availability.

Figure 3.12(a) depicts the trends in Tnet. It is worth to see the behavior
of the curve corresponding to the DP scheme. To a small ratio of connections it can provide high availability, as much as CIDED-RB – these are the short connections –, however, for most of the connections it offers a relatively low availability.

The Pan-European multidomain network, the E1Net shows different behavior as described for the other test networks. Figure 3.12(c) shows that on the E1Net topology for a moderate number of connections (approx. 8%) CIDED performs worse than CIDA. This can be explained by the fact that dedicated protection inside the domain cannot be established in each domain since there are sparse domains where disjoint paths cannot be found by the greedy shortest path finder algorithm employed in our dedicated protection scheme. The CIDA schemes are more resistant against such topology constraints than CIDED.

The same fact (sparse topology) explains the negligible dominance of the RB schemes: in many domains there were not found cycles that contain each of the border nodes.

Figure 3.12: Tail behavior of protection schemes
We have also examined the tail behavior of the CIDA0 protection schemes. Figure 3.13 underpins our expectation about CIDA0 schemes that they are not viable solutions since the availability that they provide for connection is not much more than that can be achieved without any protection. It is not worth to introduce inter-domain link protection, when the intra-domain part remains unprotected.

![Figure 3.13: Tail behavior of CIDA0 protection schemes in Tnet](image)

### 3.2.6 Performance in overloaded networks

![Figure 3.14: Tail behavior of protection schemes in overloaded networks](image)

Section 3.2.4 points out that the price of providing high availability is high: a complex protection scheme demands 3 – 4 times as much free resources as the connection provisioning without protection. Because of this,
inter-domain $p$-cycles are suggested to be used in networks with plenty of free capacity. But what happens if the network gets partially overloaded?

Figure 3.14 shows the tail behavior of the investigated protection schemes in Xnet where without any protection there was an average link load of 20% (intra-domain) and 31% (inter-domain). It can be seen that, compared to Fig. 3.12(b), the Dedicated E2E protection is the only scheme that does not suffer any loss due to resource shortage. Using other schemes only about 80% of the connections can be protected fully. The performance of CIDA-RB is surprising: despite of its relatively high resource usage (see Fig. 3.10), it provides high availability for more connections than schemes requiring less spare capacity. The reason for this behavior is that the $p$-cycle used at the lower level to protect intra-domain working path, is much more elastic than the dedicated protection: the dedicated protection scheme leaves the working path unprotected if a disjoint backup path to it cannot be found whereas the $p$-cycle tries to protect as many links as possible and finally results in much less unprotected working path links than the dedicated protection does.

The simulations also showed that the inter-domain links did not get overloaded, only the intra-domain links did at the lower level. However, after increasing the link capacity by 10%, implying that the initial intra-domain load sinks to 18%, there was no resource shortage at all.

Summarized, even though CIDED strategies provide the highest availabilities, due to their rigidity, they perform poorly in an overloaded network. Furthermore, since the network links are not homogeneously loaded, resource shortage happens sooner than the results of resource consumption of the different schemes suggest (Fig. 3.10).
Chapter 4

Improving connection availability by $p$-cycles

In recent years the $p$-cycle protection scheme became increasingly popular within the research area of resilience. Using $p$-cycles we can significantly increase the availability of the connections, and being a pre-calculated protection scheme, the amount of required additional link state information and thus the signaling overhead remains moderate.

As already presented in Chapt. 3, originally, we were applying $p$-cycles in multidomain environment to protect inter-domain links. Our primary goal was to find protection schemes that are feasible and efficient in inter-domain networks. However, during the research we have also faced interesting problems and questions that were related to $p$-cycles in general and not only to $p$-cycles in multidomain environment. In this chapter we focus on two of those questions:

- How to collect candidate cycles that may cover the working resources optimally?
- How much do $p$-cycles improve the availability of the connections exactly?

To examine these problems and to answer these questions, section 4.1 presents an enhancement of the Grow cycle search algorithm, which results in an extended set of candidate cycles compared to the result of its predecessor, the original Grow algorithm. In Sect. 4.2 we give an efficient calculation model for getting the exact availability for $p$-cycle-protected connections, and afterwards we discuss how much do we lose on the accuracy if we use the fast, well-known Serial-Parallel availability calculation model instead of the accurate one (Sect. 4.3).
4.1 The Exhaustive Grow candidate cycle search algorithm

In order to assign and allocate $p$-cycles for protection, first, a proper set of candidate $p$-cycles must be enumerated.

The principle of the candidate cycle set searching algorithms (see Sect. 1.3.4) is that they take a primary cycle set – cycles of minimum size, i.e., for each link in the network its shortest bypassing path is found and together they form a cycle – and the cycles are grown by replacing links with bypassing paths. The shortcoming of the Grow and Expand algorithms [76] is that these do not consider that replacing a link to a disjoint route may restrict the replacement of another link in the $p$-cycle, therefore the output of the algorithms depends on the sequence of the on-cycle links. This way these algorithms result in a smaller set of $p$-cycles, and they may leave out even useful $p$-cycle instances. Figure 4.1 shows an example where, starting from the same link of the same primary $p$-cycle, the result of the Grow algorithm depends on the on-cycle link sequence (clockwise or counterclockwise).

![Figure 4.1: Sequence dependency of the Grow algorithm](image)

This fact is not a practical problem in the case of a dense network topology containing a lot of links and nodes with high nodal degree, since the difference in the spare resource requirement to the optimal set of capacitated $p$-cycles is less significant [76], but in the case of less dense topology or tight resources the difference is not negligible. For this reason, we suggest the Exhaustive Grow algorithm based on Grow for less dense networks. Like its predecessor, this algorithm also requires a primary set of $p$-cycles that can be generated by the Straddling Link Algorithm (SLA) [71], for instance. Unlike the Grow algorithm, the Exhaustive Grow considers all the possible sequences of the on-cycle links, resulting in a larger set of $p$-cycles. The algorithm can be represented in pseudo-code as follows:

```plaintext
ExhaustiveGrow(CycleSet pcs, Graph G) {
    while (Cycle p = pcs.next()) {
        Graph X=Graph();
        for_each(Node n in p) X.add(n);
        for_each(Link l in p) X.add(l);
        for_each(Link l in p) {
            Route r=ShortestPath(l.head, l.tail, (G\X)∪{l.head, l.tail});
        }
    }
}
```
if (r!=null) {
    Cycle p'=p;
    p'.remove(l);
    p'.add(r);
    pcs.add(p');
}
}

4.1.1 Comparative results

We have compared the performance of the Exhaustive Grow to other algorithms by means of the size of the produced candidate cycle set and the resource consumption of the CIDA algorithm. The tests were run on domains taken from the already presented E1Net (17 domains) and Tnet (7 domains) network topologies (Sect. 3.2.1). In Figures 4.2 and 4.3 we can see two column sets. The 17 columns on the left hand side of the figures each represent a domain of E1Net, whereas the 7 columns on the right hand side each represent a domain of Tnet.

First, we want to confirm that the Exhaustive Grow finds a larger set of cycles than the Grow and Expand algorithms do.

![Figure 4.2: The strength of the Exhaustive Grow algorithm: relative amount of generated candidate p-cycles](image)

Figure 4.2 shows that, on the one hand, the Grow algorithm results in more candidate p-cycles than the Expand algorithm, on the other hand, the Exhaustive Grow algorithm produces even more candidates. In practice, however, we exploit this strength of the algorithm only in small networks since technically, in larger networks, it is difficult to manage more thousand candidates.

Next, we wanted to show, that the amount of resources required by
the CIDA algorithm is less than the resource requirement of the other algorithms. In each domain we have generated traffic demands with random end nodes, altogether 9000 requests in each network (i.e., on average 530 in each domain of E1Net and 1125 in the domains of Tnet). We have applied to this traffic the CIDA algorithm, with the candidate cycle sets produced by the Expand, Grow and Exhaustive Grow algorithms. Figure 4.3 shows the total (working and backup) capacity requirement of the Expand and Exhaustive Grow algorithms compared to the Grow algorithm. The results, on the one hand, confirm that applying the Grow algorithm instead of Expand, in many cases reduces the resource consumption (on average by approx. 5%). On the other hand, using the cycle set produced by the Exhaustive Grow we can reduce further resource consumption in most of the cases (on average by approx. 5%). Figure 4.3 also shows that the performance of the algorithms strongly depends on the topology of the network.

In [71] the authors set the performance of the Expand and Grow algorithms against the optimal (ILP) solution. Using the USA topology (with 28 nodes and 45 links) we have re-evaluated the results by simulating the CIDA algorithm with 5000 random connections. The simulations were carried out with the cycle sets of the Expand, Grow and Exhaustive Grow algorithms, respectively. We have found that both for Expand and for Grow the resource requirement of the CIDA algorithm in our simulations was nearly the same as the authors presented in [71] (less than 1% deviation). The results are summarized in Table 4.1. The compared metrics are taken from [71]: Redundancy is the ratio of the required backup and working capacities, whereas % Diff is the difference of redundancy compared to the optimal (ILP) solution expressed in percentage.

The table shows that Exhaustive Grow is a fair compromise between Grow and the ILP solution. Knowing that the running time of CIDA is proportional to the number of cycles, we can see that on the price of approx. double computation time with the cycle set of Exhaustive Grow we can get

![Figure 4.3: Relative amount of CIDA resource consumption](image)
Table 4.1: Comparison of different algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CIDA-Expand</th>
<th>CIDA-Grow</th>
<th>CIDA-Ext.Grow</th>
<th>ILP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles</td>
<td>96</td>
<td>839</td>
<td>1753</td>
<td>7321</td>
</tr>
<tr>
<td>Redundancy</td>
<td>104.1%</td>
<td>95.2%</td>
<td>91.2%</td>
<td>83.6%</td>
</tr>
<tr>
<td>% Diff</td>
<td>24.5%</td>
<td>13.9%</td>
<td>9.1%</td>
<td>0</td>
</tr>
</tbody>
</table>

a backup capacity requirement that is 34% closer to the optimal solution than with the cycle set of Grow.

4.2 Incremental availability calculation method for \( p \)-cycle-protected connections

To estimate the availability of \( p \)-cycle protected connections we can apply different methods (presented in Sect. 1.2.3). These methods are either fast but then we have to count with approximation error (e.g., \( S-P \) method and heuristics) or they provide accurate estimation but are slow (e.g., examining each atomic network state). However, we want to have an availability calculation method which is fast and accurate at the same time.

In contrast to [94] that analyzed the reliability of the whole \( p \)-cycle protected network, here we examine individual \( p \)-cycle protected connections. The calculation model we propose aims to provide accurate transformation of equipment availability metrics into connection availability metric. It is obvious that not all the devices in the network and not all the states of these devices have to be considered in the calculation: we have presented in Sect. 1.3.2 that, in contrast to many other conventional shared protection concepts, whenever two or more connections share the same \( p \)-cycle as protection, there are no network states, when these connections would switch their traffic to the given \( p \)-cycle simultaneously, using the same resources (on-cycle links of the \( p \)-cycle) concurrently.

In that sense the connections are independent. That way the availability calculation of the connections can be performed separately, i.e., independently of other connections and of those resources that are not used by the examined connection itself.

Furthermore, as the \( p \)-cycle is a local resilience concept, we want to formulate that self-evident observation that the availability calculation of the connection can be segmentized. In case the \( p \)-cycles are short so that the \( p \)-cycles protecting the head segment of the connection are independent (link-disjoint) from those \( p \)-cycles that are protecting the end segment of the connection, failures at the head segment of the connection have no impact on the device configuration around the tail of the connection, and vice versa. That way also the availability of the head segment is independent of the availability of the tail segment and these availabilities can by calculated
It is known, that having a path in the network, passing between two nodes, say from node \( s \) to node \( t \) with a known availability, if the path is lengthened to node \( t' \), the availability of the new \( s \to t' \) path does not have to be calculated from the scratch, but the availability of \( s \to t \) can be re-used in the calculations. The basic idea of the proposed calculation model is analogous: when calculating the availability of the \( p \)-cycle protected connection \( s \to t' \), we can re-use the formerly calculated availability metrics of the shorter \( s \to t \) connection part.

### 4.2.1 Scope of the model

In our network and protection model we have made the following assumptions:

1. **Link-protection:** Since \( p \)-cycle is originally a link-protection concept we do not consider node failures, however, after the well known node \( \to \) link graph transformation, where the original node is substituted by two nodes interconnected with a directed edge, the model would be able to deal also with node failures.

2. **Single failure survivable protection:** Although there are studies that propose dual protection for each link to gain multifailure survivable network [60], as an initial work, in our model we assume single \( p \)-cycle protection.

3. **Binary link state:** Failure detection is not our task. We do not deal with any subsidiary outage due to transmission delay on the control channel or protection activation delay. We assume these intervals are negligibly short compared to the link outage (MTTR). We assume that the link has binary state: it is either Up or Down. The availability metric of a link is described by a simple value expressing the probability that the link is in Up state.

4. **Static switching:** The state of the protected connections is not known by the \( p \)-cycle controllers, i.e., the controlling device that activates the backup path(s) is not aware whether a given connection is broken at another part of the network or not. It is an evident still important assumption, since it implies that the different \( p \)-cycles in the network are independent in that sense, that their protection activation mechanism depends only on the state of their own on-cycle and straddling links and this mechanism is predetermined.

5. **Monotonicity assumption:** As a consequence of the previous assumption (static switching), the state of the connection is a monotonic function of the state of the network components. This means, that changing the state of any component from Up to Down cannot result in state improvement of any connection.
6. **Priority based switching**: We assume priority-based failure handling. This means, that in case of simultaneous on-cycle and straddling link failures the behavior of the $p$-cycle is **pre-defined** regardless of the failure occurrence order. Different failure handling priority strategies are enumerated in [13].

Note that these assumptions are normal expectations, and they do not constrain the applicability of the model, they intend only to formulate basic simplifications to facilitate the calculations.

### 4.2.2 Notations

For availability calculations we model our network as an undirected graph. The network consists of atomic components (usually network links) which may fail. These components (links) are represented by the edges of graph, whereas the joining nodes of these links are the vertices of the graph. The whole set of the links is denoted by $E$, while for denoting a single link usually we use $e$ ($e \in E$) which is often used with an index or other special markings (e.g., $e_i$, $e^*$, etc.).

In our availability model we have assumed that each network link $e$ has two states ($S(e)$): it may be either operational (up, $S(e) = 1$) or in failure (down, $S(e) = 0$) state. The state of each link is independent from the state of any other link ($\forall e', e'' \in E, e' \neq e'': S(e')|_{S(e'')} = S(e')$). The availability of a link (network component) is a probability metric indicating that the link is in up state: whereas the unavailability is the complement measure of availability:

$$A(e) = P(S(e) = 1),$$
$$U(e) = P(S(e) = 0) = 1 - A(e).$$

We examine individual connections. The connection is denoted by $conn$, using the set of links $C$ ($C \subseteq E$).

Let us consider the default (working) path assigned to the connection $conn$ as a sequence of links:

$$W = (w_1, w_2, ..., w_n), \quad W \subseteq C,$$

where $w_i$ is the $i$th item in the sequence. The link $w_i$ connects nodes $v_{i-1}$ and $v_i$. As the working path is protected with $p$-cycle, which is a link-protection, for each $w_i$ link there is a cycle assigned. From resilience perspective, the question is not where the alternate backup path leads but which links must be operational to be able to provide a backup path avoiding $w_i$ (e.g., assuming Strategy I. of priority based switching (Sect. 1.3.3), in case of an on-cycle link failure even the straddling links need to be operational to provide protection to the failed on-cycle link, even if they are not part of the backup
path). This set of links, including the protection path and potentially also other on-cycle and straddling links, is denoted by

$$Prot_i = \{p_i,1, p_i,2, ... , p_i,J_i\}.$$  

From another point of view, we can treat the connection as a series of protected links. The length of the connection is $n$, meaning that along the default path $n + 1$ nodes ($v_0$ is the source node, $v_n$ is the destination) are connected with $n$ links. If we split the connection at the $i^{th}$ node into two, we get $conn_i^H$ and $conn_i^T$ as the head and the tail part of the connection, with links enumerated in set

$$H_i = \{e_{hi1}, e_{hi2}, ... e_{hih}\} = \bigcup_{k=1}^{i}(w_k \cup Prot_k).$$

and set

$$T_i = \{e_{ti1}, e_{ti2}, ... e_{tit}\} = \bigcup_{k=i+1}^{n}(w_k \cup Prot_k),$$

so that concatenating the tail to the head (concatenation denoted by the “.” symbol) results in the original connection without any loss or surplus:

$$\forall 0 \leq i \leq n : conn = conn_i^H . conn_i^T.$$  

(4.1)

For the different $H$ sets the relation

$$\forall i < j : H_i \subseteq H_j$$  

(4.2)

is true, similarly,

$$\forall i < j : T_i \supseteq T_j.$$  

(4.3)

For each head and tail pair we denote the set of overlapping links (i.e., at node $v_i$ those links that are used in both segments before and after $v_i$) by

$$L_{i,0} = H_i \cap T_i.$$  

(4.4)

The 0 in the second index of the $L$ metric indicates the overlap distance, and expresses that the head and tail connection segments that are examined are direct neighbors, i.e., the distance between them is 0. That way we can generalize this kind of annotation and denote even far overlaps:

$$L_{i,j} = H_i \cap T_{i+j}.$$  

(4.5)

This value will be used to get an important metric of the availability calculation complexity.

The state and the availability metric are also defined for connections. As already stated, we have assumed that the state of the connection depends
only on the atomic states of its components. In other words, the state of a foreign component does not influence the state of the connection, formally:

\[ \forall e \notin C : P(S(\text{conn})|S(e) = 1) = P(S(\text{conn})|S(e) = 0). \]

Note, that this assumption constrains the scope of the work since many protection strategies, that share resources, do not fulfill this requirement. Although the \( p \)-cycle protection also shares backup resources, the set of protected links, for which the \( p \)-cycle provides backup resources, are well limited: these protected links are the on-cycle and straddling links – depending on the applied priority strategy. However, all these protected links, which share their backup resources with the connection and thus may influence the state of the connection, already belong to set \( C \) [13]. This way, using the \( p \)-cycle protection scheme, we can evaluate the availability of the connections separately. This is an important remark since the results of this work were applied first of all onto \( p \)-cycle protected networks.

Examining a definite connection \( \text{conn} \), the following probability events are important:

- \( a_i = (S(w_i) = 1) \), the working path link \( w_i \) is operational.
- \( \overline{a_i} = (S(w_i) = 0) \), the complementary event of \( a_i \): the working path link \( w_i \) is down.
- \( b_{i,j} = (S(p_{i,j}) = 1) \), the link \( p_{i,j} \) of the protection \( \text{Prot}_i \) is operational.
- \( B_i = (S(p_{i,1}) = 1 \cdot S(p_{i,2}) = 1 \cdot \cdots \cdot S(p_{i,J_i}) = 1) \), the whole protection of link \( w_i \) (i.e., \( \text{Prot}_i \) ) is operational. \( B_i = \prod_{j=0}^{J_i} b_{i,j} \).
- \( X^H_i \) expresses that the request head from \( v_0 \) to \( v_i \) is available \((S(\text{conn}^H_i) = 1)\). Consequently, \( X^H_n \) expresses that the whole connection is available.
- \( X^T_i \) expresses that the request tail from \( v_i \) to \( v_n \) is available. Consequently, \( X^T_0 \) expresses that the whole connection is available.

In the formulae we will also use the complementary events \( \overline{b_{i,j}} \) and \( \overline{X^H_i} \), which may be derived similarly to \( \overline{a_i} \).

### 4.2.3 Mathematical formulation

In the calculations we will use the following equations of probability theory:

The definition of the **conditional probability**:

\[
P(AB) = P(A) \cdot P(B|A)
\]  \hspace{1cm} (4.6)

The **marginal probability** rule:

\[
P(B) = P(A) \cdot P(B|A) + P(\overline{A}) \cdot P(B|\overline{A})
\]  \hspace{1cm} (4.7)
Extended by an additional condition \( G \) we get:

\[
P(B|G) = P(A|G) \cdot P(B|A,G) + P(A^c|G) \cdot P(B|A^c,G)
\]

(4.8)

The multidimensional probability decomposition rule:

\[
P(A_1, A_2, A_3) = P(A_1|A_2, A_3) \cdot P(A_2|A_3) \cdot P(A_3)
\]

(4.9)

Bayes’ theorem for non-vanishing \( P(F) \) probabilities:

\[
P(E|F) = \frac{P(F|E) \cdot P(E)}{P(F)}
\]

(4.10)

Extending Eq. (4.10) by an additional condition \( G \) \( (P(F,G) > 0) \) we will also use

\[
P(E|F,G) = P(F|E,G) \cdot \frac{P(E|G)}{P(F|G)}
\]

(4.11)

In the remaining part of this chapter we will often use these transformations without checking the conditions whether they are zero probability events or not. Practically, since we deal with network link availabilities, both the availability and the unavailability of the network devices are nonzero, however, to fix this shortage of the calculations, we can use

\[
P(F) = 0 \quad \Rightarrow \quad P(E|F) = 1
\]

(4.12)

instead of the transformation.

As the working path of the connection is a sequence of links, it is desirable to express its availability also in a sequence. How can we originate the availability of the connection in the availability of a shorter connection? Applying Eq. (4.7) to our case we can split \( P(X_i^H) \) into two parts depending on the key element (see Key Element Method [95]) \( a_i \) and we get:

\[
P(X_i^H) = P(a_i) \cdot P(X_i^H|a_i) + P(a_i^c) \cdot P(X_i^H|a_i^c)
\]

(4.13)

The first member on the right side in Eq. (4.13) is easy to transform:

\[
P(X_i^H|a_i) = P(X_{i-1}^H|a_i)
\]

since Eq. (4.8) states that

\[
P(X_i^H|a_i) = P(X_{i-1}^H|a_i) \cdot P(X_i^H|X_{i-1}^H,a_i) + P(X_{i-1}^H|a_i) \cdot P(X_i^H|X_{i-1}^H,a_i),
\]

however, \( X_i^H \) is a subevent of \( X_{i-1}^H \), thus

\[
P(X_i^H|X_{i-1}^H,a_i) = 0.
\]

Moreover,

\[
P(X_i^H|X_{i-1}^H,a_i) = 1
\]
as the two events in the conditions, the connection till the last but one link is available \((X^H_{i-1})\) and also the last link on the working path \((a_i)\), together determine that \(X^H_i\) is also available.

Regarding the second member on the right side in Eq. (4.13), in case of event \(\overline{w_i}\), the protection of \(w_i\) must be activated to make \(X^H_i\) also available after \(X^H_{i-1}\) is available. This requires all the links in \(Prot_i\) to be operational simultaneously, which is expressed by event \(B_i\):

\[
P(X^H_i | \overline{a_i}) = P(X^H_{i-1}, B_i | \overline{a_i}).
\] (4.14)

We use the definition of the conditional probability Eq. (4.6) – extended with the \(a_i\) condition – to transform the right hand side of the equation:

\[
P(X^H_i | \overline{a_i}) = P(X^H_{i-1} | \overline{a_i}) \cdot P(B_i | X^H_{i-1}, \overline{a_i}).
\] (4.15)

At this point our original equation, expressing the availability of \(conn^H_i\), looks like

\[
P(X^H_i) = P(a_i) \cdot P(X^H_{i-1} | a_i) + P(\overline{a_i}) \cdot P(X^H_{i-1} | \overline{a_i}) \cdot P(B_i | X^H_{i-1}, \overline{a_i}).
\] (4.16)

Note, there are no \(X^H_i\) events on the right side of the equation, this way we managed to trace back the probability of \(X^H_i\) on \(X^H_{i-1}\).

However, we do not want to take any \(X^H_i\) events as condition, so applying Eq. (4.11) on the last member of Eq. (4.15) we get

\[
P(B_i | X^H_{i-1}, \overline{a_i}) = P(X^H_{i-1} | B_i, \overline{a_i}) \cdot \frac{P(B_i | \overline{a_i})}{P(X^H_{i-1} | \overline{a_i})}
\] (4.17)

where the formula on the right side is far more convenient than the left side as there are only link and link segment availability events – \(\overline{a_i}\) and \(B_i\) – among the conditions.

Explicating the event compound \(B_i = b_{i,1} \cdot b_{i,2} \cdot \ldots \cdot b_{i,J_i}\) results in

\[
P(X^H_{i-1} | B_i, \overline{a_i}) = P(X^H_{i-1} | b_{i,1}, b_{i,2}, \ldots, b_{i,k}, \overline{a_i})
\]

and as the \(b_{i,j}\) events are independent:

\[
P(B_i | \overline{a_i}) = \prod_{j=1}^{J_i} P(b_{i,j} | \overline{a_i}).
\]

We express the right hand side of Eq. (4.17):

\[
P(X^H_{i-1} | B_i, \overline{a_i}) \cdot \frac{P(B_i | \overline{a_i})}{P(X^H_{i-1} | \overline{a_i})} = P(X^H_{i-1} | b_{i,1}, b_{i,2}, \ldots, b_{i,k}, \overline{a_i}) \cdot \prod_{j=1}^{J_i} P(b_{i,j} | \overline{a_i})
\] (4.18)
and apply it to Eq. (4.17):

\[ P(B_i | X_{i-1}^H, \overline{a_i}) = \frac{P(X_{i-1}^H | b_{i,1}, b_{i,2}, \ldots, b_{i,k}, \overline{a_i})}{P(X_{i-1}^H | \overline{a_i})} \cdot \prod_{j=1}^{J_i} P(b_{i,j} | \overline{a_i}) \]  

(4.19)

After substituting Eq. (4.19) into Eq. (4.16) we get

\[ P(X_i^H) = P(a_i) \cdot P(X_{i-1}^H | a_i) + P(\overline{a_i}) \cdot P(X_{i-1}^H | \overline{a_i}) \cdot \frac{P(X_{i-1}^H | b_{i,1}, b_{i,2}, \ldots, b_{i,k}, \overline{a_i})}{P(X_{i-1}^H | \overline{a_i})} \cdot \prod_{j=1}^{J_i} P(b_{i,j} | \overline{a_i}), \]  

(4.20)

or simplified,

\[ P(X_i^H) = P(a_i) \cdot P(X_{i-1}^H | a_i) + P(\overline{a_i}) \cdot P(X_{i-1}^H | b_{i,1}, b_{i,2}, \ldots, b_{i,k}, \overline{a_i}) \cdot \prod_{j=1}^{J_i} P(b_{i,j} | \overline{a_i}) \]  

(4.21)

which traces back the availability of \( X_i^H \) to the availability of some links and \( X_{i-1}^H \).

To be able to use Eq. (4.21) recursively, we have to define how to calculate \( P(X_{i-1}^H | a_i) \), etc. or, more generally, \( P(X_i^H | G) \) for a general \( G \) condition:

\[ P(X_i^H | G) = P(a_i | G) \cdot P(X_{i-1}^H | a_i, G) + P(\overline{a_i} | G) \cdot P(X_{i-1}^H | b_{i,1}, b_{i,2}, \ldots, b_{i,k}, \overline{a_i}, G) \cdot \prod_{j=1}^{J_i} P(b_{i,j} | \overline{a_i}, G) \]  

(4.22)

The terminating equation must be defined also:

\[ P(X_0^H | G) = 1 \]  

(4.23)

expressing the fact that the connection in the source node is always (i.e., at any \( G \) condition) available.

### 4.2.4 Calculation in practice

One can see that Eq. (4.22) refers 2 times to \( P(X_{i-1}^H) \) with different \( G \) conditions. For a connection of length \( n \) that means a recursive calculation with the depth of \( n \) and a branching factor of 2 at each recursion level. Altogether, theoretically, we need \( \sum_{k=0}^{n-1} 2^k \), i.e., \( 2^{n+1} - 1 \) instances of \( P(X_i^H | G) \) evaluation to get the accurate availability metric for a \( p \)-cycle protected connection. This amount of additional calculation requirement, compared
to the $O(n)$ steps of the $S$-$P$ method, would make the accurate availability evaluation rather unattractive.

However, some of the $P(X^H_{i-1}|G)$ results are not needed because they have a $\prod_{j=1}^{l} P(b_{i,j}|G)$ coefficient that equals to 0 since the general resolution of conditional link availability probabilities for link state event $ev$ with condition $G$ is:

$$P(ev|G) = \begin{cases} 0 & \text{if } ev \supseteq G, \\ 1 & \text{if } ev \supseteq G, \\ P(ev) & \text{otherwise.} \end{cases}$$

In the calculation tree, those $P(X^H_i|G)$ branches which are multiplied by 0 should not be unfolded and evaluated. This happens in connections where the protection path of a working link overlaps working links (there are overlapping default and protection links): $\exists i_0, i_1, j$ so that $p_{i_0,j} = w_{i_1}$. This implies $a_{i_1} = b_{i_0,j}$.

This is the first reason why there are less calculations needed in practice. The second reason is the identity of two events. There are many cases when events $X|G_1$ and $X|G_2$ with different $G_1$ and $G_2$ conditions are identical.

Examining the recursion formula Eq. (4.22) and Eq. (4.23) we can assume that the conditional probabilities that we have to calculate to get $P(X^H_i)$ consist only of such atomic events which are related to a link that is a member of $H_i$. In other words,

$$\forall e \notin H_i : \quad P(X^H_i) = P(X^H_i|S(e) = 1) = P(X^H_i|S(e) = 0),$$

expressing that any link that is not involved in the connection has no impact on the availability of the connection.

This way, the $G$ conditions can be simplified: $P(X^H_i|G) = P(X^H_i|G')$ where $G'$ is a supercondition of $G$ formed only of those $a_{i}, \overline{a_{i}}, b_{i,j}$ events for which the corresponding $w_{i}, p_{i,j}$ links are members of $H_i$. Formally, we define $G'$ so that

$$G' = \prod_{\forall e \in H_i: (S(e) = 1)} (S(e) = 1) \cdot \prod_{\forall e \in H_i: (S(e) = 0)} (S(e) = 0).$$

As the $G'$ event is a superset of $G$, it may happen – and practically it happens very often – that the probabilities $P(X|G_1)$ and $P(X|G_2)$ with different $G_1$ and $G_2$ conditions can be replaced by the same $P(X|G')$. These replacements reduce the branching factor of the given recursion level.

To sum up, theoretically the calculation complexity is $O(2^n)$ with connection length $n$, however, because of the formerly mentioned two evaluation reductions, the availability calculation complexity becomes much lower in practice.

80
4.2.5 Calculation complexity

Illustrative examples demonstrating the incremental availability calculation are available in [13]. In the examples we have seen that the calculation complexity to get the accurate availability is far less than the roughly defined theoretical $O(2^n)$. Furthermore, this upper bound makes the presented calculation method unattractive for long connections. In the followings we present an upper bound of calculation complexity which is closer to the real values and does not penalize long connections with a factor exponential to the length of the connection.

Whenever we have to evaluate a $P(X^H_i|G')$ availability, condition $G'$ contains only such events which refer to links $e \in H_i \cap T_i = L_{i,0}$, since

- any link out of $H_i$ is irrelevant due to the formerly discussed $G \rightarrow G'$ transformation,
- the recursion starts at the tail of the connection (at node $v_n$) and it is expressed by $P(X^H_{n-1}|G)$ probabilities, where $G$ may bind the state of links that are present in $T_{n-1}$. Going deeper in the recursion, at level $i$, the $G$ conditions of the $P(X^H_i|G)$ probabilities may refer only to those links that are already used at the tail of the connection. These links are enumerated in $T_i$.

For the whole connection we define $L_{max}$ as the furthest overlap. This metric can be derived from the $L_{i,j}$ sets. Note that – due to the inequalities Eq. (4.2) and Eq. (4.3) – by increasing $j$ the cardinality of $L_{i,j}$ decreases.

We have to find the highest $j$ value for which there can be found any non-empty $L_{i,j}$ set:

$$L_{max} = \max_i (\arg\max_j (L_{i,j} \neq \emptyset))$$  \hspace{1cm} (4.26)

This is an important metric of the connection, since the furthest overlap also defines a constraint for the branching: we will not need more than $2^{L_{max}+1}$ different conditions at any recursion level. This can be explained as follows.

In the simplest case, if $L_{i,0} = \emptyset$, the conditional availabilities $P(X^H_i|G)$ of any $G$ condition can be reduced to $P(X^H_i)$. The next case to examine is the case when $L_{i,0} \neq \emptyset$ but $L_{i,1} = \emptyset$. We can express by means of Eq. (4.22) any $P(X^H_{i+1}|G)$ referring to $P(X^H_i|a_{i+1},G)$ and $P(X^H_i|b_{i+1,1},b_{i+1,2},...,b_{i+1,k},\overline{a_{i+1}},G)$. Normally, that transformation would double the number of conditional probabilities to evaluate. However, in the latter probabilities any $G$ condition can be eliminated, since $G$, as it is inherited from the previous ($i + 1^{th}$) recursion level, may contain only conditions referring to links of $L_{i+1,0}$. But we have assumed that $L_{i,1} = \emptyset$, hence none of the links referred to by $G$ is member of $L_{i,0}$. So any $P(X^H_i|a_{i+1},G)$ will be reduced to $P(X^H_i|a_{i+1})$ and any $P(X^H_i|b_{i+1,1},b_{i+1,2},...,b_{i+1,k},\overline{a_{i+1}},G)$ will be reduced
to $P(X_i^H | b_{i+2}, b_{i+3}, \ldots, b_{i+k}, u_{i+2}).$ Finally, we will have to evaluate only these two different conditional probabilities of $X_i^H$.

This reasoning can be continued to any $j$ distance where $L_{i,j} \neq \emptyset$ but $L_{i,j+1} = \emptyset$. In that case, at recursion level $i$, we need to evaluate at most $2^{j+1}$ different $P(X_i^H | G)$ branches. Assuming the worst case scenario, when $\forall i : L_{i,L_{\text{max}}} \neq \emptyset$, at each recursion level there will be $2^{L_{\text{max}}+1}$ different $P(X_i^H | G)$ branches. That makes altogether $n \cdot 2^{L_{\text{max}}+1}$ different branches for the worst case scenario. This is the upper bound of the calculation steps. This way, a more accurate complexity can be defined on the exact availability calculation: $O(n \cdot 2^{L_{\text{max}}+1})$.

4.3 The approximation error of the Serial-Parallel calculation model

Although the Serial-Parallel calculation method offers a very fast way for retrieving connection availability metric from basic link availability metrics, (having $n$ links its complexity is $O(n)$), it does not take into account the link overlaps, i.e., those cases when some links are common in different series. This shortcoming of the algorithm may lead to approximation error. In the followings we examine the error of the S-P method by means of geometrical representation of probabilities.

For the availability of the connection and for the availability approximation we use the following simplifications of notations:

- $A_{\text{ACC}}$ is the exact (accurate) availability estimation of the connection ($A_{\text{ACC}} = P(S(\text{conn}) = 1), A_{\text{ACC}} = P(X_i^H)$). We also use $U_{\text{ACC}} = 1 - A_{\text{ACC}}$ as the corresponding accurate unavailability.
- $A_{\text{ACC}}^H_i$ is the exact availability of the connection part $\text{conn}_i^H$ ($A_{\text{ACC}}^H_i = P(S(\text{conn}_i^H) = 1)$ or simply $A_{\text{ACC}}^H_i = P(X_i^H)$). The corresponding unavailability is denoted by $U_{\text{ACC}}^H_i = 1 - A_{\text{ACC}}^H_i$. Similarly, $A_{\text{ACC}}^T_i = P(X_i^T)$ and $U_{\text{ACC}}^T_i = 1 - A_{\text{ACC}}^T_i$.
- $A_{\text{SP}}$ (with the complementary $U_{\text{SP}} = 1 - A_{\text{SP}}$) is the availability of the connection estimated by the S-P method. $A_{\text{SP}}^H_i$ and $A_{\text{SP}}^T_i$ are the S-P estimated availabilities of $\text{conn}_i^H$ and $\text{conn}_i^T$, respectively.
- $A_{\text{SPX}}_i$ is the “mixed” availability of the connection. It assumes that the exact availability of $\text{conn}_i^H$ and $\text{conn}_i^T$ are known and applies the S-P method on these known availabilities ($A_{\text{SPX}}_i = A_{\text{ACC}}^H_i \cdot A_{\text{ACC}}^T_i$).
- $A_{\text{ACC}} | G$ is the conditional availability of the connection for any $G$ condition ($A_{\text{ACC}} | G = P(S(\text{conn}) = 1 | G)$). $A_{\text{ACC}}^H_i | G$, $A_{\text{SP}}^H_i | G$, etc. are defined in similar manner.
To express the difference between the exact and the estimated connection availabilities, we define

\[ \text{Diff} = A_{\text{ACC}} - A_{\text{SP}}, \quad (4.27) \]

\[ \text{Diff}^*_i = A_{\text{ACC}} - A_{\text{SPX}_i}. \quad (4.28) \]

As both the real and the approximated connection availability metrics are usually \( \approx 1 \) values, the quotient of the two metrics is also approximately 1 or near to 1, which does not express significantly the error of the S-P method. Hence, we define the relative accuracy of the S-P method as the quotient of the real and the approximated unavailabilities:

\[ \text{DIV}_U = \frac{U_{\text{ACC}}}{U_{\text{SP}}}, \quad (4.29) \]

\[ \text{DIV}^*_U_i = \frac{U_{\text{ACC}}}{U_{\text{SPX}_i}}. \quad (4.30) \]

As Eq. (4.1) stated, if we take head and tail parts of the connection at node \( v_i \), their concatenation constructs the original connection:

\[ \forall i : \text{conn} = \text{conn}^H_i . \text{conn}^T_i. \]

Moreover, also the state of the connection can be expressed by means of the states of the connection parts: The connection is up if and only if both the head and the tail parts are up:

\[ S(\text{conn}) = 1 \iff S(\text{conn}^H_i) = 1 \land S(\text{conn}^T_i) = 1 \]

\[ (X^H_n \iff X^T_0) \iff X^H_i \cdot X^T_i. \]

Still, concerning the availability,

\[ A_{\text{ACC}} \neq A_{\text{ACC}}^H_i \cdot A_{\text{ACC}}^T_i, \]

except for cases when \( S(\text{conn}^H_i) \) and \( S(\text{conn}^T_i) \) are independent, meaning that \( H_i \cap T_i = L_{i,0} = \emptyset \).

In fact, this is where the approximation error of the S-P method comes from, since it uses the heuristic

\[ A_{\text{SP}} = A_{\text{SP}}^H_i \cdot A_{\text{SP}}^T_i, \quad (4.31) \]

\[ A_{\text{SPX}_i} = A_{\text{ACC}}^H_i \cdot A_{\text{ACC}}^T_i, \quad (4.32) \]

and only if \( S(\text{conn}^H_i) \) and \( S(\text{conn}^T_i) \) are independent:

\[ A_{\text{ACC}} = A_{\text{SPX}_i} = A_{\text{ACC}}^H_i \cdot A_{\text{ACC}}^T_i = P(X^H_i) \cdot P(X^T_i). \quad (4.33) \]

This differs from the exact availability formula which does not require the independency of \( S(\text{conn}^H_i) \) and \( S(\text{conn}^T_i) \). The formula, according to Eq. (4.6), is:

\[ A_{\text{ACC}} = P(X^H_i \cdot X^T_i) = P(X^H_i) \cdot P(X^T_i | X^H_i) \]

\[ ( = P(X^T_i) \cdot P(X^H_i | X^T_i) ). \]

83
Note that if $\text{conn}_i^H$ and $\text{conn}_i^T$ do not contain any overlapping link, their exact and estimated availabilities are identical:

$$A_{\text{ACC}}^H = A_{\text{SP}}^H, \quad A_{\text{ACC}}^T = A_{\text{SP}}^T. \quad (4.35)$$

In the followings, we will analyze the relation between $A_{\text{ACC}}$ and $A_{\text{SPX}}$. In the analysis we will show figures (Fig. 4.4, 4.5, 4.7) illustrating the availability in two-dimensional probability square. The two dimensions of these squares correspond to the state and availability of $\text{conn}_i^H$ (horizontal dimension) and $\text{conn}_i^T$ (vertical dimension). By means of these probability squares, the availability of the whole connection can be evaluated merely by accumulating the rectangular areas where both of $\text{conn}_i^H$ and $\text{conn}_i^T$ are available. We will see that the only difference between evaluating $A_{\text{ACC}}$ and $A_{\text{SPX}}$ is that in case of $A_{\text{ACC}}$ in the dimension of $\text{conn}_i^T$, according to Eq. (4.34), we will use different conditional availability values depending on the state of the non-independent (i.e., overlapping) links.

We have the same assumptions against the network and protection model as described in Sect. 4.2.1. To derive valuable conclusion about the approximation error of the S-P method, it is important to take into account the monotonicity assumption (see Sect. 4.2.1), i.e., the state of the connection is a monotonic function of the state of the links. In other words, improving the state of any link from down to up state cannot deteriorate the state of the connection, and taking any link into down state cannot result in repaired connection.

4.3.1 Single link overlap

First, we examine a simple connection in which there is a single link overlap between $\text{conn}_i^H$ and $\text{conn}_i^T$. The common link is denoted by $e_{h^*} = e_{t^*}$.

Figure 4.4: Visual representation of the estimated and the exact availability and their difference
Figure 4.4 compares the availability got by the S-P method \((A_{SPX_i})\) to the exact availability \((A_{ACC})\). For the sake of simplicity we use the following notation:

\[
A = P(S(e_{h^*}) = 1) = P(S(e_{t^*}) = 1) \quad \text{the availability metric of the examined (overlapping) link.}
\]

\[
U = P(S(e_{h^*}) = 0) = P(S(e_{t^*}) = 0) \quad \text{the unavailability metric of the examined link.} \quad U = 1 - A.
\]

\[
x_1 = P(S(conn^{H}_{e}) = 0 | S(e_{h^*}) = 1) \quad \text{is the down state probability of} \ conn^{H}_{e} \ \text{in case of link} \ e_{h^*} \ \text{is up.}
\]

\[
x_2 = P(S(conn^{T}_{e}) = 0 | S(e_{h^*}) = 0) \quad \text{is the probability that} \ conn^{T}_{e} \ \text{is down in case of link} \ e_{h^*} \ \text{is down.}
\]

\[
y_1 = P(S(conn^{H}_{e}) = 0 | S(e_{t^*}) = 1) \quad \text{is the probability that} \ conn^{H}_{e} \ \text{is down in case of link} \ e_{t^*} \ \text{is up.}
\]

\[
y_2 = P(S(conn^{T}_{e}) = 0 | S(e_{t^*}) = 0) \quad \text{is the probability that} \ conn^{T}_{e} \ \text{is down in case of link} \ e_{t^*} \ \text{is down.}
\]

We introduce two derived metrics which help us to express the difference between \(A_{SPX_i}\) and \(A_{ACC}\):

\[
d_x = x_2 - x_1,
\]

\[
d_y = y_2 - y_1,
\]

indicating how much does \(conn^{H}_{e}\) depend on \(e_{h^*}\) and \(conn^{T}_{e}\) on \(e_{t^*}\).

We can assume that \(x_1 \leq x_2 \) and \(y_1 \leq y_2 \) (implying \(d_x, d_y \geq 0 \)) due to the monotonicity assumption (if a link is down, the connection unavailability is always higher (or equal) than in case the given link is up).

Figure 4.4 also shows that the availability probability metrics \((A_{ACC}, A_{SPX_i})\) of the connection can be calculated in a geometric way:

\[
A_{ACC} = A(1 - x_1)(1 - y_1) + U(1 - x_2)(1 - y_2), \quad (4.36)
\]

\[
A_{SPX_i} = \left(A(1 - x_1) + U(1 - x_2)\right) \cdot \left(A(1 - y_1) + U(1 - y_2)\right), \quad (4.37)
\]

and how much is the difference between them:

\[
Diff^*_{i} = A_{ACC} - A_{SPX_i} = Diff^\oplus_{i} - Diff^\odot_{i} \quad (4.38)
\]

This difference is of high importance in our analysis, so we express it:

\[
Diff^\oplus_{i} = A(y_2 - y_1) \cdot U(1 - x_1) \quad (4.39)
\]

\[
Diff^\odot_{i} = U(y_2 - y_1) \cdot A(1 - x_2) \quad (4.40)
\]

resulting in

\[
Diff^*_{i} = Diff^\oplus_{i} - Diff^\odot_{i} = AU(y_2 - y_1)(x_2 - x_1) = AU d_x d_y \quad (4.41)
\]
Note that $\text{Diff}_i^*$ is non-negative as all of its coefficients are also non-negative.

This way, for single link overlap we have proven that the estimated availability is “conservative”, meaning that it is never greater than the exact availability.

The next step is to find the maximal amount of the error. The difference of unavailabilities is the inverted difference of availabilities. By transforming Eq. (4.28) we get:

$$U_{\text{ACC}} = U_{\text{SPX}i} - \text{Diff}_i^*, \quad (4.42)$$

Substituting Eq. (4.42) into Eq. (4.30) results in

$$DIV_{U_i^*} = \frac{U_{\text{SPX}i} - \text{Diff}}{U_{\text{SPX}i}} = 1 - \frac{\text{Diff}}{U_{\text{SPX}i}}. \quad (4.43)$$

We already know that $DIV_{U_i^*} \leq 1$. Now we define a lower bound for $DIV_{U_i^*}$. We state that

$$DIV_{U_i^*} \geq 1 - \frac{d_x d_y}{d_x + d_y} \quad (4.44)$$

and in the followings we will prove it.

Equation (4.44) is equivalent with

$$\frac{\text{Diff}_i^*}{U_{\text{SPX}i}} \leq \frac{d_x d_y}{d_x + d_y}, \quad (4.45)$$

which can be transcribed into

$$\text{Diff}_i^*(d_x + d_y) \leq U_{\text{SPX}i} d_x d_y \quad (4.46)$$

$$AU d_x d_y (d_x + d_y) \leq U_{\text{SPX}i} d_x d_y \quad (4.47)$$

Relation Eq. (4.47) is confirmed by the geometric representation of $U_{\text{SPX}i}$. First, we introduce a lower bound of $U_{\text{SPX}i}$.

Figure 4.5 shows an area $U_{\text{SPX}i}^{\text{low}}$.

$$U_{\text{SPX}i}^{\text{low}} = AU y_2 + AU x_2 = AU (x_2 + y_2), \quad (4.48)$$

which is a part of $U_{\text{SPX}i}$, thus

$$U_{\text{SPX}i}^{\text{low}} < U_{\text{SPX}i}. \quad (4.49)$$

Next, applying the inequalities $d_x \leq x_2$, $d_y \leq y_2$, we can write:

$$AU (d_x + d_y) \leq AU (x_2 + y_2). \quad (4.50)$$
Putting together Eq. (4.50), Eq. (4.48) and Eq. (4.49), respectively, we get
\[ AU(d_x + d_y) \leq AU(x_2 + y_2) = U_{SPX_i}^{low} < U_{SPX_i}, \] (4.51)
where we find exactly Eq. (4.47) at the left and the right end. Therefore, Eq. (4.44) is also always true.

Putting together the lower and the upper bound estimations we get
\[ 1 - \frac{d_x d_y}{d_x + d_y} \leq DIV_{U_i^*} \leq 1. \] (4.52)

We get rougher but simpler lower bound estimation by exploiting that – analogously to the aggregate resistance of parallel switched devices –
\[ \frac{d_x d_y}{d_x + d_y} \leq \min(d_x, d_y) \quad \text{and} \quad \min(d_x, d_y) \leq \min(x_2, y_2) \]
\[ 1 - \min(x_2, y_2) \leq DIV_{U_i^*} \leq 1. \] (4.53)

Recalling the meaning of \( x_2 \) and \( y_2 \), Eq. (4.53) formalizes that the exact unavailability of the connection is not only lower than the approximated value, but these values are also close to each other: The deviation between the exact and the approximated connection unavailables is never worse than the unavailability of the more viable connection-part (head or tail of the connection) in case the overlapping link is down. And – as the connection is protected – even in case the overlapping link is down, the availability of the more viable connection-part is still high. Its unavailability is in order of magnitude of an unprotected link. Thus:
\[ \lim_{\forall e \in E; P(S(e) = 1) \to 1} DIV_{U_i^*} = 1, \] (4.54)
expressing that the more the links get available the more accurate the approximation will be.

Figure 4.6 illustrates how near is \( DIV_{U_i^*} \) to its lower and upper bound estimations. In the figures we can see scenarios of different \( U, x_1, y_1, x_2 \) and
Table 4.2: Intervals of variables in certain scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$U$</th>
<th>$x_1$</th>
<th>$y_1$</th>
<th>$x_2$</th>
<th>$y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1.a</td>
<td>(0, $10^{-1}$)</td>
<td>(0, $10^{-4}$)</td>
<td>(0, $10^{-4}$)</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-2}$)</td>
</tr>
<tr>
<td>Scenario 1.b</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-4}$)</td>
<td>(0, $10^{-4}$)</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-2}$)</td>
</tr>
<tr>
<td>Scenario 1.c</td>
<td>(0, $10^{-3}$)</td>
<td>(0, $10^{-4}$)</td>
<td>(0, $10^{-4}$)</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-2}$)</td>
</tr>
<tr>
<td>Scenario 2.a</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-4}$)</td>
<td>(0, $10^{-4}$)</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-2}$)</td>
</tr>
<tr>
<td>Scenario 2.b</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-4}$)</td>
<td>(0, $10^{-3}$)</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-2}$)</td>
</tr>
<tr>
<td>Scenario 2.c</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-3}$)</td>
<td>(0, $10^{-3}$)</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-2}$)</td>
</tr>
<tr>
<td>Scenario 2.d</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-3}$)</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-2}$)</td>
</tr>
<tr>
<td>Scenario 3.a</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-4}$)</td>
<td>(0, $10^{-4}$)</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-2}$)</td>
</tr>
<tr>
<td>Scenario 3.b</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-4}$)</td>
<td>(0, $10^{-4}$)</td>
<td>(0.5 $10^{-3}$)</td>
<td>(0, $10^{-1}$)</td>
</tr>
<tr>
<td>Scenario 3.c</td>
<td>(0, $10^{-2}$)</td>
<td>(0, $10^{-4}$)</td>
<td>(0, $10^{-4}$)</td>
<td>(0.5 $10^{-3}$)</td>
<td>(0.5 $10^{-1}$)</td>
</tr>
</tbody>
</table>
values. In each scenario the variables take their values uniformly from a predefined domain, as shown in Table 4.2.

Figure 4.6(a) shows 3 scenarios. If the domain of variable \(\mathbf{U}\) is smaller and closer to 0, \(DIV_{U_i}^*\) is closer to its upper bound. We observe the same tendency if the \(x\) values get comparable (comparably high) to the \(y\) values (see Fig. 4.6(b)). This behavior corresponds to the fact that the difference between the \(y\) and \(x\) values (i.e., \(d_y\) and \(d_x\)) are getting low. By increasing one of the \(x_2\) or \(y_2\) (in our case \(y_2\)), there will be more \(DIV_{U_i}^*\) values near the lower bound (Fig. 4.6(c)).

4.3.2 Approximation error of the Serial-Parallel calculation model in case of two overlapping links

In case of multiple overlapping links we want to get answer to the following questions:

- Is \(DIV_{U_i}^* \leq 1\)?
- Can we find a lower bound for \(DIV_{U_i}^*\) similar to the single link overlap scenario?
- we have proven that \(DIV_{U_i}^* \to 1\) if all the links are protected and the link availability \(\to 1\) \((\lim_{\forall e: P(S(e) = 1) \to 1} DIV_{U_i}^* = 1)\). Where does \(DIV_{U_i}^*\) converge in case of multiple overlaps?

In this section we examine a connection that contains two overlapping links between \(conn^H_i\) and \(conn^T_i\).

We recall Fig. 4.4 and extend it first to dual overlap scenario (with two overlapping edges: \(e_h^\cdot = e_t^\cdot\) and \(e_h^\circ = e_t^\circ\)). In Fig. 4.7 we see a general case with the following notation:

\[
A_1 = P(S(e_h^\cdot) = 1) = P(S(e_t^\cdot) = 1) \quad \text{– the availability of the first overlapping link.}
\]

\[
U_1 = P(S(e_h^\cdot) = 0) = P(S(e_t^\cdot) = 0) \quad \text{– the unavailability of the first overlapping link.}
\]

\[
A_2 = P(S(e_h^\circ) = 1) = P(S(e_t^\circ) = 1) \quad \text{– the availability of the second overlapping link.}
\]

\[
U_2 = P(S(e_h^\circ) = 0) = P(S(e_t^\circ) = 0) \quad \text{– the unavailability of the second overlapping link.}
\]

\[
x_1 = P(S(conn^H_i) = 0 | S(e_h^\cdot) = 1, S(e_h^\circ) = 1).
\]

\[
x_2 = P(S(conn^H_i) = 0 | S(e_h^\cdot) = 1, S(e_h^\circ) = 0).
\]

\[
x_3 = P(S(conn^H_i) = 0 | S(e_h^\cdot) = 0, S(e_h^\circ) = 1).
\]

\[
x_4 = P(S(conn^H_i) = 0 | S(e_h^\cdot) = 0, S(e_h^\circ) = 0).
\]

\[
y_1 = P(S(conn^T_i) = 0 | S(e_t^\cdot) = 1, S(e_t^\circ) = 1).
\]

\[
y_2 = P(S(conn^T_i) = 0 | S(e_t^\cdot) = 1, S(e_t^\circ) = 0).
\]
Figure 4.7: Visual representation of the approximation error of the $S$-$P$ method in case of double link overlap

\[ y_3 = P(S(\text{conn}^T_i) = 0 | S(e^*_{i'}) = 0, S(e^*_{i''}) = 1), \]
\[ y_4 = P(S(\text{conn}^T_i) = 0 | S(e^*_{i'}) = 0, S(e^*_{i''}) = 0). \]

Additionally, for each $l \neq j$ pair, we will denote

\[ d_{xlj} = x_l - x_j, \text{ and} \]
\[ d_{ylj} = y_l - y_j. \]

Due to the monotonicity assumption, the following inequalities are true regarding the $x$ and $y$ values:

\[ x_1 \leq x_2 \leq x_4; \quad x_1 \leq x_3 \leq x_4 \]
\[ y_1 \leq y_2 \leq y_4; \quad y_1 \leq y_3 \leq y_4. \]

(4.55)
These inequalities restrict the range of some $d$ values:

$$\forall (1 \leq l \leq 4) \forall (1 \leq j \leq 4) : l = 4 \lor j = 1 \Rightarrow d_{xlj} \geq 0 \land d_{ylj} \geq 0.$$  

Nevertheless, we must emphasize that we cannot state neither $d_{x32} \geq 0$ nor $d_{y32} \geq 0!$ And this absence makes the examination difficult.

**Bounds of the approximation error**

First, we want to answer the question whether $DIV_{U^*_i} \leq 1$. To do this, we have to collect the pieces of $Diff^\oplus_i$ and $Diff^\ominus_i$. They are:

$$Diff^\oplus_i = D_1^\oplus + D_2^\oplus + D_3^\oplus + D_4^\ominus + D_5^\oplus + D_6^\oplus$$

$$= A_1 U_2 d_{y21} \times A_1 A_2 (1 - x_1) +$$

$$+ U_1 A_2 d_{y31} \times A_1 A_2 (1 - x_1) +$$

$$+ U_1 U_2 d_{y41} \times A_1 A_2 (1 - x_1) +$$

$$+ U_1 A_2 d_{y32} \times A_1 U_2 (1 - x_2) +$$

$$+ U_1 U_2 d_{y42} \times A_1 U_2 (1 - x_2) +$$

$$+ U_1 U_2 d_{y43} \times U_1 A_2 (1 - x_3)$$  \hspace{1cm} (4.56)

$$Diff^\ominus_i = D_1^\oplus + D_2^\oplus + D_3^\oplus + D_4^\ominus + D_5^\oplus + D_6^\ominus$$

$$= A_1 A_2 d_{y31} \times U_1 A_2 (1 - x_3) +$$

$$+ A_1 A_2 d_{y41} \times U_1 U_2 (1 - x_4) +$$

$$+ A_1 U_2 d_{y32} \times U_1 A_2 (1 - x_3) +$$

$$+ A_1 U_2 d_{y42} \times U_1 U_2 (1 - x_4) +$$

$$+ U_1 A_2 d_{y43} \times U_1 U_2 (1 - x_4)$$  \hspace{1cm} (4.57)

After putting Eq. (4.56) and Eq. (4.57) together and replacing $(1 - x_j) - (1 - x_l)$ with $d_{xlj}$ we get:

$$Diff^*_i = Diff^\oplus_i - Diff^\ominus_i =$$

$$= A_1^2 A_2 U_2 d_{y21} d_{x21} + A_1 U_1 A_2^2 d_{y31} d_{x31} +$$

$$+ A_1 U_1 A_2 U_2 (d_{y41} d_{x41} + d_{y32} d_{x32}) +$$

$$+ A_1 U_1 U_2^2 d_{y42} d_{x42} + U_1^2 A_2 U_2 d_{y43} d_{x43}$$  \hspace{1cm} (4.58)

Recalling Eq. (4.30) stating that

$$DIV_{U^*_i} = \frac{U_{ACC}}{U_{SPX_i}} = 1 - \frac{Diff^*_i}{U_{SPX_i}}.$$
we can prove that $DIVU_i^* \leq 1$ by proving that $0 \leq Diff_i^*$.

The only term of Eq. (4.58) which is not evidently non-negative is $A_1U_1A_2U_2(d_{y41}d_{x41} + d_{y32}d_{x32})$, since both $d_{y32}$ and $d_{x32}$ may be negative. Fortunately, as a consequence of Eq. (4.55), for the absolute values of the differences, the relations $|d_{y41}| \geq |d_{y32}|$ and $|d_{x41}| \geq |d_{x32}|$ are true. That way

$$d_{y41}d_{x41} + d_{y32}d_{x32} \geq d_{y41}d_{x41} - |d_{y32}d_{x32}| \geq 0.$$ 

This means that $DIVU_i^* \leq 1$ even in case of dual link overlap. In other words, the $S-P$ method does not overestimate the connection availability.

Note that this inequality is also true for multiple link overlaps (see Sect. 4.3.3).

The next question is how much is the maximal positive divergence of $U_{SPX_i}$, i.e., what is the lower bound of $DIVU_i^*$. We can transform the task of defining a lower bound of $DIVU_i^*$ into the task of defining an upper bound of $Diff_i^* U_{SPX_i}$. We approximate $U_{SPX_i}$ with $U_{SPX_i}^{low}$ which is not greater than $U_{SPX_i}^{low}$.

$$\frac{Diff_i^*}{U_{SPX_i}} \leq \frac{Diff_i^*}{U_{SPX_i}^{low}}$$ (4.59)

Figure 4.7 helps us to unfold $U_{SPX_i}^{low}$:

$$U_{SPX_i}^{low} = A_1^2A_2U_2(x_2 + y_2) + A_1U_1A_2^2(x_3 + y_3) +$$
$$+ A_1U_1A_2U_2(\max(x_3, y_2) + \max(y_3, x_2) + x_4 + y_4) +$$
$$+ A_1U_1U_2^2(x_4 + y_4) + U_1^2A_2U_2(x_4 + y_4)$$ (4.60)

Note that, opposed to the annotated area in Fig. 4.7, we use $\max(x_3, y_2)$ instead of $x_3$ (and $\max(y_3, x_2)$ instead of $y_3$) to maximize $U_{SPX_i}^{low}$ when set against the $Diff_i^*$ metric.

To be able to compare with $Diff_i^*$ we introduce $U_{SPX_i}^{low*} \leq U_{SPX_i}^{low}$ as

$$U_{SPX_i}^{low*} = A_1^2A_2U_2(d_{x21} + d_{y21}) + A_1U_1A_2^2(d_{x31} + d_{y31}) +$$
$$+ A_1U_1A_2U_2(\max(x_3, y_2) + \max(y_3, x_2) + d_{x41} + d_{y41}) +$$
$$+ A_1U_1U_2^2(d_{x42} + d_{y42}) + U_1^2A_2U_2(d_{x43} + d_{y43})$$ (4.61)

Now, first we will give an evident lower bound estimation:

$$\frac{1}{2} \leq DIVU_i^*$$ (4.62)

which can be transformed into

$$\frac{Diff_i^*}{U_{SPX_i}} \leq \frac{1}{2}$$ (4.63)

$$2 \cdot Diff_i^* \leq U_{SPX_i}.$$ (4.64)
In Eq. (4.64) we can find for each term of $\text{Diff}^*_i$ the corresponding term in $U_{\text{SPX}_i}^{\text{low}^*}$ so that (e.g., for the first term):

$$2 \cdot A_2^1A_2U_2d_{y21}d_{x21} \leq A_1^2A_2U_2(d_{x21} + d_{y21})$$

$$d_{y21}d_{x21} + d_{y21}d_{x21} \leq d_{x21} + d_{y21}$$

$$0 \leq d_{x21}(1 - d_{y21}) + d_{y21}(1 - d_{x21}), \quad (4.65)$$

where in the last line all the terms on the right hand side are non-negative, for that reason Eq. (4.65) is always true.

Note that in Eq. (4.61) we still employ max$(x_3, y_2)$ and max$(y_3, x_2)$. The relation of the corresponding $\text{Diff}^*_i$ and $U_{\text{SPX}_i}$ parts to be proven is:

$$2 \cdot A_1U_1A_2U_2d_{y32}d_{x32} \leq A_1U_1A_2U_2(\max(x_3, y_2) + \max(y_3, x_2))$$

$$2 \cdot d_{y32}d_{x32} \leq \max(x_3, y_2) + \max(y_3, x_2) \quad (4.66)$$

This relation is evidently true if one of $d_{y32}$ and $d_{x32}$ is negative, since the left hand side of the relation will be less than zero. If $d_{y32}, d_{x32} \geq 0$, we substitute $d_{y32} + d_{x32} \leq x_3 + y_3 \leq \max(x_3, y_2) + \max(y_3, x_2)$ on the right side and apply Eq. (4.65) as regular, otherwise, if $d_{y32}, d_{x32} \leq 0$, meaning that $y_3 \leq y_2$ and $x_3 \leq x_2$, we can use the $|d_{y32}| + |d_{x32}| \leq y_2 + x_2 \leq \max(x_3, y_2) + \max(y_3, x_2)$ substitution on the right hand side of the relation. Finally we get

$$2 \cdot d_{y32}d_{x32} \leq |d_{y32}| + |d_{x32}|$$

which relation is valid. That way we have proven that Eq. (4.64) is valid implying that the original assumption in Eq. (4.62) was right.

**Convergence of the approximation error**

We already know that $U_{\text{SPX}_i}$ is at most twice as much as $U_{\text{ACC}}$. We can achieve, however, even a much closer lower bound estimation if we define a limit onto the $d$ values:

$$\forall l, j : |d_{xlj}|, |d_{ylj}| \leq \delta_i. \quad (4.67)$$

meaning that the difference between the conditional connection-part availabilities (having the states of the overlapping links as conditions) cannot be greater than $\delta_i$.

If Eq. (4.67) is true, the lower bound is

$$\text{DIV}^{\star}_U \geq 1 - \frac{1}{2}\delta_i \quad (4.68)$$
because after similar transformation as before,

\[
\frac{\text{Diff}_{i}^*}{U_{SPX_i}} \leq \frac{\delta_i}{2} \\
2 \cdot \text{Diff}_{i}^* \leq \delta_i \cdot U_{SPX_i},
\]

instead of Eq. (4.65) we will get now ordered pairs like this:

\[
2 \cdot A_1^2 A_2 U_2 d_{y21} d_{x21} \leq \delta_i A_1^2 A_2 U_2 (d_{x21} + d_{y21}) \\
d_{y21} d_{x21} + d_{y21} d_{x21} \leq \delta_i (d_{x21} + d_{y21}) \\
0 \leq d_{x21} (\delta_i - d_{y21}) + d_{y21} (\delta_i - d_{x21}),
\]

(4.69)

where the terms on the right side are still non-negative.

This way we have proven that Eq. (4.68) is true, and for low \(d\) values we will get \(\text{DIV}_{U_i}^*\) close to 1. However, there are cases when in spite of increasing the link availabilities, the conditional unavailabilities do not converge to 0. In those cases we want to know whether \(\text{DIV}_{U_i}^*\) still converges to 1.

### 4.3.3 Approximation error in case of more than two overlapping links

For single and for dual link overlapping we have already proven that \(\text{DIV}_{U_i}^* \leq 1\). Moreover, for dual overlapping we have also shown that \(\text{DIV}_{U_i}^* \geq 1 - \frac{1}{2} \delta_i\). Here we give proofs for more than two overlapping links. Unfortunately, these proofs cannot be supported by geometric representation.

We assume that there are \(k\) overlapping links between \(\text{conn}_i^H\) and \(\text{conn}_i^T\) \((e_1, e_2, ..., e_k)\). The corresponding link states are denoted by \(S_j \Leftrightarrow S(e_j) = 1\) and \(\overline{S_j} \Leftrightarrow S(e_j) = 0\). The corresponding link availabilities are \(P(S_j) = A_j\) and the unavailabilities are \(P(\overline{S_j}) = 1 - A_j = U_j\).

The S-P approximation states that

\[
A_{SPX_i} = A_{ACC_i^H} \cdot A_{ACC_i^T}
\]

(4.70)

First, we analyse the upper bound of \(\text{DIV}_U\). We assume that there is a single link \((e_1)\) overlap between \(\text{conn}_i^H\) and \(\text{conn}_i^T\). The marginal probability rule Eq. (4.7) states that

\[
P(B) = P(A) \cdot P(B|A) + P(\overline{A}) \cdot P(B|\overline{A})
\]

After substituting \(P(B) = A_{ACC_i^H}\) and \(A = S_1\) (and \(P(A) = A_1\)) into the equation we get

\[
A_{ACC_i^H} = A_1 \cdot A_{ACC_i^H|S_1} + (1 - A_1) \cdot A_{ACC_i^H|\overline{S_1}}
\]

(4.71)
To express the conditional availabilities by means of $A_{\text{ACC}}^H$, we transform Eq. (4.71):

$$A_1 \cdot A_{\text{ACC}}^H|_{S_1} + (1 - A_1) \cdot A_{\text{ACC}}^H|_{\overline{S_1}} = A_1 \cdot A_{\text{ACC}}^H|_{S_1} + (1 - A_1) \cdot A_{\text{ACC}}^H|_{\overline{S_1}},$$

$$A_1 \cdot (A_{\text{ACC}}^H|_{S_1} - A_{\text{ACC}}^H) = (1 - A_1) \cdot (A_{\text{ACC}}^H - A_{\text{ACC}}^H|_{\overline{S_1}})$$

(4.72)

Let $d_1^H$ denote the value of Eq. (4.72):

$$A_1 \cdot (A_{\text{ACC}}^H|_{S_1} - A_{\text{ACC}}^H) = d_1^H,$$

$$(1 - A_1) \cdot (A_{\text{ACC}}^H - A_{\text{ACC}}^H|_{\overline{S_1}}) = d_1^H.$$  

It is important to note that $d_1^H \geq 0$ due to the monotonicity assumption.

The equations can be transformed into

$$A_{\text{ACC}}^H|_{S_1} = A_{\text{ACC}}^H + \frac{d_1^H}{A_1},$$

(4.73)

$$A_{\text{ACC}}^H|_{\overline{S_1}} = A_{\text{ACC}}^H - \frac{d_1^H}{1 - A_1}.$$  

(4.74)

We can write similar equations regarding the tail segment ($\text{conn}_i^T$) of the connection:

$$A_{\text{ACC}}^T = A_1 \cdot A_{\text{ACC}}^T|_{S_1} + (1 - A_1) \cdot A_{\text{ACC}}^T|_{S_1}$$

(4.75)

and we can find a proper $d_1^T \geq 0$ value so that

$$A_{\text{ACC}}^T|_{S_1} = A_{\text{ACC}}^T + \frac{d_1^T}{A_1},$$

(4.76)

$$A_{\text{ACC}}^T|_{\overline{S_1}} = A_{\text{ACC}}^T - \frac{d_1^T}{1 - A_1}.$$  

(4.77)

Using $e_1$ as key element [95] we can express $A_{\text{ACC}}$ as

$$A_{\text{ACC}} = A_1 \cdot A_{\text{ACC}}|_{S_1} + (1 - A_1) \cdot A_{\text{ACC}}|_{\overline{S_1}}.$$  

(4.78)

Since $e_1$ is the only overlapping link, $A_{\text{ACC}}|_{S_1} = A_{\text{SPX}}|_{S_1}$ and $A_{\text{ACC}}|_{\overline{S_1}} = A_{\text{SPX}}|_{\overline{S_1}}$. We can apply these substitutions to Eq. (4.78):

$$A_{\text{ACC}} = A_1 \cdot A_{\text{SPX}}|_{S_1} + (1 - A_1) \cdot A_{\text{SPX}}|_{\overline{S_1}},$$

(4.79)

$$= A_1 \cdot A_{\text{ACC}}^H|_{S_1} \cdot A_{\text{ACC}}^T|_{S_1} + (1 - A_1) \cdot A_{\text{ACC}}^H|_{\overline{S_1}} \cdot A_{\text{ACC}}^T|_{\overline{S_1}},$$

$$= A_1 \cdot (A_{\text{ACC}}^H + \frac{d_1^H}{A_1}) (A_{\text{ACC}}^T + \frac{d_1^T}{A_1})$$

$$(1 - A_1) (A_{\text{ACC}}^H - \frac{d_1^H}{1 - A_1}) (A_{\text{ACC}}^T - \frac{d_1^T}{1 - A_1})$$

(4.80)
If we examine the right hand side of Eq. (4.80), we can simplify it.

\[
A_{ACC} = A_1 \cdot A_{ACC}^H \cdot A_{ACC}^T + d_1^T \cdot A_{ACC}^H + d_1^H \cdot A_{ACC}^T + \frac{d_1^H \cdot d_1^T}{A_1}
\]

\[
+ A_{ACC}^H \cdot A_{ACC}^T
\]

\[
- A_1 \cdot A_{ACC}^H \cdot A_{ACC}^T - d_i^T \cdot A_{ACC}^H - d_i^H \cdot A_{ACC}^T + \frac{d_i^H \cdot d_i^T}{1 - A_1}
\]

\[
= A_{ACC}^H \cdot A_{ACC}^T + \frac{d_i^H \cdot d_i^T}{A_1} + \frac{d_i^H \cdot d_i^T}{1 - A_1}
\]

\[
= A_{SPX} + \frac{d_i^H \cdot d_i^T}{A_1 \cdot (1 - A_1)} \quad (4.81)
\]

As \(d_i^H\) and \(d_i^T\) are nonnegative values Eq. (4.81) expresses that in case of single link overlap \(A_{ACC}\) is not less than \(A_{SPX}\) implying that \(\text{DIV}_{U_i} \leq 1\).

In case of multiple link overlaps, we need to generalize Eq. (4.78) for any \(G\) condition and \(e_j\) link:

\[
A_{ACC}|_G = A_j \cdot A_{ACC}|_{G, S_j} + (1 - A_j) \cdot A_{ACC}|_{G, \overline{S_j}}. \quad (4.82)
\]

Similarly, Eq. (4.73) and Eq. (4.74) also need to be generalized using a proper \(d_{g,j}^H \geq 0\) value:

\[
A_{ACC}^H|_{G, S_j} = A_{ACC}^H|_G + \frac{d_{g,j}^H}{A_j}, \quad (4.83)
\]

\[
A_{ACC}^H|_{G, \overline{S_j}} = A_{ACC}^H|_G - \frac{d_{g,j}^H}{1 - A_j}. \quad (4.84)
\]

In the same manner we define \(A_{ACC}^T|_{G, S_j}\) and \(A_{ACC}^T|_{G, \overline{S_j}}\). Based on the monotonicity assumption, resulting in \(d_{g,j}^H, d_{g,j}^T \geq 0\), and using these generalized expressions we can express relationship between \(A_{SPX}|_G\) and \(A_{SPX}|_{G, S_j}\) (and \(A_{ACC}|_{G, \overline{S_j}}\)). According to Eq. (4.79) and Eq. (4.81) we can state that

\[
A_j \cdot A_{SPX}|_{G, S_j} + (1 - A_j) \cdot A_{SPX}|_{G, \overline{S_j}} = A_{SPX}|_G + \frac{d_{g,j}^H \cdot d_{g,j}^T}{A_j \cdot U_j}, \quad (4.85)
\]

\[
A_j \cdot A_{SPX}|_{G, S_j} + (1 - A_j) \cdot A_{SPX}|_{G, \overline{S_j}} \geq A_{SPX}|_G. \quad (4.86)
\]

To get the relation between \(A_{ACC}\) and \(A_{SPX}\) we have to express \(A_{ACC}\) using Eq. (4.82) recursively for each \(e_j\) overlapping link. For example, if there are two overlapping links (\(e_1\) and \(e_2\)) then first we use \(e_1\) as key element and express \(A_{ACC}\) as Eq. (4.78) shows. Next, we have to unfold the conditional availabilities of Eq. (4.78) using \(e_2\) as key element:

\[
A_{ACC}|_{S_1} = A_2 \cdot A_{ACC}|_{S_1, S_2} + (1 - A_2) \cdot A_{ACC}|_{S_1, \overline{S_2}};
\]

\[
A_{ACC}|_{S_1} = A_2 \cdot A_{ACC}|_{S_1, S_2} + (1 - A_2) \cdot A_{ACC}|_{S_1, \overline{S_2}}.
\]
If the condition $G$ binds the state of each key element, $A_{ACC}|G$ and $A_{SPX_i}|G$ are identical. In the fully unfolded expression of $A_{ACC}$ each $A_{ACC}|G$ conditional availability can be replaced by the corresponding $A_{SPX_i}|G$ value:

$$(\forall 1 \leq j \leq k (S_j \supseteq G \lor \overline{S_j} \supseteq G)) \Rightarrow (A_{ACC}|G = A_{SPX_i}|G) \quad (4.87)$$

After these replacements we can eliminate the $G$ conditions in the expression by applying Eq. (4.86) onto each term pair. Finally we get to the relation

$$A_{ACC} \geq A_{SPX_i} \quad (4.88)$$

which is equivalent to

$$DIV_{U^*_i} \leq 1. \quad (4.89)$$

To find a lower bound of $DIV_{U^*_i}$ we have to recall Fig. 4.7 and generalize the formulae describing $Diff^*_i$ (Eq. (4.58)) and $U^*_{SPX_i}$ (Eq. (4.60)) to handle more than two overlapping links. First we define a basic condition that binds the state of each overlapping link:

$$G = \bigcap_{0 < j \leq k} S_j \quad (4.90)$$

In the generalized expressions we will use $G^H \supseteq G$ which binds the state of only a subset of the the overlapping links, and we will use also the extended version of $G^H$ that binds the state of those links that are not bound by the condition $G^H$:

$$G^H \supseteq G \iff G^H_X = G^H \bigcap_{S_j \supseteq G^H} \overline{S_j} \quad (4.91)$$

This means that $G^H_X$ binds the state of each considered overlapping link: those links which are bound also by $G^H$ are $Up$, the other links are in $Down$ state. The conditions $G^T$ and $G^T_X$ are defined similarly.

In Eq. (4.58) and Eq. (4.60) we see that the conditional connection unavailabilities ($x$ values) and their differences ($d$ values) are accompanied by $A$ and $U$ link availabilities (unavailabilities) depending on the state of the overlapping link. Here we define the availability of the head part of the connection as a sum of different conditional availabilities weighted by the proper link availabilities (and unavailabilities):

$$A_{ACC_i}^H = \sum_{G^H \supseteq G} \left( \prod_{S_j \supseteq G^H_X} A_j \right) \left( \prod_{\overline{S_j} \supseteq G^H} U_j \right) \cdot A_{ACC_i}^H|G^H_X \quad (4.92)$$

Similarly,

$$A_{ACC_i}^T = \sum_{G^T \supseteq G} \left( \prod_{S_j \supseteq G^T_X} A_j \right) \left( \prod_{\overline{S_j} \supseteq G^T} U_j \right) \cdot A_{ACC_i}^T|G^T_X \quad (4.93)$$
By means of these expressions we can define $A_{ACC}$ as

$$A_{ACC} = \sum_{G^H \supseteq G} \left( \prod_{S_j \supseteq G^H_X} A_j \right) \left( \prod_{S_j \supseteq G^H_X} U_j \right) \cdot A_{ACC}^H|_{G^H_X} \cdot A_{ACC}^T|_{G^T_X} \quad (4.94)$$

and $A_{SPX_i}$:

$$A_{SPX_i} = A_{ACC}^H \cdot A_{ACC}^T$$

$$= \sum_{G^H \supseteq G} \left( \prod_{S_j \supseteq G^H_X} A_j \right) \left( \prod_{S_j \supseteq G^H_X} U_j \right) \cdot A_{ACC}^H|_{G^H_X}$$

$$\cdot \sum_{G^T \supseteq G} \left( \prod_{S_j \supseteq G^T_X} A_j \right) \left( \prod_{S_j \supseteq G^T_X} U_j \right) \cdot A_{ACC}^T|_{G^T_X} \quad (4.95)$$

To get the difference of the estimated and exact availabilities ($\text{Diff}^*_i$) we have to subtract Eq. (4.95) from Eq. (4.94):

$$\text{Diff}^*_i = \sum_{G^H \supseteq G} \left( \prod_{S_j \supseteq G^H_X} A_j \right) \left( \prod_{S_j \supseteq G^H_X} U_j \right)$$

$$\cdot \sum_{G^T \supseteq G} \left( \prod_{S_j \supseteq G^T_X} A_j \right) \left( \prod_{S_j \supseteq G^T_X} U_j \right)$$

$$\cdot (A_{ACC}^H|_{G^H_X} \cdot (A_{ACC}^T|_{G^T_X} - A_{ACC}^T|_{G^H_X})) \quad (4.96)$$

Similarly, we could have written

$$\text{Diff}^*_i = \sum_{G^H \supseteq G} \left( \prod_{S_j \supseteq G^H_X} A_j \right) \left( \prod_{S_j \supseteq G^H_X} U_j \right)$$

$$\cdot \sum_{G^T \supseteq G} \left( \prod_{S_j \supseteq G^T_X} A_j \right) \left( \prod_{S_j \supseteq G^T_X} U_j \right)$$

$$\cdot (A_{ACC}^H|_{G^H_X} \cdot (A_{ACC}^T|_{G^T_X} - A_{ACC}^T|_{G^H_X})) \quad (4.97)$$

We can apply the transformation $a \cdot (b - c) = -a \cdot (c - b)$ on Eq. (4.97):

$$A_{ACC}^H|_{G^T_X} \cdot (A_{ACC}^T|_{G^T_X} - A_{ACC}^T|_{G^H_X}) = -A_{ACC}^H|_{G^T_X} \cdot (A_{ACC}^T|_{G^T_X} - A_{ACC}^T|_{G^H_X}) \quad (4.98)$$

and we can use unavailabilities instead of availabilities in the differences:

$$A_{ACC}^T|_{G^H_X} - A_{ACC}^T|_{G^T_X} = U_{ACC}^T|_{G^H_X} - U_{ACC}^T|_{G^T_X} \quad (4.99)$$

We can define $\text{Diff}^*_i$ as the average of Eq. (4.96) and Eq. (4.97). After
applying the transformations of Eq. (4.98) and Eq. (4.99) we get:

\[
\text{Diff}^*_i = \sum_{G^H \supseteq G} \left( \prod_{s_j \supseteq G^H_X} A_j \right) \left( \prod_{s_j \supseteq G^T_X} U_j \right) \\
\cdot \sum_{G^T \supseteq G} \left( \prod_{s_j \supseteq G^T_X} A_j \right) \left( \prod_{s_j \supseteq G^T_X} U_j \right) \\
\cdot \frac{1}{2} \left( (U_{\text{ACC}}^H|_{G^H_X} - U_{\text{ACC}}^H|_{G^T_X}) \cdot (U_{\text{ACC}}^T|_{G^H_X} - U_{\text{ACC}}^T|_{G^T_X}) \right)
\]

We have to compare this \( \text{Diff}^*_i \) value to \( U^\text{low}_{SP_X_i} \). We obtain \( U^\text{low}_{SP_X_i} \) from \( A_{SP_X_i} \). We examine the terms of Eq. (4.95). For general \( a_1 = 1 - u_1 \) and \( a_2 = 1 - u_2 \) probability values we can state that

\[
1 - a_1 \cdot a_2 = 1 - (1 - u_1) \cdot (1 - u_2) \\
= 1 - 1 + u_1 + u_2 - u_1 \cdot u_2 \\
= u_1 + u_2 - u_1 \cdot u_2 \\
= (1 - u_1) \cdot u_2 + u_1 \geq u_1 \\
\text{or} \quad = (1 - u_2) \cdot u_1 + u_2 \geq u_2 \quad (4.101)
\]

Simplified:

\[
1 - a_1 \cdot a_2 \geq \max(u_1, u_2) \quad (4.103)
\]

The terms of \( A_{SP_X_i} \) are

\[
A_{\text{ACC}}^H|_{G^X_X} \cdot A_{\text{ACC}}^T|_{G^X_X}
\]

To obtain \( U_{SP_X_i} \) all the terms need to be inverted. The terms of \( U_{SP_X_i} \) are:

\[
1 - A_{\text{ACC}}^H|_{G^H_X} \cdot A_{\text{ACC}}^T|_{G^T_X}
\]

For each term of \( U_{SP_X_i} \) we can define a lower bound according to Eq. (4.103):

\[
1 - A_{\text{ACC}}^H|_{G^H_X} \cdot A_{\text{ACC}}^T|_{G^T_X} \geq \max(U_{\text{ACC}}^H|_{G^H_X}, U_{\text{ACC}}^T|_{G^T_X}) \quad (4.104)
\]

Our original statement was Eq. (4.68) \( (DIVU^*_i \geq 1 - \frac{1}{2}\delta_i) \) where \( \delta_i \) is the highest difference between the considered conditional unavailabilities. Here we define \( \delta_i \) as:

\[
\forall G^H \supseteq G, G^T \supseteq G \quad \left| U_{\text{ACC}}^H|_{G^H_X} - U_{\text{ACC}}^H|_{G^T_X} \right| \leq \delta^H_i \\
\forall G^H \supseteq G, G^T \supseteq G \quad \left| U_{\text{ACC}}^T|_{G^H_X} - U_{\text{ACC}}^T|_{G^T_X} \right| \leq \delta^T_i \\
\max(\delta^H_i, \delta^T_i) = \delta_i \quad (4.105)
\]
For multiple link overlapping we can prove Eq. (4.68) by proving \( \text{Diff}^*_i \leq \frac{1}{2} \delta_i U_{SPX_i} \) term by term. According to Eq. (4.100) and Eq. (4.104) the corresponding relations are

\[
\frac{1}{2} ((U_{ACC}^H|_{G^H_i} - U_{ACC}^H|_{G^X_i}) \cdot (U_{ACC}^T|_{G^H_i} - U_{ACC}^T|_{G^X_i})) \leq \frac{1}{2} \delta_i \max(U_{ACC}^H|_{G^H_i}, U_{ACC}^T|_{G^T_i}) \tag{4.106}
\]

If either of \((U_{ACC}^H|_{G^H_i} - U_{ACC}^H|_{G^X_i})\) and \((U_{ACC}^T|_{G^H_i} - U_{ACC}^T|_{G^X_i})\) is negative Eq. (4.106) is true, since the right hand side of the inequality will be nonpositive.

If both the coefficients at the right hand side are positive, we substitute \((U_{ACC}^H|_{G^H_i} - U_{ACC}^H|_{G^X_i})\) and \((U_{ACC}^T|_{G^H_i} - U_{ACC}^T|_{G^X_i})\) and Eq. (4.106) is true.

Finally, if both the coefficients at the right hand side are negative, we can substitute \((U_{ACC}^T|_{G^X_i} - U_{ACC}^T|_{G^H_i})\) and \((U_{ACC}^H|_{G^X_i} - U_{ACC}^H|_{G^H_i})\) \(\leq \delta_i\). This way we have proven that Eq. (4.106) is true for each term. Thus

\[
DIV^*_i = \frac{U_{ACC}}{U_{SPX_i}} \geq 1 - \frac{1}{2} \delta_i \tag{4.107}
\]

in case there are multiple overlapping links between \(\text{conn}_i^H\) and \(\text{conn}_i^T\).

In this section we have defined upper and lower bounds of \(DIV^*_i\) in case of multiple overlapping links. However, we also want to derive lower and upper bounds of \(DIV_U\).

The upper bound can be derived easily. If neither \(\text{conn}_i^H\) nor \(\text{conn}_i^T\) contains overlapping links, then \(A_{SPX_i} = A_{SP}\). If any of the connection parts contains overlapping link, then we can examine that given connection part as a separate connection. For this connection part either we can state that \(A_{ACC} \geq A_{SPX_j} = A_{SP}\) or we have to examine its parts (recursively). Finally, this recursion results in

\[
A_{ACC} \geq A_{SP}, \tag{4.108}
\]

which is equivalent to

\[
DIV_U \leq 1. \tag{4.109}
\]

Concerning the lower bound, we assume that there are \(m \leq n - 1\) nodes in the connection where overlap happens: \(\forall i \in \{i_1, i_2, \ldots, i_m\} : \text{conn}_i^H\) and \(\text{conn}_i^T\) have common links. As a generalization of \(U_{SPX_i}\) we introduce \(U_{SPX_{i,j,\ldots}}\) to denote that the connection is split into multiple parts at nodes \(i, j, \ldots\), the accurate availability of the connection parts are known and the unavailability of \(\text{conn}\) is calculated by applying the S-P method at nodes
We can express:

$$DIV_U = \frac{U_{ACC}}{U_{SP}}$$

$$= \frac{U_{ACC}}{U_{SPX_i}} \cdot \frac{U_{SPX_i}}{U_{SP}}$$

$$= \frac{U_{ACC}}{U_{SPX_i}} \cdot \frac{U_{SPX_i}}{U_{SPX_{i_1}} \cdot \ldots \cdot U_{SPX_{i_1,i_2}} \cdot \ldots \cdot U_{SPX_{i_1,i_2,\ldots,i_{m-1}}}}{U_{SP}}$$

we have to find the proper $i_x$ where $DIV_U^* = \frac{U_{ACC}}{U_{SPX_{i_x}}}$ is the minimal. In this case we can state that

$$DIV_U \geq \left( \frac{U_{ACC}}{U_{SPX_{i_1}}} \right)^{m-1} \geq \left( 1 - \frac{1}{2} \delta_x \right)^{m-1}$$

4.3.4 Scenarios illustrating approximation error

To analyze the convergence of $DIV_U^*$ and $DIV_U$ we have examined some scenarios. Figure 4.8 depicts connection segments of seven basic scenarios. In each example the connection is 10 hops long and within a connection the links – or in the last three scenarios the link pairs – of the default path are protected in the same manner. The figures show the neighboring default links (solid lines) with their protection (dashed lines). Parallel lines between two nodes mean overlapping links. In the calculations, the value of $\varepsilon$ is the order of magnitude of the link unavailabilities:

$$\forall e \in E : P(S(e) = 0) = \Theta(\varepsilon).$$

This means by definition that using proper $c_1$ and $c_2$ constant values we can state that

$$\forall e \in E : c_1 \cdot \varepsilon \leq P(S(e) = 0) \leq c_2 \cdot \varepsilon.$$
Note that unprotected link chains inherit this order of magnitude – e.g.,

\[
P(S(\text{Prot}_i) = 0) = \overline{F}_i = 1 - B_i = 1 - \prod_{j=0}^{J_i} b_{i,j} \approx \sum_{j=0}^{J_i} b_{i,j}
\]

\[
= \sum_{j=0}^{J_i} P(S(p_{i,j}) = 0) = \Theta(\varepsilon).
\]

Simulations were carried out on the examples shown in Fig. 4.8 to define the value of \(DIV_U\). In the simulations we have assumed that each link in the network has the same availability metric \(P(S(e) = 0) = \varepsilon, c_1 = c_2 = 1\). The results are shown in Fig. 4.9.

Scenario (a) does not have any overlapping links at all, scenario (b) contains only backup link overlapping. In these basic examples the conditional unavailabilities \((U_{ACC}|G)\) values converge to 0, this way \(\delta_i\) also converges to 0 implying that \(DIV_U \rightarrow 1\).

In scenarios (c) and (d), examining the connection at node \(v_i\), the protection of \(w_i\) leads over \(w_{i+1}\). Using the notation of Sect. 4.3.2 we see that \(p_{i,1} = w_{i+1} = e_{h^*}\); moreover, in (d) \(p_{i,3} = p_{i+1,4} = e_{h^*}\). In the latter case if both commonly used links are down, the connection becomes unavailable. This is expressed by the coefficient \(y_4 = 1\). \(y_4 = 1\) implies that \(d_{y_{41}}, d_{y_{42}}, d_{y_{43}} \rightarrow 0\), which means that there cannot be found any \(\delta_i < 1\) for \(DIV_U^*_i\), hereby the lower bound of the accuracy of the \(S-P\) is \(\frac{1}{2} \leq DIV_U^*_i\).

Still, \(DIV_U^*_i\) and \(DIV_U\) converges to 1. The reason for this behavior is the following: all the remaining conditional probabilities of \(x_l\) and \(y_j\) for \(l \in \{1, 2, 3, 4\}\) and \(j \in \{1, 2, 3\}\) will converge to 0, since they refer to states of \(conn^H\) or \(conn^T\), when either the working path or the protection path (or both) are available. All these cases imply a conditional unavailability of order \(\varepsilon\): we can find a proper \(c_3\) value so that \(x_l, y_j \leq c_3 \cdot \varepsilon\) for \(l \in \{1, 2, 3, 4\}, j \in \{1, 2, 3\}\). This upper bound \((c_3 \cdot \varepsilon)\) is inherited also by the remaining \(d\) values – all except for the previously mentioned \(d_{y_{41}}, d_{y_{42}}, d_{y_{43}}\).

Without loss of generality, but for the sake of simple calculations, we assume that \(c_2 \leq c_3\). This way, substituting \(c_3 \cdot \varepsilon\) values into Eq. (4.58) we get that \(Diff_i^* \leq 6 \cdot c_3^3 \cdot \varepsilon^3\). Whereas – after applying the lower bound of \(U_{SPX_i} \geq U_{12y_i}\) taken from Fig. 4.7 – \(U_{SPX_i} \geq c_1^2 \cdot \varepsilon^2\). Thus the quotient of \(Diff_i^*\) and \(U_{SPX_i}\) is

\[
\frac{Diff_i^*}{U_{SPX_i}} \leq \frac{6 \cdot c_3^3 \cdot \varepsilon^3}{c_1^2 \cdot \varepsilon^2} = 6 \cdot c_3^3 \cdot c_1^{-2} \cdot \varepsilon,
\]

which also converges to 0. The simulation results in Fig. 4.9 confirm this reasoning.
Scenarios (e), (f) and (g) are more complicated than the former ones. Here we use the annotations of Sect. 4.3.3. In these scenarios the backup paths of \( \omega_i \) and \( \omega_{i+1} \) mutually overlap the default link of each other (\( \omega_i = e_1 \) and \( \omega_{i+1} = e_2 \)). The question is where do the \( \text{DIV}_U \) values of these connections converge.

On the one hand, we can observe that if both overlapped links are down, the connection, formally, the half-connections \( \text{conn}_i^H \) and \( \text{conn}_i^T \) become unavailable: \( U_{\text{ACC}_i}^H|_{S_1,S_2^1} = U_{\text{ACC}_i}^T|_{S_1,S_2^2} = 1. \)

On the other hand, however, if the default path is available we do not care about the state of the backup path (in case of \( S_1 \) we use \( U_{\text{ACC}_i}^H|_{S_1} = U_{\text{ACC}_i}^H|_{S_1} \) and in case of \( S_2 \) we use \( U_{\text{ACC}_i}^T|_{S_2} = U_{\text{ACC}_i}^T|_{S_2} \)). Both \( U_{\text{ACC}_i}^H|_{S_1} \) and \( U_{\text{ACC}_i}^T|_{S_2} \) are \( \Theta(\varepsilon^2) \), since the connection unavailability depends in these cases only on the remaining connection segments \( \text{conn}_i^{H-L} \) and \( \text{conn}_i^{T-L} \), and these segments are protected (against single failure), i.e., they get unavailable only in case of two or more failures, resulting in unavailability of \( \Theta(\varepsilon^2) \). These conditional unavailabilities were easy to evaluate.

The remaining probabilities, i.e., when only the backup paths are available, are more complicated to evaluate. For \( U_{\text{ACC}_i}^H|_{S_1,S_2} \) and \( U_{\text{ACC}_i}^T|_{S_1,S_2} \) the connection segments contain an unprotected link chain – practically, the backup path is operational and it is not protected – thus their unavailabilities are \( \Theta(\varepsilon) \).

Knowing that \( U_{SPX_i} \) is \( \Theta(\varepsilon^2) \), instead of \( U_{SPX_i} \) we want to analyze the convergence of a more expressive metric. If we want to find a metric that converges to a constant non-zero value, its order of magnitude has to be \( U_{SPX_i} \). For example, \( \frac{U_{SPX_i}}{\varepsilon^2} \) fulfills this criterion.

\[
\lim_{\varepsilon \to 0} \frac{U_{SPX_i}}{U_1 U_2} = \lim_{\varepsilon \to 0} \frac{U_{\text{ACC}_i}^T + U_{\text{ACC}_i}^H - U_{\text{ACC}_i}^T \cdot U_{\text{ACC}_i}^H}{U_1 U_2} \\
= \lim_{\varepsilon \to 0} \frac{A_1 U_2 U_{\text{ACC}_i}^T|_{S_1,S_2^1} + U_1 U_2 + U_1 A_2 U_{\text{ACC}_i}^H|_{S_1,S_2^2} + U_1 U_2}{U_1 U_2} - \\
\frac{(A_1 U_2 U_{\text{ACC}_i}^T|_{S_1,S_2^1} + U_1 U_2) \cdot (U_1 A_2 U_{\text{ACC}_i}^H|_{S_1,S_2^2} + U_1 U_2)}{U_1 U_2} \\
= 2 + \lim_{\varepsilon \to 0} \frac{U_2 U_{\text{ACC}_i}^T|_{S_1,S_2^1} + U_1 U_{\text{ACC}_i}^H|_{S_1,S_2^2}}{U_1 U_2} \\
= 2 + \lim_{\varepsilon \to 0} \frac{U_{\text{ACC}_i}^T|_{S_1,S_2^1}}{U_1} + \lim_{\varepsilon \to 0} \frac{U_{\text{ACC}_i}^H|_{S_1,S_2^2}}{U_2}. \tag{4.112}
\]

Regarding the \( \text{Diff}_i^4 \) metric our starting point is Eq. (4.58). We already showed that \( d_{x4t} \) and \( d_{y4t} \) values converge to 1, whereas the other \( d \) values are \( \Theta(\varepsilon^n) \) where \( n \) is 1 or higher. This way there will remain only one term of the sum \( \text{Diff}_i^4 \) which is \( \Theta(\varepsilon^2) \) making the limit value calculation easy for
the following expression:

\[
\lim_{\varepsilon \to 0} \frac{\text{Diff}^*_i}{U_1U_2} = \lim_{\varepsilon \to 0} \frac{U_1U_2d_{41}d_{41}}{U_1U_2} + \varepsilon \cdot \text{rest} = \lim_{\varepsilon \to 0} d_{41}d_{41} = 1.
\]

Putting Eq. (4.112) and Eq. (4.113) together, finally we get that

\[
\lim_{\varepsilon \to 0} \text{DIV} U^*_i = 1 - \frac{1}{2 + \lim_{\varepsilon \to 0} \frac{U_{\text{ACC}^T}|_{S_1S_2}}{U_1} + \lim_{\varepsilon \to 0} \frac{U_{\text{ACC}^H}|_{S_1S_2}}{U_2}}.
\]

The lines corresponding to the last three scenarios in Fig. 4.9 show that in these cases

\[
\lim_{\varepsilon \to 0} \text{DIV} U^*_i = \lim_{\varepsilon \to 0} \text{DIV} U.
\]

The explanation of this equation is that initially, if we take connections that are \( n = 2 \) long, where \( U_{SPX_1} = U_{SP} \) implying \( \text{DIV} U^*_1 = \text{DIV} U \). As the connection gets longer, both \( U_{\text{ACC}} \) and \( U_{SP} \) increases by nearly the same factor. Thus their quotient \( (\text{DIV} U = U_{\text{ACC}}/U_{SP}) \) remains nearly invariant.

Recalling that \( U_1 \approx U_2 \approx \varepsilon \), in scenario (e) the cycles are 3 links long, this way \( U_{\text{ACC}^T}|_{S_1S_2} \approx U_1 \) and \( U_{\text{ACC}^H}|_{S_1S_2} \approx U_2 \). \( \text{DIV} U \) will converge to \( 1 - 1/(2 + 1 + 1) = 1 - 1/4 = 3/4 \). In scenario (f) the cycles are 4 links long, this way \( U_{\text{ACC}^T}|_{S_1S_2} \approx 2U_1 \) and \( U_{\text{ACC}^H}|_{S_1S_2} \approx 2U_2 \). \( \text{DIV} U \) will converge to \( 1 - 1/(2 + 2 + 2) = 1 - 1/6 = 5/6 \). Finally, in scenario (g) the cycles are 6 links long, this way \( U_{\text{ACC}^T}|_{S_1S_2} \approx 4U_1 \) and \( U_{\text{ACC}^H}|_{S_1S_2} \approx 4U_2 \). \( \text{DIV} U \) will converge to \( 1 - 1/(2 + 4 + 4) = 1 - 1/10 = 0.9 \).

4.3.5 Summary of results

In this section, first, we have studied single link overlaps (Sect. 4.3.1). Next, we have extended the examinations for dual link overlaps (Sect. 4.3.2). Finally, in Sect. 4.3.3, we have generalized the results for multiple link overlappings. Here we give a summary of this section using the general notations introduced in Sect. 4.3.3. We have proven the following:

- The accurate unavailability of a connection \( U_{\text{ACC}} \) is always lower than the unavailability approximated by the serial-parallel method \( (U_{SP}). \)

- There can be defined lower bounds of the divergence expressed by \( \text{DIV} U = U_{\text{ACC}}/U_{SP} \). We have examined this problem for connections where link overlapping is only between \( \text{conn}_i^H \) and \( \text{conn}_i^H \). Denoting the maximal difference between conditional unavailabilities by \( d_i^H \) (for
the head segment of the connection) and by $\delta_i^T$ (for the tail segment of the connection) we have proven that

$$DIV_U \geq \begin{cases} 
1 - \min(\delta_i^H, \delta_i^T), & \text{in case of single link overlap} \\
1 - \frac{1}{2} \max(\delta_i^H, \delta_i^T), & \text{in case of multiple link overlaps.}
\end{cases}$$

• By increasing the link availability metric in the network, i.e., $\forall e : P(S(e) = 0) \leq \varepsilon$, the divergence of the calculated and the accurate connection unavailability converges to a defined value. If the protections of neighboring working links ($e_1$ and $e_2$) mutually overlap each other, then $U_1$ and $U_2$ denote the unavailability of these links, furthermore $USP_i^H|S_1, S_2$ and $USP_i^T|S_1, S_2$ denote the connection segment unavailabilities with bound state of the given links. If link overlapping is only between $conn_i^H$ and $conn_i^H$, we have shown that

- If there are no mutual working link overlaps,

$$\lim_{\varepsilon \to 0} DIV_U = 1.$$

- If there are mutual working link overlaps,

$$\lim_{\varepsilon \to 0} DIV_U = 1 - \frac{1}{2 + \lim_{\varepsilon \to 0} \frac{U_{ACC_i^T}|S_1, S_2}{U_1} + \lim_{\varepsilon \to 0} \frac{U_{ACC_i^H}|S_1, S_2}{U_2}}.$$
The presented theoretical result can be applied in several cases. For example, the different protection alternatives of the same connection can be partially ranked by their availability, as long as the difference between the results of the \textit{S-P} approximation are higher than the maximal approximation error of the \textit{S-P} method. We achieved further applicability in developing most available path searching algorithm for link-protected (e.g., \textit{p}-cycle-protected) connections. Using the \textit{S-P} approximation and knowing the upper bound of its divergence, for each protected link we can define a value higher than the real availability (overestimation). In our application we use these overestimation values as hint within an \textit{A*}-algorithm.
Chapter 5

Conclusion

In this dissertation I summarized the main achievements of the research I have carried out during my doctoral studies. The problems that we have examined are related to the connection provisioning. When provisioning (virtual) circuit switched connections, from the point of view of the routing decision making entity, it is essential to have sufficient information available about the network topology and the link states.

In optical networks the amount of link state changes does not scale well with the number of wavelength channels. To solve this problem, in Sect. 2.2, we have presented a topology advertisement scheme that takes into consideration the correlation of link state changes and offers a much more scalable way of link state advertisement. Moreover, this solution has been extended to support protection (Sect. 2.4), and at this point we had to find a way of describing the failure-dependent switching and resource requirement of possibly any conventional protection scheme in a generalized way. Such a general description can be given with the Generalized Protection Formula (Sect. 2.3).

My theses regarding this topic are as follows:

**Thesis 1.1** I have proposed the RBTA (*Rule Based Topology Advertisement*) method for lossless and compact topology information advertisement. In case of dynamic connection provisioning, the amount of information advertised by RBTA is independent of the size and complexity (wavelength and port number) of the network devices, thus, in heterogeneous optical networks, the RBTA outperforms conventional LSA methods.

**Thesis 1.2** I have introduced the *GPF (Generalized Protection Formula)* function, that provides a unified way to describe the switching and allocation requirement of paths realizing the network connections. This generalized description provides a simple and integrated way for network operators to deploy any of the conventionally used protection schemes in the network and use them simultaneously.
Thesis 1.3 Based on Theses 1.1 and 1.2 I have proposed the method RBTA-P (RBTA supporting Protection) for state advertisement and information processing, which provides a compact and lossless description of the switching state of the network in any failure-dependent – including failure-free – cases. The signaling and data overhead of the method is proportional to the number of state changes and the average number of number of considered failure scenarios along the working paths.

In an inter-domain environment the topology advertisement has to face certain problems, as the operators do not want to share the topology of their domains for security and scalability reasons. We came to the conclusion that the best choice to protect connections in inter-domain networks is to use pre-configured protection paths. Hence, as a feasible solution in inter-domain networks, we have proposed multidomain p-cycles in Chapt. 3, and have examined the performance of this protection scheme by means of provided availability vs. resource consumption compared to other protection schemes.

Within this topic I have 3 theses:

Thesis 2.1 I have proposed the Multidomain p-Cycle (MDPC) solution to protect traffic routed on inter-domain network links. MDPC, unlike many other protection schemes, is feasible in inter-domain environment.

Thesis 2.2 I have proposed the Least Cost (LC) and Ring Based (RB) methods for intra-domain part resolution of inter-domain p-cycles. These methods require less resources (LC) and provide higher availability (RB) than the original Most Reliable (MR) solution.

As another product of the research with multidomain p-cycles, we have found a straight-forward extension of the Grow candidate p-cycle searching algorithm, the Exhaustive Grow algorithm (Sect. 4.1).

Another problem of connection provisioning is the availability estimation of the provisioned connection. Being aware of the link availability metrics, for p-cycle protection we have proposed an incremental calculation method to get the accurate availability of the connection (Sect. 4.2). Besides, we have examined the approximation error of the Serial-Parallel availability estimation heuristic, and have found constraints of its approximation error (Sect. 4.3).

We can summarize the results of this topic as:

Thesis 3.1 I have proposed the Exhaustive Grow algorithm, as an extension of the Grow algorithm, to detect a larger (at least equal) set of candidate cycles that is still smaller than the theoretically maximal set. Compared to Grow, using the cycle set of Exhaustive Grow we can protect the connections by less resources, and compared to the
theoretically maximal set, Exhaustive Grow produces less cycles, thus the protection assignment requires less computational steps.

**Thesis 3.2** I have proposed a recursive probability calculation method for getting the exact availability metric of a connection protected by $p$-cycles. The calculation complexity of the method is of $O(n \cdot 2^{L_{\text{max}}} + 1)$.

**Thesis 3.3** For $p$-cycle protected connections I have defined the approximation error of the $S\cdot P$ method, the bounds of this error and with certain restrictions its limit value as the link unavailabilities tend to 0 in case of overlapping working or backup links.
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASON</td>
<td>Automatically Switched Optical Network</td>
</tr>
<tr>
<td>ASTN</td>
<td>Automatic Switched Transport Network</td>
</tr>
<tr>
<td>ATM</td>
<td>Asynchronous Transfer Method</td>
</tr>
<tr>
<td>BLSR</td>
<td>Bi-directional Line Switched Ring</td>
</tr>
<tr>
<td>CBN</td>
<td>on-Cycle link Border Node</td>
</tr>
<tr>
<td>CC</td>
<td>Cable-Cuts</td>
</tr>
<tr>
<td>CCAMP</td>
<td>Common Control and Measurement Plane</td>
</tr>
<tr>
<td>CIDA</td>
<td>Capacitated Iterative Design Algorithm</td>
</tr>
<tr>
<td>CP</td>
<td>Control Plane</td>
</tr>
<tr>
<td>DP</td>
<td>Data Plane</td>
</tr>
<tr>
<td>DWDM</td>
<td>Dense Wavelength Division Multiplexing</td>
</tr>
<tr>
<td>E-NNI</td>
<td>Exterior Network-Network Interface</td>
</tr>
<tr>
<td>EGP</td>
<td>Exterior Gateway Protocol</td>
</tr>
<tr>
<td>GMPLS</td>
<td>Generalized Multiprotocol Label Switching</td>
</tr>
<tr>
<td>GPF</td>
<td>Generalized Protection Formula</td>
</tr>
<tr>
<td>ILP</td>
<td>Integer Linear Programing</td>
</tr>
<tr>
<td>IGP</td>
<td>Interior Gateway Protocol</td>
</tr>
<tr>
<td>LC</td>
<td>Least Cost (MDPC resolution)</td>
</tr>
<tr>
<td>LFC</td>
<td>Link Failure Coefficient</td>
</tr>
<tr>
<td>LSA</td>
<td>Link State Advertisement</td>
</tr>
<tr>
<td>LSP</td>
<td>Label Switched Path</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>MDPC</td>
<td>Multidomain $p$-cycle</td>
</tr>
<tr>
<td>MP</td>
<td>Management Plane</td>
</tr>
<tr>
<td>MPLS</td>
<td>Multiprotocol Label Switching</td>
</tr>
<tr>
<td>MR</td>
<td>Most Reliable (MDPC resolution)</td>
</tr>
<tr>
<td>MTBF</td>
<td>Mean Time Between Failures</td>
</tr>
<tr>
<td>MTTR</td>
<td>Mean Time To Repair</td>
</tr>
<tr>
<td>OADM</td>
<td>Optical Add/Drop Multiplexer</td>
</tr>
<tr>
<td>OCC</td>
<td>Optical Connection Controller</td>
</tr>
<tr>
<td>OPEX</td>
<td>Operational Expenses</td>
</tr>
<tr>
<td>OXC</td>
<td>Optical Cross-Connects</td>
</tr>
<tr>
<td>PCC</td>
<td>Path Computation Client</td>
</tr>
<tr>
<td>PCE</td>
<td>Automatic Switched Transport Network</td>
</tr>
<tr>
<td>PNNI</td>
<td>Private Network-Node Interface</td>
</tr>
<tr>
<td>PPP</td>
<td>Partial Path Protection</td>
</tr>
<tr>
<td>RB</td>
<td>Ring Based (MDPC resolution)</td>
</tr>
<tr>
<td>RBTA</td>
<td>Rule-Based Topology Advertisement and Maintenance</td>
</tr>
<tr>
<td>RBTA-P</td>
<td>Rule-Based Topology Advertisement and Maintenance with Protection</td>
</tr>
<tr>
<td>ROADM</td>
<td>Reconfigurable Optical Add/Drop Multiplexer</td>
</tr>
<tr>
<td>RWA</td>
<td>Routing and Wavelength Assignment</td>
</tr>
<tr>
<td>SBN</td>
<td>Straddling link Border Node</td>
</tr>
<tr>
<td>SBPP</td>
<td>Shared Backup Path Protection</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SLA</td>
<td>Straddling Link Algorithm</td>
</tr>
<tr>
<td>SRLG</td>
<td>Shared Risk Link Group</td>
</tr>
<tr>
<td>TDM</td>
<td>Time Division Multiplexing</td>
</tr>
<tr>
<td>TE LSP</td>
<td>Traffic Engineering Label Switched Path</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>TED</td>
<td>Traffic Engineering Database</td>
</tr>
<tr>
<td>TSI</td>
<td>time slot interchange</td>
</tr>
<tr>
<td>VC</td>
<td>Virtual Circuit</td>
</tr>
<tr>
<td>WDM</td>
<td>Wavelength Division Multiplexing</td>
</tr>
<tr>
<td>WG</td>
<td>Wavelength Graph</td>
</tr>
<tr>
<td>WI-OXC</td>
<td>Wavelength-interchangeable OXC</td>
</tr>
<tr>
<td>WS-OXC</td>
<td>Wavelength-selective OXC</td>
</tr>
<tr>
<td>WSON</td>
<td>Wavelength Switched Optical Networks</td>
</tr>
</tbody>
</table>
Bibliography


State Information”, in Proc. of the 3rd International Workshop on QoS in Multiservice IP Networks (QoS-IP 2005), Catania, Italy, Feb. 2004


117


