University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Vedran Miletić

METHOD FOR OPTIMIZING AVAILABILITY OF OPTICAL TELECOMMUNICATION NETWORK IN PRESENCE OF CORRELATED FAILURES

DOCTORAL THESIS

Zagreb, 2015

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Vedran Miletić

METHOD FOR OPTIMIZING AVAILABILITY OF OPTICAL TELECOMMUNICATION NETWORK IN PRESENCE OF CORRELATED FAILURES

DOCTORAL THESIS

Supervisor: Professor Branko Mikac, PhD

Zagreb, 2015

Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Vedran Miletić

METODA OPTIMIRANJA RASPOLOŽIVOSTI OPTIČKE TELEKOMUNIKACIJSKE MREŽE U PRISUSTVU KORELIRANIH KVAROVA

DOKTORSKI RAD

Mentor: Prof. dr. sc. Branko Mikac

Zagreb, 2015.

The doctoral thesis was completed at the University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Telecommunications.

Supervisor: Professor Branko Mikac, PhD

The thesis has 130 pages.

Thesis number: ___________

About the Supervisor

Branko Mikac was born in Gospić in 1947. He received B.Sc., M.Sc. and Ph.D. degrees in electrical engineering from the University of Zagreb, Faculty of Electrical Engineering and Computing (FER), Zagreb, Croatia, in 1970, 1980 and 1986, respectively.

In the period (1970–1974) he was with the Institute for electronics, telecommunications and automation, RIZ, Zagreb. From 1974 he has been working at the Faculty of Electrical Engineering in the Department for Telecommunications. For educational and scientific research purposes he stayed in Ericsson Stockholm, University of Delft and France Telecom R&D, Lannion. In 2003 he was promoted to permanent Full Professor. He participated in 6 scientific projects of Ministry of Science, Education and Sports and SIZ-1 for Science. Since 1991 he was involved in 3 projects of Eurepean Science Foundation COST (239, 266 i 291) and EU projects: FP6 NoE e-Photon/ONe and FP7 NoE BONE. He published as author, co-author, and editor about 100 scientific, profesional papers and chapters, in journals, conference proceedings and books, in the area of optical trasmision systems and reliability of telecommunication networks.

Prof. Mikac is a member of IEEE. He has been participating in international program committees of 5 conferences and he serves as a reviewer for a number of international journals and conferences. In 2007 he received Golden medal “Josip Lončar” from FER for the work in European projects.

O mentoru

Branko Mikac rođen je u Gospiću 1947. Diplomirao je, magistrirao i doktorirao u polju elektrotehnike na Sveučilištu u Zagrebu Fakultetu elektrotehnike i računarstva (FER) 1970., 1980. odnosno 1986.

U periodu (1970–1974) radio je u Institutu za elektroniku, telekomunikacije i automatiku RIZ-a, Zagreb, odakle je prešao na Elektrotehnički fakultet u Zagrebu u Zavod za telekomunikacije, gdje je i danas zaposlen. Radi edukacije i znanstvenog usavršavanja boravio je u Ericssonu u Stockholmu, u Svečilištu u Delftu i u Institutu France Telecom R&D, Lannion. Godine 2003. izabran je za redovitog profesora u trajnom zvanju. Sudjelovao je na 4 znanstvena projekta Ministarstva znanosti, obrazovanja i sporta i SIZ-a I za znanost. Od 1991. sudjeluje u znanstvenim projektima Europske fundacije za znanost: COST (239, 266 i 291) i projektima Europske unije FP6 NoE e-Photon/One(+) i FP7 NoE BONE. Samostalno i u suautorstvu objavio je i uredio stotinjak znanstvenih, stručnih radova, poglavlja i knjiga u području optičkih prijenosnih sustava i pouzdanosti telekomunikacijskih mreža.

Prof. Mikac član je stručne udruge IEEE. Član je programskog odbora 5 međunarodnih konferencija i recenzira radove za veći broj međunarodnih časopisa i konferencija. Godine 2007. primio je Zlatnu plaketu Josip Lončar FER-a za rad na europskim projektima.

Thanks

First and foremost, I would like to thank my supervisor Professor Branko Mikac for the guidance and support over the years this thesis was in the making. I am particularly grateful that he enabled me to solve scientific and engineering problems by using and further developing free and open source software, which is a life passion of mine. People from Department of Telecommunications at FER helped me retain inspiration for doing research work, despite all the challenges. In particular, Matija Džanko, Mirko Sužnjević, Valter Vasić, Krešimir Pripužić, and Luka Humski: thank you for all the research advice and motivation, highly technical and everyday discussions, and (of course) fun that we had together. You guys rock.

I want to thank all my colleagues at University of Rijeka Department of Informatics for their help and all the random tips. Particularly, I would like to thank Martina Ašenbrener Katić and Ema Kušen for all the discussions, Martina Holenko Dlab and Professor Nataša Hoić-Božić for being the senior colleagues and guiding me through the formal aspects of the PhD process, Luka Vretenar for being the coding wizard, and Vanja Slavuj for jumping in and taking the teaching load off me when I needed it the most. Many great people work at University of Rijeka outside my department. I would like to thank Robert Peter, Iva Šarić, Ivna Kavre, Sanda Bujačić, Milena Sošić, Željko Svedružić, Mladen Tomić, and Kristijan Lenac for fruitful discussions on science and technology that transcend the barriers of different fields and particular research topics. I had the pleasure of meeting many brilliant PhD students and their advisors from all over the world. I wish to thank Dimitris Maniadakis and his advisor Dimitris Varoutas for their contribution in the availability analysis of networks produced by topology models.

Network simulator ns-3 has built an awesome community over the years. I want to thank everyone from the community who helped me in some way, but the list would be far too long, so here is a short one: Mathieu Lacage for describing trampoline objects (among others) and encouraging me to code what I need, Peter D. Barnes for all the design alternatives that were thrown away and never implemented (but taught me a lot about the software design), Tommaso Pecorella for that *energetic* midnight discussion in front of the hotel, Alina Quereilhac and Alex Afanasyev for fixing Waf with me, and last, but certainly not least, Tom Henderson for being the leader of an open organization and herding cats more often than not.

I have met many capable students over the years of teaching at University of Rijeka. I would like to mention Tomislav Šubić who helped me with carrying out the CUDA programming work in the thesis, Ivan Ivakić who helped me in teaching ns-3 at University of Rijeka, and Domagoj Margan who helped me see the beautiful world of science in many different ways.

Finally, thanks to my mother Đurđica, my father Damir, and my brother Mladen for all the love, support, and, above everything else, understanding. I would not have been able to finish this thesis without the three of you and all that you did for me. Thank you.

Abstract

The work describes the design goals and methodology in creating a new model of optical telecommunication network used for studying network resilience. The model is implemented by discrete-event network simulator ns-3. The advantages of using the existing simulator core infrastructure provided by ns-3 are analyzed and compared to building own simulator from scratch, or selecting a tool among other existing simulators such as ns-2, OMNeT++, and commercial simulators. The requirements for feature functionality are outlined and high-level overview of the model architecture and its components are provided. The model is extended to support availability evaluation.

Network availability is of paramount importance in optical telecommunication networks. Their rising connectivity and consequently their availability is compromised by link and node failures, usually due to physical force (e.g. digging, earthquake or fire). In optical networks a group of logically distinct links can unintentionally share a physical resource (e.g. a cable or a duct). Such a group, called shared risk link group (SRLG), introduces a situation where a single failure of common resource can cause multiple failures. Failure of common resource usually occurs due to physical force and causes failures of multiple links. Specifically, such a failure can cause both working and spare wavelength path of a logical connection between two edge nodes to fail at the same time, leaving them disconnected until a repair is done. The usual approach to solving this problem consists of introducing more spare capacity to the network and also using a routing algorithm that takes SRLGs into account when computing paths. Such a routing algorithm avoids creating working and spare path pairs that have links contained in the same SRLG, to minimize the negative impact of SRLG failure on logical connection availability.

The number and length of SRLGs, as well as the characteristics of the underlying physical topology can significantly affect network availability. Especially, the physical topology can be represented by realistic synthetic graphs which are created by numerous geographic graph generators. The implementation and usage of six different physical topology models (Random Geometric, Gabriel, Relative Neighborhood, K-Nearest Neighbor, Waxman and Spatial Barabási-Albert) for investigation of the influence of the underlying topology on the optical telecommunication network availability is described. Network availability is estimated using Monte Carlo simulations based on a model of optical telecommunication network implemented by network simulator ns-3. Scenarios utilizing six topology models both in absence and presence of SRLGs are studied, and the optical network availability sensitivity to the underlying physical network topology is presented as the main result.

Routing algorithms were proposed to ensure working and spare paths of a connection in a network are SRLG-disjoint to avoid such common cause failures. However, complete SRLG-disjointedness of working and spare path is not always possible due to limited number of links or limited capacity available in the network, so maximum SRLG-disjoint paths algorithm is taken instead. Maximum SRLG-disjoint path problem is in general NP-hard. In terms of solution quality greedy algorithms for maximum SRLG-disjoint path problem are as good as more complicated heuristics. To optimize maximum SRLG-disjoint path routing and wavelength assignment algorithm, a novel path weighting scheme was used. To improve the run-time performance of maximum SRLG-disjoint path greedy algorithm, it was implemented using NVIDIA CUDA heterogeneous parallel programming platform and executed on graphics processing unit.

Keywords: optical transport network, network reliability, network availability, network failure modeling, Monte Carlo simulation, network simulation, shared risk link group, routing and wavelength assignment, algorithm optimization, heterogeneous paralell programming

Prošireni sažetak

Brzi porast količine prenesenog prometa putem interneta, podržan isto tako brzim povećanjem kapaciteta optičke transportne mreže čini otpornost mreže na kvarove zahtjevom koji je potrebno uključiti u procesu dizajna mreže. Kvar mrežnog elementa (primjerice, vlakna u kabelu ili prospojnika u čvoru) može uzrokovati prekid mnogih svjetlosnih puteva, što vodi gubitku podataka i prihoda.

U slučaju kvara komponente puta koji koristi logički kanal u mreži, alternativni put (koji zovemo rezervnim) mora se koristiti sve dok sepopravak komponente osnovnog puta ne dogodi. Grupa veza s dijeljenim rizikom (shared risk link group, SRLG) je grupa veza u mreži koje dijele fizičku lokaciju. To može biti kabel, cijev ili izlaz na čvoru. Sve veze koje se nalaze u SRLG-u imaju mogućnost biti oštećene u slučaju kvara jedne veze koja se nalazi u SRLG-u. Takvo fizičkog oštećenje rezultira situacijom u kojoj višestruki logički kvarovi u mreži nastaju zbog jednog fizičkog kvara.

SRLG uvodi zavisnost između kvarova veza, obzirom da se radi o skupu veza koje dijele zajednički fizički resurs kao što je prijelaz mosta, kabel ili cijev. Česta je pretpostavka da je korelacija između kvarova deterministička, što implicira da kvar pojedine veze u SRLG-u uvijek uzrokuje kvar svih ostalih veza koje on sadrži. U stvarnosti to nije nužno slučaj, pa su izučavani vjerojatnosni modeli u kojima veze sadržane u SRLG-u doživljavaju oštećenje s određenom vjerojatnosti. Zaključeno je da je utjecaj koreliranih kvarova (uključujući SRLG-ove) na raspoloživost mreže značajan.

Brojni pristupi pružanja resursa putevima, specifično usmjeravanja i dodjele valnih duljina (routing and wavelength assignment, RWA) u optičkim mrežama koje sadrže SRLG-ove su razmatrani sa zajedničkim ciljem izbjegavanja istovremenog kvara osnovnog i rezervnog puta. RWA problem se može iskazati kao cjelobrojni linearni program. Obzirom da je općenito NP-težak, često se za rješavanje koriste heuristike.

Uvodno poglavlje iznosi kontekst problema koji se u radu rješava, objašnjava motivaciju za istraživanjem i navodi sadržaj rada po poglavljima.

Drugo poglavlje “Osnove optičkih telekomunikacijskih mreža” (Basics of Optical Telecommunication Networks) predstavlja temeljne pojmove područja optičkih mreža, počevši od optičkih kanala, preko čvorova u optičkoj mreži koji izvode obradu signala u električnoj domeni, sve do čvorova u mreži koji rade u sveoptičkoj domeni (tj. ne pretvaraju signal u električnu domenu). Poseban naglasak stavljen je na optičke prospojnike s mogućnošću rekonfiguracije, te s njima povezanu funkcionalnost za kontrolu i upravljanje mrežom.

Treće poglavlje “Temelji simulacije mreža” (Fundamentals of Network Simulation) iznosi osnove područja simulacije mreža zasnovane na diskretnim događajima. Poseban naglasak stavljen je na simulaciju Monte Carlo koja je uzeta u razmatranje kao jedan od mogućih postupaka za rješavanje problema proračuna raspoloživosti optičke mreže. Navedeni su i opisani zahtjevi za model optičke mreže i simulator koji će ga koristiti. Zahtjevi uključuju podršku za kanale s većim brojem valnih duljina, model optičkog komutatora, granularnost komutacije, različite arhitekture komutatora, kontrolnu ravninu i mehanizme zaštite i obnavljanja u slučaju kvarova na komponentama mreže. Izneseni su razlozi zbog kojih nijedan od postojećih simulatora ne zadovoljava navedene zahtjeve. Zbog toga je u ovom poglavlju predložen i razvijen novi simulator zasnovan na mrežnom simulatoru ns-3.

Četvrto poglavlje "Otpornost telekomunikacijskih mreža na kvarove" (Resilience of Telecommunication Networks) daje pregled područja otpornosti telekomunikacijskih mreža. Opravak nakon kvara u optičkoj telekomunikacijskoj mreži koji se koristi dalje u radu zasniva se na zaštiti puta i metodi zaštite s dodijeljenim kapacitetom. Također se definiraju metrike za proračun performansi raspoloživosti mreže, specifično raspoloživost.

Peto poglavlje "Korelirani kvarovi veza u mreži" (Correlated Failures of Network Links) daje prikaz rezultata simulacije Monte Carlo zas mreže u prisustvu koreliranih kvarova na mrežnim vezama. Koristi se metoda zaštite puta s dodijeljenim kapacitetom. Opisani su modeli veza u mreži, modeli grupa veza s dijeljenim rizikom, modeli svjetlosnih putova i logičkih kanala. Svi opisani modeli implementirani su u u mrežnom simulatoru ns-3. U scenarijima gdje je to moguće izvesti, rezultati izračuna raspoloživost mreže dobiveni simulacijom Monte Carlo verificiraju se usporedbom s rezultatima dobivenim analitičkim putem. Rezultati izvođenja simulacije zasnovane na predloženom modelu pokazuju da fizička svojstva grupa veza s dijeljenim rizikom, specifično njihova duljina, značajno utječu na raspoloživost mreže.

Šesto poglavlje "Utjecaj koreliranih kvarova na različite modele topologija" (Impact of Correlated Failures on Various Topology Models) analizira utjecaj grupa veza s dijeljenim rizikom na raspoloživost mreže za šest različitih modela sintetičkih topologija (slučajni geometrijski, Gabrielov, model relativnog susjedstva, model k-najbližih susjeda, Waxmanov i Barabási-Albertov). Rezultati simulacije pokazuju da se utjecaji koreliranih kvarova grupe veza s dijeljenim rizikom znatno razlikuju među analiziranim modelima.

Sedmo poglavlje "Optimizacija usmjeravanja i dodjele valnih duljina koja uzima u obzir grupe veza s dijeljenim rizikom" (Shared Risk Link Group-aware Optimization of Routing and Wavelength Assignmen) predstavlja postojeće pristupe usmjeravanju i dodjeli valnih duljina razvijene s ciljem maksimiziranja disjunktnosti grupa veza s dijeljenim rizikom. Predložen je i implementiran novi algoritam za rješavanje problema usmjeravanja i dodjele valnih duljijna. Predloženi algoritam koristi svojstva grupa veza s dijeljenim rizikom prilikom odabira rezervnih puteva. Cilj je poboljšati rezultirajuće raspoloživosti uspostavljenih svjetlosnih puteva. Verifikacija na studijskom primjeru mreže pokazuje da predloženi algoritam daje jednake ili bolje vrijednosti raspoloživosti mreže kod usporedbe rezultata s onima dobivenim izvođenjem postojećih algoritama. Predlažu se i raspravljaju, također, buduća poboljšanja predloženog algoritma.

Osmo poglavlje "Optimizacija performansi korištenjem heterogenog paralelnog programiranja" (Performance Optimization Using Heterogeneous Parallel Programming) analizira vremensku složenost predloženog algoritama za usmjeravanje i dodjelu valnih duljina. Algoritam je implementiran korištenjem metoda heterogenog paralelnog programiranja i izvodi se na grafičkim procesorima NVIDIA s tehnologijom CUDA (Compute Unified Device Architecture). Rezultati pokazuju da paralelizacija korištenjem grafičkih procesora značajno smanjuje vrijeme izvođenja, čak do sedam puta. Dan je kratak pogled u budućnost računanja korištenjem grafičkih procesora te se navode smjerovi budućih istraživanja.

Deveto, zaključno poglavlje rezimira iznesene rezultate i predlaže njihove primjene.

Izvorni znanstveni doprinosi doktorskog rada sastoje se u sljedećem:

- Model raspoloživosti optičke telekomunikacijske mreže koji uzima u obzir postojanje grupa veza s dijeljenim rizikom uz pretpostavku varijabilnih duljina koreliranih veza i proizvoljnih stupnjeva korelacije kvarova.
- Metoda proračuna raspoloživosti primjenom simulacije Monte Carlo zasnovana na predloženom modelu raspoloživosti optičke telekomunikacijske mreže u prisustvu koreliranih kvarova.
- Algoritam za usmjeravanje i dodjelu valnih duljina u optičkim mrežama s valnim multipleksiranjem koji optimira raspoloživost logičkih kanala obzirom na značajke grupa veza s dijeljenim rizikom, uz primjenu paralelizacije izvođenja korištenjem naprednih procesorskih arhitektura.

Ključne riječi: optička telekomunikacijska mreža, pouzdanost mreže, raspoloživost mreže, modeliranje kvarova mreže, simulacija Monte Carlo, simulacija mreže, grupa veza s dijeljenim rizikom, usmjeravanje i dodjela valnih duljina, optimizacija algoritma, heterogeno paralelno programiranje

Contents

Chapter 1

Introduction

> *"This 'telephone' has too many shortcomings to be seriously considered as a means of communication. The device is inherently of no value to us."* (Western Union internal memo, 1876.)

Western Union's early doubts about long distance telecommunication over the copper wire might sound ridiculous today. Fortunately, even back in the days, the doubts were not successful in stopping researchers and engineers from pursuing their ideas about how to develop communication technologies. The copper wire communication technologies developed over the century before the memo was crated. These technologies were eventually successfully used to create a country-wide telephone network, defying the memo in the process.

The optical fiber was already known in the beginning of the 20th century. Despite the knowledge, the first applications in telecommunications appeared more than half a century later. Improvements in attenuation were done during the 1970s and 1980s by doping optical silica glass with other elements, such as titanium, and compounds, such as germanium oxide [1]. Finally, with fiber attenuation decreasing sufficiently that tens of kilometers could be reached without repeaters, optical fiber replaced copper wires in long-haul networks.

The network society and the information age of today were enabled by communication infrastructure of the Internet, which is again enabled by its optical core [2]. The amount of Internet data traffic finally surpassed the amount of telephone traffic in the year 2000 [3], and has since been growing at about 100% per year [4].

Services such as high-definition multimedia, video calls, and games are used by many users over widely available broadband and fiber access networks. The number of users with broadband connection is relatively large even in developing countries. For example, in Croatia data from Croatian Post and Electronic Communications Agency (HAKOM) for year 2010 claims there are 1 132 212 users of broadband Internet, which is an increase of 20.8% compared to the year before [5]. With the rise of mobile devices such as smartphones and tablets providing users with high-speed wireless connection, the demands placed on the core network are huge.

These demands on the core network were met by capacity growth of optical transport networks. Optical networks of today can transmit tens to hundreds of gigabits per second per channel, and each fiber can carry dozens of channels assigned to different wavelengths. In Chapter 2 we give an overview of optical transmission network components.

Network research relies on experimentation. Since real-world optical network testbeds can be unavailable or too expensive to maintain, simulation models of real-world components are employed in experiments. We describe the approach to modeling and usage of simulation in optical network research in Chapter 3 and reference [6].

In Chapter 4 we turn our attention to network recovery in presence of failures. In case of a failure of a component of a path used by logical channel in the network, an alternative path (called spare path) has to be used until the component of working path is repaired. Schemes dealing with such challenges can be classified by time of operation, on protection and restoration, by routing type, on link-based and path-based schemes, and by criteria of spare resource sharing, on dedicated and shared.

Restoration-based schemes generally need longer recovery time than protection-based schemes, but give better performance in case of multiple time-overlapping failures of network components [3]. Path-based schemes offer better performance compared to link-based at higher run-time cost. To evaluate and compare different schemes, network performance metrics such as availability and reliability are employed.

Considering the bandwidth provided by a single fiber in the optical network, network resilience becomes an important consideration. A failure of any network component (e.g. a fiber or a switching element in network node) can cause outage for many lightpaths, and lead to user dissatisfaction and effectually decreased operator revenues. Correlated failures are particularly undesirable since they cause simultaneous failure of multiple logically distinct links or nodes. In Chapter 5 and reference [7] we study the impact of correlated failures of network links on network availability.

In the recent years, various topology models were studied with the goal of being able to produce synthetic networks that have properties of real networks. Impact of correlated failures varies depending on network topology. In Chapter 6 and reference [8] we study the impact of correlated failures on availabilities of networks having synthetic topologies generated by six different topology models.

Various approaches have been tried to avoid simultaneous correlated failures of working and spare path. Routing and wavelength assignment in presence of failure correlation between links is NP-hard problem, and a number of algorithms for solving it has been proposed. Building upon the studies done in Chapters 5 and 6, in Chapter 7 we propose a new routing algorithm based on a novel spare path weighting scheme utilizing the properties of shared risk link groups. The specific properties used are length and number of links.

Routing and wavelength algorithms in presence of correlated failures have high run-time complexity. In Chapter 8 and reference [9] we utilize GPUs to optimize performance and reduce run time of routing and wavelength assignment algorithms studied in Chapter 7.

Chapter 9 summarizes the results presented in previous chapters, suggests their possible applications, and concludes the thesis.

Chapter 2

Basics of Optical Telecommunication Networks

The bandwidth provided by optical fiber exceeds all other known transmission mediums. Optical fiber has total bandwidth of 25 000 GHz, compared to radio band that has a thousand times less (25 GHz) [10, 11]. Optical fiber also has other characteristics that are important for its usage as a communication medium, such as low attenuation [12, 13]. These features enable creation of optical networks in a cost-effective way and make optical communication feasible for large-scale deployment.

Optical fiber has been deployed in the infrastructure of high-speed networks that use it to connect geographically distributed network nodes.

This chapter is organized as follows. First we describe point-to-point links in optical networks, and then study optical networks which use electrical processing of data at nodes. Then we classify multiplexing, and finally turn our attention to all-optical networks. Particular aspects of all-optical networks are studied further in the following chapters.

2.1 Point-to-point Links in Optical Networks

During the early 1980s optical networks were built as point-to-point transmission systems. At the transmitting side, electrical signals containing data were converted to optical signal to be transferred over optical fiber; at the receiving side, the arriving signal is converted from optical to electrical for subsequent processing and storage.

For multi-hop communication between two nodes, multiple single-hop optical point-to-point links are used. These point-to-point links can be arranged to form various network topologies, most commonly ring (Figure 2.1) and star (Figure 2.2).

In case of star, a device called star coupler is used to combine all optical signals and distribute them across all output ports. Alternatively, networks using star topology can be built

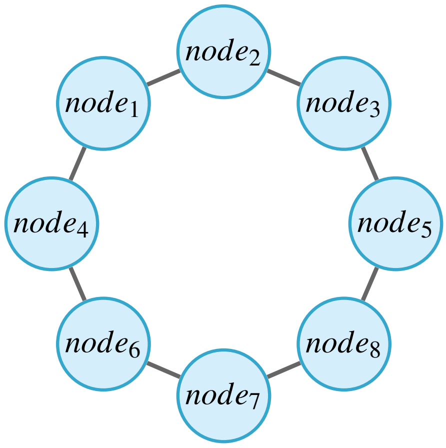


Figure 2.1: Example ring topology.

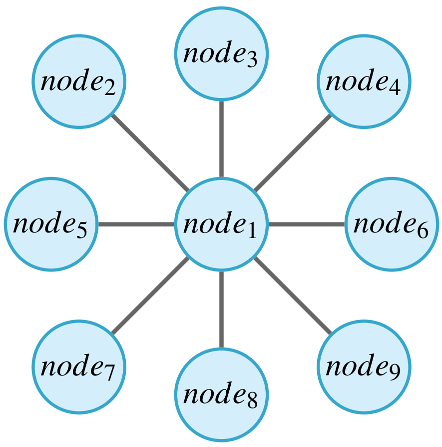


Figure 2.2: Example star topology.

using opto-electrical conversion and electro-optical conversion at the node in the center of the network. In rings, each node performs opto-electrical conversion on receiving side and electro-optical conversion on transmitting side. The combined opto-electrical and electro-optical conversion is usually referred to as opto-electro-optical (OEO) conversion.

2.2 Synchronous Optical Network and Synchronous Digital Hierarchy

Synchronous Optical Network (SONET) and synchronous digital hierarchy (SDH) are standards for optical point-to-point links. Both SONET and SDH were standardized in 1988 with the goal to allow interconnection of equipment from different manufacturers and carriers and provide new network features [14, 15]. Both standards define bit rates, frame structure, and procedures for network operation.

SDH is based upon a time division multiplexing (TDM) signal hierarchy. Time frame of 125 µs is periodically recurring and is able to carry payload traffic. Aside from payload traffic, SDH frame carries overhead bytes that are used for channel provisioning, network monitoring, and network maintenance.

The most common topology for SDH is ring with OEO conversion used at nodes. SDH rings have two main types of OEO nodes: the digital cross-connect system (DXC) and the add-drop multiplexer (ADM). DXC adds and drops SDH channels, while ADM aggregates or splits SDH traffic. The basic difference between the two devices is that DXC can be used to connect a larger number of links [11, 16].

2.3 Types of Multiplexing

Optical fiber has huge bandwidth, which is unlikely to be used by a single client, connection, or application. Instead, traffic of multiple sources shares available bandwidth by using a technique called multiplexing. Multiplexing allows dividing bandwidth across time, space, and wavelength. We discuss each of the three approaches in the following text.

2.3.1 Time Division Multiplexing

Time division multiplexing (TDM) has been used in traditional electronic network communication for as long as digital communication existed and can be used in optical networks as well [11, 17]. The basic idea is that each time interval can be divided into equal slots and each slot is provided to a different traffic source.

TDM line speed is aggregate of all traffic sources. When used with high-speed optical networks that do OEO conversion, TDM is limited by the speed of electronic transmitting, receiving, and processing technology.

2.3.2 Space Division Multiplexing

An approach to avoiding the electro-optical bottleneck of TDM is space division multiplexing (SDM). In SDM multiple fibers are used instead of one, and each of these can operate at any line rate. In particular, this rate can be the peak rate of electronics.

While SDM does well for local area and other short-distance communication networks, it is not suitable for long-distance transmission due to need to install and maintain multiple fibers between nodes.

2.3.3 Wavelength Division Multiplexing

Wavelength division multiplexing (WDM) is the approach that avoids the shortcomings of TDM and SDM. Wavelength division multiplexing (WDM) technology allows partitioning the large available bandwidth into a number of smaller channels. It is basically the frequency domain multiplexing applied to optical fiber frequency domain. The name WDM comes from the more frequent usage of wavelength term as opposed to frequency term in the field of optical networks. Speed of light c equals 299 792 458 m/s. Wavelength λ and frequency are f are related as

$$c = \lambda \times f.$$

Each of the clients sends the traffic on a different wavelength and the transmission of all the different wavelengths is done in parallel [18]. Each of the N transmitters is allocated a different wavelength $\lambda_i, i = 1, 2, \ldots, N$. At the transmitting side, the multiplexer gathers all the wavelengths and sends them into a fiber. On the receiving side, demultiplexer splits different wavelengths and sends each of them to a different receiver.

Unlike SDM, WDM does not require multiple fibers, and unlike TDM, any line rate can be used for each wavelength channel. These two features made WDM widely used by equipment manufactures and network operators. WDM is also an area of active research, as it has been for some decades [19, 20, 21]. The technology is advancing rapidly, the number of channels is increasing, and this progress has made WDM one of the key parts of future optical network environments.

2.4 Optical WDM Networks

Optical networks that avoid OEO conversion at nodes are called transparent; networks with OEO conversion are called opaque. In transparent networks, optical signal is received and transmitted without electrical processing. Some of the advantages of transparent networks are independence of bit rates and signal formats, and reduced power consumption compared to opaque networks. Some disadvantages of transparent networks are lack of implicit signal regeneration at nodes, which cause optical signal impairments to accumulate, more complicated network engineering, and difficult performance monitoring and fault localization [22]. Optical WDM networks can be transparent, opaque, or a combination of both. Networks that have transparent and opaque nodes are called translucent, and parts of such network that are transparent are called islands of transparency [23, 24].

In WDM networks fiber links carry multiple wavelength channels instead of only one. Such networks can consist of point-to-point links and have OEO conversion at each node. For example, in case of multihop transmission with OEO conversion at intermediate nodes, source and destination nodes are not able to choose line rate, modulation format, and protocol for communication. Instead, the intermediate nodes make the choice and the network is not transparent. As another example, consider a network with star topology where star coupler is an optical device that does not perform OEO conversion and instead receives and transmits in the optical domain. End nodes in this network can communicate using protocol, modulation format, and line rate they choose, so the network is transparent. This approach is used in broadcast and select networks in which the central star coupler broadcasts all incoming wavelengths to all attached receiving nodes [25, 26]. Each receiving node has an optical filter that selects a single wavelength from the band and reads it for processing; other wavelengths are dropped.

2.4.1 All-optical Networks

Intermediate nodes in a WDM network can be configured to electrically process only a small subset of wavelengths, and forward others in the optical domain. Therefore, data sent from the source using wavelengths that remain in the optical domain will remain so until the destination, enabling transparency in the network. Optical WDM networks with such bypassing capability at intermediate nodes are named all-optical networks (AONs) [27, 28, 29, 30]. AONs have been successfully used in building local, metro, and wide area networks.

AONs use optical circuit switching (OCS), and intermediate nodes switch circuits at wavelength channel granularity. Because of such granularity, AONs that use OCS are called wavelength-routed optical networks [31, 32]. In wavelength-routed networks, optical circuits and wavelength channels are equivalent. All-optical nodes in AONs are called OOO nodes to emphasize that no conversion from optical to electrical domain is done on nodes, which is not the case when OEO nodes are used.

Despite operating in a different domain, AONs have the comparable functions as SDH networks. Both of them are circuit-switched systems. Multiplexing, switching and processing of wavelength channels in AONs has the same role as multiplexing, switching and processing of TDM time slots in SDH. More concretely, in SDH lower-speed channels are multiplexed into higher-speed signal, and SDH signal is able to carry a combination of different data rates and traffic types. ADMs and DXCs allow SDH to access and manipulate individual channels. AON has comparable functions: OOO nodes are ADM and DXC ported to optical domain by replacing electrical parts by optical components. Optical device comparable to electrical ADM is called optical add-drop multiplexer (OADM) and optical device comparable to electrical DXC is called optical cross-connect (OXC) [33, 34]. OXC is also found in the literature by the name wavelength ADM (WADM) and wavelength-selective cross-connect (WSXC) [35].

OADM with single input and single output fiber works as follows. At the input fiber signal containing M wavelengths $\lambda_1, \lambda_2, \ldots, \lambda_M$ is amplified by an optical amplifier, usually Erbium doped fiber amplifier (EDFA) [36]. After amplification, signal is partitioned into M separate wavelengths using $1 \times M$ demultiplexer. First K wavelengths in demultiplexer are bypassing the node, and they are sent straight to the multiplexer. Remaining $M - K$ wavelengths are dropped at the node, meaning that they are converted from optical to electrical signal for processing in electrical domain. Since there the node now has $M - K$ wavelengths unused at the multiplexer, it can add local traffic by converting it from electrical signal to optical signal and sending it to applicable multiplexer ports. Multiplexer combines all $K + (M - K) = M$ wavelengths into one signal and sends them on a fiber. Another optical amplifier, usually EDFA, is used on output fiber to amplify the outgoing signal.

OXC with N input and output fiber links uses N demultiplexers and N multiplexers. Again, each fiber carries M wavelengths $\lambda_1, \lambda_2, \ldots, \lambda_M$. Multiplexers are used to split the signals into individual wavelengths, which arrive to M space division switches (one switch is used for each wavelength). Space division switches direct the light to multiplexers, which multiplex the wavelengths and send them on the output fiber. Additionally, OXCs provide restoration features in case of network failures. They are able to reconfigure routing to accustom for failed link or adjust for increased amount of traffic.

An AON that uses both OADMs and OXCs is called an optical transport network (OTN) [16, 37, 38]. AONs, and OTNs in particular, support various services and applications. Services can demand high-speed point-to-point or point-to-multipoint channels. Applications include voice and video, multimedia streaming (such as HDTV broadcast), medical imaging, data storage over the network, high performance computer interconnects, and others [39, 40]. Due to the transparency, it is possible to support all these applications in a cost-effective way. However, large transparent AON is not feasible due to fiber nonlinearities and crosstalk between wavelengths that limit the distance that a signal can travel before it needs to be regenerated and also number of wavelengths that can be used for transmission [41, 42]. Therefore, it is possible to and might be necessary to split large networks into islands of transparency interconnected by nodes doing OEO conversion.

Major design goals of AONs include scalability and modularity [27, 43]. Scalability is the ability to increase network size to offer network services to arbitrarily large number of users. Modularity is related to scalability; it is the ability to add only one more node when increasing network size. Additionally, AONs are designed to support wavelength reuse, which allows same wavelength to be used in multiple locations in the network as long as the paths using the wavelength never overlap [29, 30]. When wavelength reuse is available, bandwidth is used more efficiently. This in turn increases effective network capacity and decreases cost. Significant progress has been done towards both modularity and scalability of AONs [27, 44].

IN AONs, the path between the source and destination nodes will remain entirely in the optical domain. The path in AON is for that reason called lightpath [45, 46]. Lightpaths can be generalized from single-destination to multi-destination transmission; in this case they are called light-trees [47]. Lightpaths and light-trees can be optically amplified along the way. Both of them can keep the initial wavelength along the entire path or change wavelength one or more times. If wavelength has to remain unchanged, then the setup of lightpaths (and light-trees) in AON is said to satisfy wavelength continuity constraint. If this constraint is enforced, on average it will be harder to set up lightpaths. More precisely, the blocking probability of new lightpaths is increased.

2.4.2 Wavelength Converters

To decrease blocking probability, OXCs in the network can be equipped with wavelength converters; such OXCs are called wavelength-interchanging cross-connects [35, 48]. Since wavelength continuity constraint can be excluded, blocking probability is decreased which in turn makes network more flexible.

Wavelength conversion can be fixed, limited range, or full-range. Fixed wavelength conversion always converts input wavelength λ_i to output wavelength λ_j. Limited-range wavelength conversion supports converting input wavelength λ_i to a subset of output wavelengths. Full-range wavelength conversion removes the restriction and allows input wavelength to be converted to any output wavelength. Since wavelength converters are expensive, costs of building an AON can be decreased by equipping only a subset of nodes with wavelength conversion. This approach to wavelength conversion is called sparse wavelength conversion [49, 50]. Wavelength converters can also be shared per node or per link to increase efficiency and decrease costs.

Wavelength conversion can be implemented by doing OEO conversion, or entirely in the optical domain by exploiting fiber nonlinearities [51, 52]. The benefits of wavelength conversion remain the same in both cases.

2.4.3 Reconfigurability

A particular type of wavelength converters are tunable wavelength converters (TWCs), which allow operation on several output wavelengths instead of only one. Just like ordinary wavelength converters, TWCs can be implemented as all-optical or opto-electrical devices. Deploying TWCs allows dynamic reconfiguration of the network to adjust to variances in traffic and other conditions such as failures. Reconfigurability enables rerouting and is therefore considered a favorable feature.

To illustrate the benefits of reconfigurability, we will once again take a look at OADMs. In ordinary OADMs the optical add, drop, and bypass paths are fixed. Therefore, ordinary OADMs are static; they can add and drop a particular set of wavelengths and are unable to reconfigure this predefined set. On the other hand, reconfigurable optical add-drop multiplexer (ROADM) can be made using optical switches. Optical switch is a very simple device with two inputs and two outputs, which can forward input signal to output in two ways, cross and bar. In cross mode, first input port gets forwarded to second output, and second input gets forwarded to first output. In bar mode, first input port gets forwarded to first output, and second input to second output. Cross-bar mode is switched by the electronic control. In ROADM the bypass is connected to first input port of the switch, and local add is connected to the second input. On the output side, situation is analogous: the first output is connected to bypass, and second output is connected to local drop. Since all switches are controlled independently of each other, electronic control can now change mode of a particular switch to change which input port gets dropped, which local add gets added instead, and which ports are forwarded. ROADM architecturs are actively researched [53, 54, 55]

OXCs can be made reconfigurable in a similar way to ROADMs, by electronically controlling the space division switch. Such OXC is called reconfigurable optical cross-connect (ROXC) and can adapt to failure conditions and variance in traffic demands by changing input to output cross-connections. With reconfigurability as an option, network optimization for a particular traffic demand and failure scenario becomes a problem. From here onwards we will consider all OXCs to be reconfigurable OXCs.

2.4.4 Network Control and Management

Network control and management has to be integrated into AONs to make reconfigurable networks commercially feasible. Network control is used to set up, modify, and tear down optical circuits (lightpaths and light-trees) in the optical network by reconfiguring ROADMs, OXCs, wavelength converters, and tunable transmitters [56]. Management functions are monitoring, detection, isolation and diagnosis of network failures, and triggering restoration mechanisms to mitigate the link and node failures. Survivability in presence of network failures is considered to be very important aspect of AONs, along with scalability, modularity, reconfigurability, and transparency [56, 57].

We already talked about control information in SDH networks being carried in frame overhead bytes. Unfortunately, the same approach can not be used in transparent optical networks since optical bypass can be used on intermediate nodes, so the control information would just pass through the node that should receive it. For that reason, in transparent optical networks a particular wavelength is reserved for control and management purposes. This wavelength channel is called optical supervisory channel (OSC), and it is converted from optical to electrical domain, processed, and then converted to optical domain for transmission. The conversion and processing is done at each node. OSC is used to exchange control and management information among network nodes. For example, OSC can be used to configure a tunable transmitter to change its wavelength to λ_j from λ_i, or change a cross-bar switch from bar to cross in a ROADM.

Network control can be centralized or distributed. When centralized control is used, each connection request is processed by a single controller, which then decides how to set up lightpath and sends configuration messages using OSC [58]. Distributed control places a control unit at each node. Lightpath setup and teardown is done in a coordinated way by exchanging the messages between these control units. In large networks distributed control is desired due to better scalability and modularity.

Network management system (NMS) maintains a global view of present network status by issuing queries to network elements and processing their replies. Network elements such as OADMs and OXCs receive those queries over OSC, and use the same channel to send replies. NMS uses the information it receives to update information about configuration, link and node status, and resulting topology of the network. In case of failures, NMS has the option to use the information about network state to initiate set-up or tear-down of end-to-end lightpaths.

For management of transparent reconfigurable AONs, a framework named Telecommunications Management Network (TMN) has been standardized by International Telecommunication Union Telecommmunication Standardization Sector (ITU-T) and International Organization for Standards (ISO) [59, 60, 61]. TMN includes planning, provisioning, installing, maintaining, operating and administering networks. TMN consolidates a large number of standards that relate to network management in a way called FCAPS model. Management subjects covered by FCAPS are Fault, Configuration, Accounting, Performance and Security Management.

Fault management implies monitoring network equipment and detecting fault conditions, informing NMS about alarms, and configuring restoration mechanisms. Examples of parameters that can be monitored are optical signal power and SNR, which can be used to assess the quality of established lightpaths. When fault conditions are detected, network element generates an alarm notification. Fault conditions include cable cuts, cross-bar switch failures, and software errors.

Configuration management equips the network with connection set-up and tear-down capabilities. Two paradigms for connection set-up and tear-down are used: management provisioning and end-user signaling. When management provisioning is used, network administrator does connection set-up via a management system interface. Such connections are expected to have a relatively long life-span. On the other hand, end-user signaling is used for setting up connections of shorter life-span used for low latency transport of traffic bursts. In this case, signaling is initiated by an end user using a signaling interface without interference of network management system. Connections in optical network are also referred to as logical channels.

Accounting management records network resource usage by clients and charges respective accounts. Security management implies protection of network (including management system) from unauthorized access.

2.5 Chapter Conclusions

We described the optical networks starting with point-to-point links, then discussing SDH as an optical network in which the nodes operate in electrical domain, and finally described AONs. We discussed advantages and disadvantages of AONs over SDH. Transparency of AONs removes the possibility of monitoring digital signal in electrical domain. Therefore, some failures might be hard to detect and isolate.

As an example of hard to detect failure, consider the following. It is possible to detect an OXC cross-connecting inputs and outputs in a wrong way by purely optical monitoring if wavelengths are placed on the output ports in an incorrect way. However, if wavelengths are placed correctly to output ports, but the (digital) information carried by wavelength is not correct, this is undetectable by optical monitoring. This, along with fiber impairments for long-distance transmission, suggests maintaining reasonably sized islands of transparency as a rational solution to this problem. Besides enabling failure detection, reasonably sized islands of transparency reduce time to failure isolation.

Chapter 3

Fundamentals of Network Simulation

Computer simulation is one of the three different methods for performance evaluation of systems (including telecommunication networks), other two being mathematical analysis and real-world measurements [62]. Each of these methods has its own advantages and disadvantages. Recommendations and best practices for using each method on a particular class of problems can be found in the literature.

To analyze performance of a system using real-world measurements one has to implement it first. On the other hand, for mathematical analysis and computer simulation, a model is used instead. Obviously, in terms of cost and effort computer simulation and mathematical analysis have the advantage compared real-world measurements. However, these methods are not competing and each of them has its uses in research.

Due to the fact that with increasing complexity of the system mathematical analysis becomes intractable, computer simulation is very often used instead. Simulation can help both for comparison of different design alternatives of a system in development or for optimization of an existing system design.

The rest of this chapter is organized as follows. Section 3.1 classifies simulation and describes discrete-event simulation in detail. Section 3.2 describes specifics of optical network simulation. Section 3.3 compares available simulation software. Section 3.4 lists requirements for optical WDM network model to be implemented inside a network simulator. Section 3.5 gives high-level overview of our WDM network simulator design. Section 3.6 describes the simulation of a circuit-switched optical network in case of failures. Section 3.7 concludes the paper and lists possible directions for future work.

3.1 Simulation Types and Modeling for Simulation

Many different types of simulations exist: discrete-event simulation (also called event-driven), continuous simulation, Monte Carlo simulation, trace-driven simulation and others [62]. In case of computer and telecommunication network simulation the most used method is discrete-event simulation [63]. Unlike continuous simulation, state of an entity in a discrete-event simulation can change only at discrete time points which are named events.

Discrete-event simulation has been used for research on all layers of computer networks, from physical, over link, network, and transport, up to application layer. There are two key advantages of this type of simulation. First, it fits computer networks very well, and second, it is very easy to use.

3.1.1 Discrete-event Simulation

An entity is an abstraction of a particular subject or object in real world. An entity consists of attributes. For example, an entity network link could have attributes physical length, physical medium and bandwidth. A specific instance of an entity is called object, i.e., one can consider an entity to be a template for any number of objects with same attributes but with potentially different values for each attribute.

A system is a set of entities and their relationships. For example, a network may be considered as a system of entities that are nodes and links, and relationships describing the incidence of nodes and links. In discrete-event simulation, change of a system state is triggered by an event. In the case of network simulation, an event can be anything from a packet getting dropped on reception at link layer, an application doing POST request over HTTP, IP updating the routing table, and a network card sensing physical cable got disconnected.

Systems we study are often very large and complex, so we build models. As we already said, a model is an abstraction of the system, and it consists of *selected* entities of the system and *selected* relationships between these entities. This approach to building models implies that simulation entities will always be simpler than the real world subjects and objects and therefore results obtained using simulation should always be considered to be approximations of those one would obtain using real-world measurements.

Central idea of discrete-event simulation is jumping from event to event in the increasing time order [64]. The simulator maintains a queue of events ordered by simulated time in which they will occur. The simulator then reads the queue and creates new events resulting from execution of each event. We should make a distinction between simulated time and real time here; it is not very important whether the simulated time goes faster or slower than real time, or even if they match exactly*. Data produced by the simulation is accessed and processed after the simulation is done executing and used for producing conclusions.

All discrete-event simulators share the following components [62]:

*There is a particular type of network simulations that interact with real world networks, often categorized as network emulation. Simulations of such type are done in a way that simulated time is synced to real time, but they are outside the scope of this thesis. More information can be found in [65].

- system state – set of state variables,
- clock – current time,
- future event list,
- statistical counters – set of variables containing statistical data about system performance,
- initialization routine – a routine that sets the clock to 0 and does model initialization,
- timing routine – a routine that retrieves next event from event list and advances the clock,
- event routine – a routine called when a particular event occurs, also called event handler; it differs from event to event, and might schedule more events to occur in the future.

Simulation lasts a certain amount of simulated time. That amount can be fixed predefined by user or can depend on some condition being fulfilled. In case of predefined stopping time, a user can set simulation to end at time 100 seconds. In case of predefined stopping condition, user can set simulation to end once 1 000 000 bytes have been received on a particular node. Finally, a simulation can simply end because no future events are scheduled, that is, stop in and of itself. In the last two cases amount of time that will be simulated is not known in advance and may significantly vary depending on model characteristics and input parameters.

Simulation runs are classified into transient and steady state simulations [62]. A transient run terminates after a defined simulation time or after some condition has been fulfilled. On the other hand, steady state run is not expected to terminate and is used to study long-term behavior of the system, i.e., when initial conditions no longer influence simulation results. The hard task is figuring out the proper simulation run times for steady state runs.

3.1.2 Modeling for Network Simulation

Model of the system we want to study is based on simplifications and assumptions. Since model is an abstraction, and there are multiple ways to abstract any system, our concern here is how to measure quality of a model and how to obtain a good model. Both concerns can be addressed by aiming for models that are [62]:

- simple – model should serve the purpose of the evaluation and it should implement only the features required for the evaluation,
- credible – model should be validated against real-world system it describes,
- documented – assumptions and simplifications in the model should be written down, and documentation should follow the evolution of the model.

Modeling in case of computer simulations has some additional concerns to be addressed. Since computer simulations use models implemented in software, additional constraints arise and might pose a significant source of error. Therefore, a model that aims for a software implementation should also satisfy the following [62]:

- efficiency – implementation should be done in a way that simulation duration is feasible, i.e., simulation execution does not last 2 years of real world time,

- verification – implementation should be verified, i.e., checked to match the model itself,
- code quality – consistent coding style, use of object oriented programming, etc.
- availability – implementation should be made available to other researchers to study, validate, verify and potentially improve and expand it†.

In addition, there is a trade-off between model reusability and universality on one side and simplicity on the other. While both sides should be taken into account, it is believed that a good model in general does not have to be universal nor generally reusable [62]. However, reusability in general reduces development time, so one can in practice often find model implementations that are made to be expandable and reusable.

Building a good model is one of the hardest steps in computer simulation, and the approach to do so varies with different systems one decides to model. In addition to experience with modeling various systems, to obtain a good model one should have deep knowledge of the system under study.

Furthermore, one should ensure that approach to performance evaluation is consistent with the goals of performance evaluation study that is to be done with the simulation. Performance evaluation study based on doing simulations has the following steps [62]:

1. Problem formulation and definition of system/model,
2. Choice of metrics, factors, and levels,
3. Data collection and modeling,
4. Choice of simulation environment, model implementation, and verification,
5. Validation and sensitivity analysis,
6. Experimentation, analysis and presentation.

This process is iterative due to unexpected problems that can arise at each of the steps.

Common practice can differ from these steps. Namely, it is quite often very hard to validate a simulation model. Computer simulation can be applied at a point when no particular design of a system has been chosen and in effect system has not yet been built. In this case there is no data available from real system to validate simulation models. In such case third method is still an option; one can try to obtain results from mathematical analysis of the system, but in many cases this approach does not lead to sufficient level of validity. This is the reason why simulation results are less authoritative than results achieved by mathematical analysis or real-world measurements.

Validation concerns are also relevant when reusing existing code. There is always a possibility that reused simulation model is based on assumptions that do not fit the current problem unless the code being reused is checked and evaluated thoroughly before reusing it. However, such effort invested in checking and evaluating eventually pays off with shorter development

†Note that while this requirement implies that implementation source code should be made available, it does not necessarily imply that implementation source code should also be made free software. In fact, as we will see below, not all open source network simulators are free software.

time for model implementation.

3.1.3 Specifics of Network Simulations

Network simulations model only the events that are related to data transmission, namely creation and processing of packets that are exchanged by the hosts interconnected by a communication network [66]. Examples of such events are expiration of a TCP timeout, start of data reception at physical layer, and link failure.

3.1.4 Monte Carlo Methods

Monte Carlo methods are computational algorithms that use a large number of random samples as input parameters to obtain numerical results. Monte Carlo methods have found their applications in network simulation where multiple simulations are ran and averages of output results are computed. We will use the term Monte Carlo simulation to refer to a simulation using Monte Carlo methods, i.e. random sampling, multiple runs, and averaging.

To illustrate the benefit of using Monte Carlo method in network simulation, we will consider the following scenario: a network has a sender and receiver node connected using point-to-point link with packet loss of 10%. We want to study impact of this loss on TCP congestion control algorithms. If only a single simulation is ran, the specific random numbers that decided which particular packets will be dropped will affect results. However, if multiple simulations are ran and averages of output results are computed, the specifics resulting from particular choice of random numbers will be reduced.

3.2 Simulation of Optical WDM Networks

To achieve optimal working of optical WDM networks, considerable research activity is needed. Simulation can help here by providing researchers with a cost-effective method to study and compare the behavior of proposed algorithms.

On the other hand, a lack of single uniform simulation platform for optical WDM network simulation makes it very difficult for researchers and engineers to compare results. Namely, model specifics of different simulators can lead to significant differences in results. Furthermore, disparate sets of features provided by different simulators and lack of integration usually limit research possibilities.

To address this issue, a simulator named Optical wavelength division multiplexing (WDM) network simulator (OWns) [67, 68] was developed by extending ns-2 simulator [69]. OWns models key features of WDM networks, including optical switching nodes, multi-wavelength links, routing and wavelength assignment (RWA) algorithms.

A simulation tool is required for research purposes that can extended into various fields of optical network research as needed. We found that none of the existing simulators, aside from OWns and some commercial solutions, had any of the required feature functionality. Also, we decided to avoid commercial solutions due to reasons we describe in detail below, and we found OWns to be based on outdated simulation platform with limited extensibility.

Finally, we developed a new model for optical network in network simulator ns-3 [70] and named it Photonic WDM Network Simulator (PWNS)[‡]. In the early stages of the project, we used the word *Prototype* in place of the word *Photonic* [6]. In the next section we explain why have we selected ns-3 among other available simulation platforms.

3.3 Overview of Network Simulation Software

We did not consider proprietary network simulators due to limited use conditions, limited extensibility, and also license cost. In addition, proprietary solution can bring us to vendor lock-in situation. Such lock-in occurs when one has to do forced software upgrades to continue using the simulator when the previous version goes out of support. Also, we would have to trust the vendor to provide interoperability with other software and allow exporting of data in open formats. This was not acceptable for us.

We wanted a solution that is extensible and freely available to researchers and engineers. In addition, we wanted a solution that is free and open source software so researchers working in various subfields of optical network research can extend the model to fit their performance evaluation needs. In this approach, by accepting outside contributions, the simulator could potentially provide a large spectrum of very specific functionality.

We evaluated the possibility of developing own simulator, in terms of feature functionality somewhat similar to COSMOS [71] but more extensible. This approach where we would implement the entire simulator ourselves would give us in depth knowledge of the software, albeit at the cost of additional work. Also, to make such development feasible, we would limit ourselves to simulating network layers from L3 to L1. However, this approach has some disadvantages as well. Aside from additional implementation work it would also limit simulation to optical networks. Since we would like to support researching networks that are partially optical on the physical layer and in parts use other technologies such as wireless, we decided to extend an existing open source network simulator.

We thoroughly analyzed open source network simulators ns-2 [69], OMNeT++ [72] and ns-3 [70, 73, 74]. We describe our findings now.

[‡] Since "pwn" is a leetspeak slang term derived from the verb "own", the name PWNS can be considered a pun on OWns. We leave it to the reader to judge whether or not this was intentional.

3.3.1 ns-2 Network Simulator

Funding of ns-2 development has decreased a lot in the last decade, and this has resulted in decreased integration of additional models developed by network researchers into mainline ns-2. Today there are many incompatible (and therefore incomparable) models with various features that can be found on the Internet, and many of them also depend on specific version of ns-2 so it is not realistic to expect integration into mainline anytime soon.

Aside from these concerns, ns-2 has a bunch of other design limitations [75]:

- split object model, using C++ and object-oriented Tcl (OTcl) [63, 76],
- relatively high amount of abstraction in network layer and below increases the difficulty in connecting simulation and real world,
- lack of additional simulation tools, for example, steady-state simulation detector,
- lack of model validation, and
- lack of model documentation.

We did not further consider using and extending ns-2 because of the limitation listed. Additionally, we expect that transition to ns-3 in network research community will continue over the coming years. Despite these facts, it is worth mentioning OWns [67] [68] variant that implements WDM network model. However, OWns is no longer developed and its source code is not officially available anymore.

While looking for a simulator to base our work on, our main criteria were:

- it already has a model for optical transmission network,
- it supports analysis of network reliability and availability,
- it provides an extensible architecture, and
- it has high performance in terms of execution time duration and memory requirements.

3.3.2 OMNeT++ Network Simulator

On the first point, there exists passive optical network model for OMNeT++ [77]. However, this model does not fit our requirements; it is model of access network, and our research interest is in core network. Aside from that particular model, OMNeT++ has some of the drawbacks of ns-2 and some of its own:

- component model is similar [66],
- its architecture is bilingual, using custom language called NEtwork Description (NED) and C++, and
- it is tightly integrated in its IDE, implemented as an Eclipse IDE [78] plugin.

Finally, OMNeT++ uses Academic Public License that prohibits commercial use. We consider this choice of license a major obstacle because it eliminates the possibility of interested companies using our code and contributing implementation of additional features. Taking all these points into account made us look for another simulator.

3.3.3 ns-3 Network Simulator

Ns-3 network simulator was designed and written from scratch. As shortcomings of ns-2 come largely from its design decisions, it was impossible to resolve them and at the same time keep compatibility with existing simulator core and already developed models. During ns-3 development ideas and parts of code were taken from GTNetS, yans [79], and ns-2 simulators.

The development of ns-3 network simulator was supported by French National Institute for Research in Computer Science and Control (Institut national de recherche en informatique et en automatique, INRIA) and American National Science Foundation (NSF). The goal of the project was to create a tool that will be developed by the academic community and companies even after the initial funding dries up. To achieve that, ns-3 Project created a community of maintainers, people responsible for a certain part of simulator code.

The infrastructure behind ns-3 development was set up so that any interested person can join and contribute, either by further developing existing models, or by creating new ones [80]. The entire ns-3 code is available under GNU General Public License (GPL), version 2.

Ns-3 simulator is based on discrete events. Simulated time is represented using integer type to avoid problems with portability on different processor architectures and operating systems [66]. Size of data type that is used to represent a moment in simulated time is 64 bit. This allows simulating 584 years with nanosecond precision. Time arithmetic is implemented using a 128 bit integral type: 64 bit is used for integer, and 64 bit for fractional part. All the operations required for time data type (addition, subtraction, multiplication, division, comparison) are implemented using only operations on integer data types to ensure computational consistency across machines of different architectures.

Simulator is single-threaded by design. Multi-threaded simulator was evaluated, but because smart pointers are used for automatic garbage collection, it happens that multi-threaded variant performs slower than single-threaded [81]. Our primary interest is reliability analysis of optical telecommunication network, where one has the option to run multiple independent simulations, so this is not a problem.

Network simulator ns-3 is a C++ library that can be compiled using GNU Compiler Collection (GCC) [82] or Clang compiler [83] on Linux, FreeBSD or Mac OS X operating systems. On Windows operating system a virtualization software such as VirtualBox can be used.

Network simulation descriptions in ns-3 are C++ programs using the ns-3 library. Python bindings can be used for writing simulations instead of C++ if one desires. This simplifies writing model prototypes and allows ns-3 to be used in combination with other Python scientifc software such as NumPy [84], SciPy [85] and matplotlib [86].

3.3.4 Random Number Generation

Ns-3 contains pseudorandom number generator MRG32k3a [87]. MRG32k3a generator offers $1.8 \cdot 10^{19}$ independent sequences of random numbers, each containing $2.3 \cdot 10^{15}$ subsequences. Each subsequence has period $7.6 \cdot 10^{22}$. Period of the entire generator is $3.1 \cdot 10^{57}$. Other random number generators, such as Mersenne twister [88] with total period $2^{19937} - 1$, can be used if necessary.

3.3.5 Performance Comparison

Despite the fact that among these three described simulators ns-3 has demonstrated the best overall performance [89], both ns-3 and OMNeT++ are capable of carrying out large-scale network simulations in an efficient way, with ns-2 exhibiting longer simulation run time in simulations consisting of a few thousand simulation nodes.

3.3.6 Extending Ns-3

Up until now, ns-3 lacked a model for optical transport network components. None of the models contained in other simulators were found to be and adequate fit for our research requirements. Flexible architecture of ns-3 simulator motivated us to consider extending it with optical network model of our own development. Concepts and ideas in already mentioned solutions for other two simulators can be useful as a pointer in certain direction.

3.4 Model Requirements

We evaluated the present feature functionality of ns-3 network simulator. To achieve our goals, we had to identify the specific areas where it was to be extended to support simulating optical WDM network. We considered the differences between the networks that have existing models in ns-3, which operate almost entirely in the electronic domain, and the optical WDM networks, which operate in both the optical and electronic domain.

Some of the requirements outlined in [67] apply to almost any optical network simulator. Specifically, the following is required:

- **Multi-wavelength Channels:** Optical WDM technology uses multiple wavelengths for data transmission over a fiber link. The support is needed for simulating the usage of both coarse and dense WDM technologies. Additionally, the support for both unidirectional transmission channels and bidirectional transmission channels is required.
- **Optical Switch Devices:** Models for devices in the optical network should include devices that act as switches with varying degrees of wavelength conversion capabilities.

- **Switching Granularity:** The model has to support various degrees of switching granularity. For example, it has to support switching at fiber level, at wavelength level, and at sub-wavelength level [90].
- **Switching Paradigms:** Model has to support Optical Circuit Switching (OCS) and leave open the possibility to implementat other switching paradigms such as Optical Packet Switching (OPS) and Optical Burst Switching (OBS) [91].
- **Switching Architectures:** The model has to allow detailed specification of interconnections of switch device parts to allow performance evaluation of different switching architectures, such as architecture on demand [53].
- **Control Plane:** The model has to implement a control plane to be used for routing, resource reservation, failure recovery etc. The control plane should operate in a centralized way and optionally allow for a distributed implementation. If possible, implementation of a control plane should provide an interface for reusing already existing solutions such as OpenFlow [92].

Once these components are implemented, optical WDM network research and engineering community will be provided with a simulation tool it can use. Networks that had to be studied using real-world measurements can now also be studied by simulation method using a network simulator. This is specifically interesting to subfields such as multilayer recovery[93]. In this particular subfield the possibility of using simulation in research depends heavily on having optical WDM network model implemented inside a network simulator. This dependence is because a network simulator such as ns-3 implements the entire layer stack.

3.5 Modelling the Optical Transmission Network

Inherent similarity between models of various types of telecommunication networks (including optical networks) suggests that approach that involves adding reusable features or functional parts to a component is more appropriate than implementation of the whole solution in one large monolithic model at once. Model based on components is easier to develop, test, verify and validate, because the implementation of feature functionality can happen iteratively, first implementing a feature and then testing the implementation. In addition, code reusability inherent in object-oriented design reduces the time needed to develop a similar model.

Components of ns-3 are modules, which consist of one or more classes which together make one or more models of real world communication devices, communication channels, network protocols etc. Abstract base classes used implemented by every model of a physical network in ns-3 are `NetDevice` and `Channel`. `NetDevice` describes a network interface card at a network node. `Channel` interconnects two or more network cards and contains delays, losses etc. Models of real-world network devices (e.g. WiFi, WiMAX, LTE) usually separate PHY layer of the network card from its MAC layer to allow combining various MAC devices and PHY devices and facilitate code reuse.

3.5.1 Models for Optical Network Devices and Channels

Our optical network model implementation consists of models for common optical transport network components: edge network devices (class `OtnEdgeNetDevice`), core network devices (classes `OpticalCrossConnectNetDevice`, `OpticalMultiplexerNetDevice` and `OpticalDemultiplexerNetDevice`), physical interfaces (class `OpticalPhy`) and channels (class `OpticalFiberUnidirectionalChannel`).

Classes that implement the model of optical telecommunication network are divided in ns-3 modules `optics` and `otn`; models for generic optical components are in the first module, while the second module contains models specific for Optical Transport Network (OTN). We describe the most used ones.

`OpticalNetDevice` and `OpticalFiberChannel` are abstract base classes that have features common to all optical network devices and channels. For network devices, this includes receive error model, lists of physical interfaces and elements needed by ns-3. For channel, this includes propagation loss and delay models.

`OpticalNetDevice` class is used by `OpticalPassthroughNetDevice` abstract base class and `OtnEdgeNetDevice`. `OtnEdgeNetDevice` is network device used at the edge of optical transport network does conversion from electrical to optical signal on transmission and from optical to electrical signal on reception, as well as adding and removal of OTN headers.

`OpticalPassthroughNetDevice` is used as a base class for classes that model behavior of optical network devices that signal passes through in some way. (Class hierarchy can be seen in Figure 3.1.) Examples of such devices are multiplexers (`OpticalMultiplexerNetDevice` class), demultiplexers (`OpticalDemultiplexerNetDevice` class), and optical cross-connects (`OpticalCrossConnectNetDevice` class).

All of these devices share common code for physical interfaces, implenented in `OpticalPhy` class, modeling physical reception and transmission interface (depending on how it is used). A diagram representation of a simple example that shows the relation between physical interfaces, network devices and channels can be seen in Figure 3.2.

`OpticalPhy` is a physical interface between the device and the channel, that does signal transmission and reception. It supports using ITU-T dense WDM and coarse WDM grids, and dense WDM grid is used by default. Optical signal is modeled by utilizing the functionality provided by ns-3 module `spectrum`, adapted slightly to fit our requirements. Namely, `spectrum` module provides `SpectrumChannel` and `SpectrumPhy` base classes, from which `OpticalFiberChannel` and `OpticalPhy` are derived (respectively). However, it assumes that a physical interface has an antenna model, and this does not hold true for optical networks

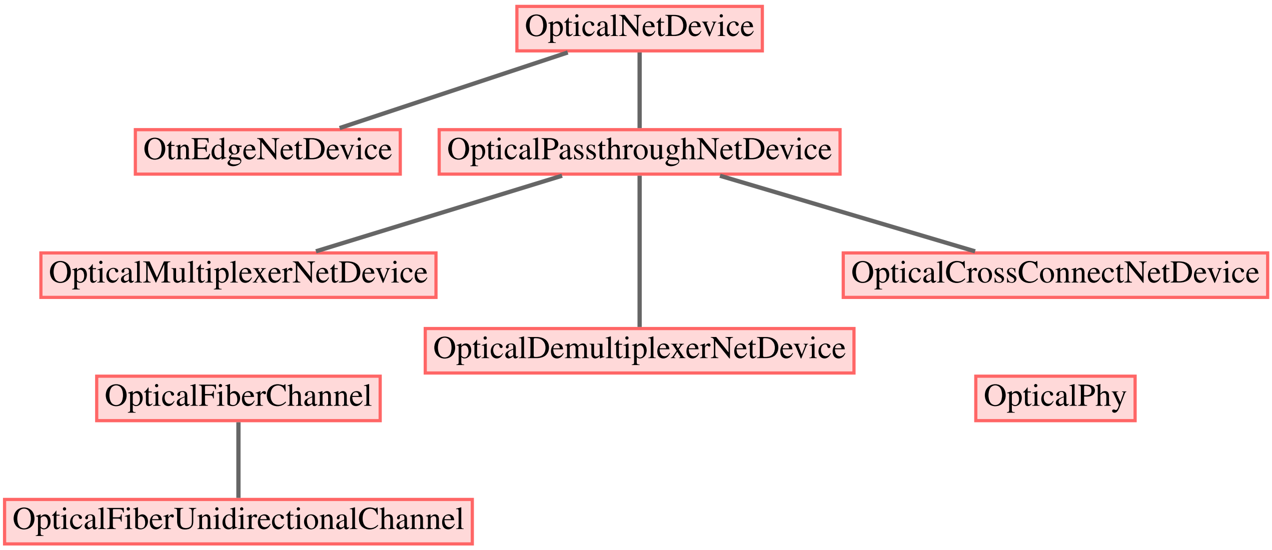


Figure 3.1: Class hierarchy.

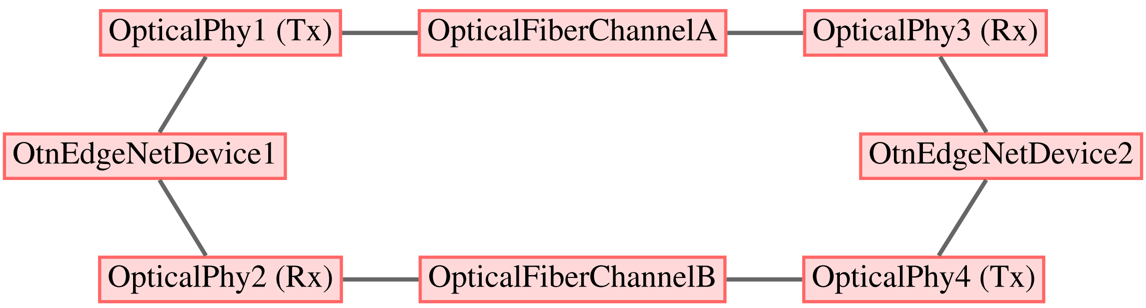


Figure 3.2: Relation between device, physical interface and channel [6].

which use lasers and photodetectors. We solved this by introducing an abstract class providing information about signal power gain. Models of lasers and photodetectors as well as antennas can derive from the said class.

`OpticalFiberChannel` class is used by `OpticalFiberUnidirectionalChannel` class that models fiber used for transmission in a single physical direction. It is expected to have `OpticalPhys` at both ends. Physical effects other than loss and delay (various types of wave scattering, chromatic dispersion, four wave mixing etc.) are not modeled at the moment. However, the spectrum model allows the implementation of such features should the research requirements eventually create a demand for it.

3.5.2 Modeling Failed and Working States of Components

Since our primary interest is in the field of reliability, we evaluated the the existing possibilities for modeling failures and repairs of network components.

ResumeNet[§] project, finished in August 2011, extended ns-3 simulator to analyze network availability [94, 95]. Implementation of reliability model presented in these papers works by stopping network interface at a node in case of node failure, and network interface at both ends in case of a link failure. This is a rather simple but quite unrealistic solution. Big issue here is that model specifically uses IPv4 and can not be used in case of IPv6 nodes or do not use IP at all for sending and receiving. Other frameworks developed in this project (e.g. topology and failure specification) might be usable in the future with some adaptations.

We opted to implement failure state directly in ns-3's base object class `Object`. Since mentioned classes `OpticalNetDevice`, `OpticalFiberChannel` and `OpticalPhy` derive from this class, this allows them to get information about current state of the object (failed or working). By default, this does not alter anything on existing classes in ns-3 simulator and researcher creating or extending a model has to explicitly use this feature.

Building upon this, `FailureRepairModel` is attached to `Object` and can change its state in accordance to elapsed time. It does so by calculating next event (be it failure or repair) and scheduling it to happen a certain time interval after the current simulation time. Times to failure and repair of objects are calculated during simulation run time according to user configurable probability distributions.

[§]Resilience and Survivability for Future Networking (ResumeNet) is a collaboration between The University of Kansas (KU), Lancaster University, ETH Zürich, Techniche Universität München (TUM), Techniche Universiteit Delft, Université de Liège (ULg), Universität Passau, Uppsala Universitet (UU), NEC Laboratories Heidelberg and France Telecom – Orange Labs. ResumeNet researches framework, mechanisms and experimental evaluation of network resilience and survivability in presence of failures for future networks and is funded by EU Future Internet Research & Experimentation (FIRE) from Seventh Framework Programme (FP7).

3.6 Example Case Study

The simulation presented here is based on the following scenario: the network consists of four OXCs which are modeled as `OpticalCrossConnectNetDevices` at nodes interconnected by pairs of fibers modeled as pairs of `OpticalFiberUnidirectionalChannels`. Each OXC is assumed to be transparent meaning that it does not read packet headers; it demultiplexes them based on the input wavelength, and switches them to the appropriate output link and wavelength, based on preconfigured information. OXCs are assumed not to possess any kind of wavelength conversion capabilities. Traffic generators (ns-3 on-off applications) and packet sinks are attached to the edge nodes.

The simulation can be configured by varying device and physical interface attributes such as usage of coarse vs dense WDM and number of inputs and outputs.

3.6.1 Case Study Setup

Consider an eight node network with physical topology as shown in the Figure 3.3. Dense WDM with 100 GHz channel spacing is used and each channel has bandwidth of 10 Gbit/s. All channels have delay set to 10 ms. In this scenario, main and backup lightpaths are statically defined (in addition to this approach, lightpaths can also be computed using any of the frequently used heuristic techniques):

- In case of no failure or in case of failure of link $n3-n4$, for communication between node $n5$ and node $n8$ path over nodes $n1$, $n2$ and $n4$ is used, and for communication between node $n6$ and node $n7$ path over nodes $n2$, $n1$ and $n3$ is used,
- In case of failure of link $n1-n2$, for communication between node $n5$ and node $n8$ backup path over nodes $n1$, $n3$ and $n4$ is used, and for communication between nodes $n6$ and $n7$ path over nodes $n2$, $n4$ and $n3$ is used.
- In case of failure of link $n1-n3$, for communication between nodes $n5$ and $n8$ main path is used, for communication between $n6$ and $n7$ backup path is used.
- In case of failure of link $n2-n4$, for communication between nodes $n5$ and $n8$ backup path is used, for communication between $n6$ and $n7$ main path is used.

We run the simulation for mean time to failure values 30 days, 60 days, 90 days, 120 days, and 180 days (all exponentially distributed) for $n1-n2$, $n2-n4$, $n1-n3$ and $n3-n4$. Mean time to repair is set to 8 hours (constant). Links $n1-n5$, $n2-n6$, $n3-n7$, and $n4-n8$, as well as all nodes are assumed to be completely reliable, i.e. assumed to be unable to fail.

On-off application at nodes $n5$ and $n6$ and packet sinks at nodes $n7$ and $n8$ are used for simulating traffic flows. On-off application at $n5$ is sending packets to $n8$ at using signal frequency 190 100 GHz (1577.03 nm wavelength), and on-off application at $n6$ is sending packets to $n7$ using signal frequency 190 200 GHz (1576.20 nm wavelength), so data transmissions occur in

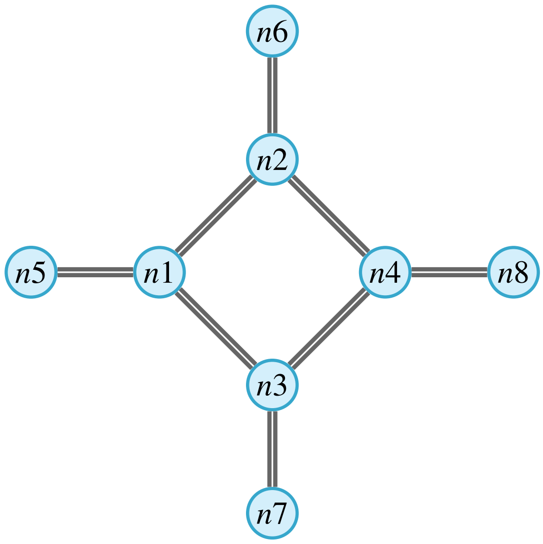


Figure 3.3: Topology for the case study.

parallel. Both applications send 1 Kbit/s of data in packets of 1400 bytes during "on" time which lasts 20 seconds (exponentially distributed), and then have "off" time which lasts between 1 and 3 seconds (uniformly distributed). On and off times alternate. We observed that while larger data rates (100 Mbit/s, 1 Gbit/s, 10 Gbit/s) increase simulation duration in terms of wall clock time approximately by a factor of 10^5, 10^6, and 10^7 (respectively), they do not significantly alter the percentage of packets lost due to channel failures so we opted for smaller data rate.

We measure packet loss due to link failures. Even though there are backup routes, packet loss still occurs due to one of the following factors:

- Packets in transit when the failure occurs get dropped on optical cross connect adjacent to the failed link, and
- Failures of two or more links in the same period of time (usually called dual and multiple failures respectively) which cause that no working backup path exists.

Exactly 1 year of time is simulated. Simulation results are shown in Table 3.1. Please note that the number of packets sent remains the same since on-off application configuration does not change between simulation runs.

3.6.2 Case Study Results Discussion

While this case study is very simple and done only to demonstrate the basic functionality of the model implemented in ns-3, results still deserve some discussion. We can observe that mean time to failure is correlated with percentage of lost traffic, and that correlation is linear. As mean time to failures increases, percentage of successfully transmitted traffic also increases. Therefore, percentage of lost packets decreases.

Mean time to failure for links	**Number of packets sent**	**Number of packets received**	**Number of packets lost**	**Percentage of packets lost**
30 days	5122922	5010003	112919	2.2%
60 days	5122922	5069166	53756	1.05%
90 days	5122922	5078451	44471	0.87%
120 days	5122922	5087854	35068	0.68%
180 days	5122922	5094886	28036	0.55%

Table 3.1: Simulation results.

3.7 Chapter Conclusions

In this chapter we presented prototype WDM network simulator based on ns-3 network simulation framework. We described why ns-3 was selected as a foundation among other network simulators, we outlined requirements for optical WDM network model, and we described the model architecture. Finally, we demonstrated the functionality doing simulation of a simple eight node network.

In comparison to OWns for ns-2, our model offers possibility of simulating failure and repair of optical links and components. Other tool we mentioned, EPON for OMNET++, has different goals compared to PWNS: it models access network, while PWNS models core network.

In ns-3's Google Summer of Code 2012 one of the projects proposed to be mentored by us was developing WDM components for ns-3. Due to a large amount of high-quality applications in other areas (Internet protocol stack, wireless networks, simulation infrastructure etc.), our proposal unfortunately was not selected for funding. Regardless, we decided to do the implementation work ourselves.

In the future we plan to validate existing models by comparing them to real world measurements and already published simulation results. In addition to this, in the following chapters we describe how we model a wider set of devices. However, additional devices not covered here, such as reconfigurable add-drop multiplexers and various architectures for multigranular optical cross-connects, are also a potential direction for future development.

While we implement basic signal properties such as frequency and duration, more complex physical properties of optical waves (chromatic dispersion, scattering, four wave mixing etc.) are not modeled. Also, other types of optical networks in addition to core (such as passive optical networks in access domain) would be an interesting extension.

On the side of reliability, failure and repair model mentioned here is described in more details along with applications in Chapters 5 and 6.

Chapter 4

Resilience of Telecommunication Networks

Presently used optical telecommunication networks suffer from frequent failures of network equipment, and some of those failures have serious consequences in terms of resulting service quality. Fibers are placed in cables, and cables are subject to damage which result in fiber cuts. Main cause of fiber failures is construction work, but fiber cuts can also be a consequence of natural disasters or human errors [3, 56]. Other network equipment is also subject to failures; among others, those are switching node failures, transmitter, receiver, and amplifier failures. These failures occur for the same reasons as fiber failures, and are additionally a result of hardware and software bugs.

With the introduction of wavelength-division multiplexing (WDM) in commercial metro and core networks the amount of data that can be transferred over the network per time unit multiplied. This growth created the situation in which failure of a single cable causes cuts of many fibers and many channels in each fiber. Therefore, it is necessary to clear up network reliability issues in advance, that is before the network is deployed in production.

In the rest of the chapter we first discuss failure recovery, classify and describe recovery schemes. Then we turn our attention to various concepts relating to network resilience. We discuss network challenges and path disjointedness. Finally, we discuss network performance measures used when evaluating network resilience, in particular reliability and availability.

4.1 Failure Recovery in Optical Telecommunication Networks

Fault management of WDM networks is a requirement for deployment; networks must have the capabilities to detect a failure, isolate it, and recover from it. Much research on the topic of fault management architectures has been done in the last decades [96, 97, 98, 99], and a lot of it is still ongoing. Many of the research findings helped in standardization of techniques related to failure detection; for example, in case of MPLS data plane failures, the detection has been described in RFC 4379 [100]. Many papers have been published on design of reliable architectures, failure detection, prevention and recovery, conformance testing and verification, and fault-tolerance [99, 101, 102, 103, 104]. Among these, we will focus on failure recovery.

Failure recovery is defined as the process of reestablishing traffic continuity in the event of a failure condition affecting that traffic by rerouting the signals on diverse facilities (nodes and links) after the failure [56]. The meaning of this term encompasses both protection and restoration, which we will define later in this chapter.

Failure recovery is obviously a critical feature of today's telecommunication networks. Users, be they individuals or institutions, rely on communication networks for everyday services. Institutions such as universities, corporations, government agencies, stock market companies, hospitals, and schools require their Internet access to just work all the time, since numerous and/or long lasting periods of service disruption could have severe consequences. For a more concrete example, consider a university laboratory which relies on the Internet to access and store experiment data, research papers, and books. Should a network failure disrupt the service, laboratory staff will be unable to carry on their daily duties. We can think of more extreme example where an emergency hospital call center loses the ability to receive calls due to a cable cut, which could potentially have devastating consequences.

A network fault that remains unresolved for a long period of time often causes losses for the service provider and its clients. Due to a long outage, the service provider loses revenue from the clients. Beside revenue loss, there is also a loss of credibility and good reputation. Service level agreement (SLA), established between service provider and its client, defines the acceptable levels of service outage and penalties. Penalties are paid by the service provider if the quality of service fails to meet the requirements agreed upon in the SLA [105, 106, 107]. Through the SLA the customer is guaranteed service availability, typically in the order of 99.999% (approximately 5 minutes of service outage per year) but it can vary. To illustrate the requirement for availability, consider a customer who is managing servers forming a content delivery network* (CDN) and who leases network resources from service provider. This customer requires the network to be available to sync the content between servers in the CDN; in case the service provider is unable to offer the acceptable service availability, the customer's servers will often be out of sync with each other, which is detrimental to the CDN service it provides to its clients. Thus, the trend for network providers is to provide the networks that are virtually uninterruptible, that is, appear to be continuously up from user's perspective [56].

4.1.1 Classifications of Failure Recovery Schemes

We say that a network is survivable if it is capable of failure recover in the event of a failure occurrence [56]. The degree of network survivability is induced from the ability to survive

*Content delivery network is a distributed system composed of servers in different geographical locations, used to serve content to users with high performance and availability.

single or multiple link or node failures, and is a consequence of network design choices. From now on, we will focus on how network reacts in case of failure; network design is a separate topic which is covered in [3, 108, 109, 110, 111].

The underlying assumption before any recovery techniques are employed is that network has sufficient amount of redundant capacity to withstand any single failure. Networks are rarely designed to withstand any combination of uncorrelated multiple failures due to additional redundant capacity requirements, which increases total cost. However, compared to single failures, uncorrelated multiple failures are very rare so this design choice is justified.

Various recovery methods have each their own advantages and disadvantages. For example, a method can be slow but require very little extra spare capacity. Alternatively, another method might perform significantly faster but also require a lot of extra spare capacity. Some of the metrics to evaluate and compare various methods are: speed of recovery, capacity efficiency, cost of implementation, and amount of signaling traffic. Customer requirements can vary significantly, and networks operators can bind different recovery techniques to different classes of customers. For example, quality levels can be [56]:

- guaranteed fast recovery service (50 ms recovery time) using dedicated 1+1 diverse routing, called dedicated backup path protection (DBPP),
- shared backup path protected (SBPP) service (order of 100 ms recovery time),
- services with multiple diverse paths,
- unprotected (non-preemptible) services,
- best effort (preemptible) services using the redundant capacity available in the network.

We now introduce definitions of basic concepts in survivability. While there are multiple ways to define the basic concepts actively used in the research community, definitions provided by [56] and [3] are more prevalent than others so we will follow them.

Protection is a recovery technique which precomputes spare path (backup path, alternate path used in case of working path failure) and its channels prior to failure occurrence. We will also consider techniques that precompute spare path to be protection techniques regardless of path preconfiguration (this distinction can vary in different literature).

Restoration is a recovery technique which does not precompute spare path and spare path channels prior to failure occurrence but instead calculates them in real time after a failure occurrence. Spare capacity and switching equipment combined with a rerouting scheme are used in case of failure. This technique, of course, requires switching equipment to be reconfigurable. Restoration techniques have the advantage of not requiring specific redundant resources to perform recovery, but they indeed depend on redundant capacity being available in existing carrier resources. Recovery is then provided through reconfiguration of routing around the failed links or nodes using the network state available in a centralized control plane or distributed among the individual switching nodes.

Ring-based protection in mesh networks consists of using precomputed cycles in mesh networks to reroute the signal around a link or node failure.

Link- or span-based protection (restoration) in mesh networks does rerouting of the failed connection only around the failed link. It is done at the end nodes of the failed link, therefore it is considered a local protection (restoration). More generally, when a single link failure affects more than one channel, it is possible that affected channels will recover from failure by using different routes between the two end-nodes of the failed link.

Channel protection is the case when a spare channel is used on the same link as the working channel. This technique is used for failure recovery in case of, for example, transmitter or receiver failures.

Path-based protection (restoration) in mesh networks consists of rerouting the failed connection end-to-end. Spare path from source to termination node is used in case of failure of working path, unlike link-based approaches where rerouting is done locally.

Dedicated backup path protection (DBPP) implies that redundant resources are dedicated solely for rerouting of a specific connection in case a failure hits the working path of the connection. In other works, spare paths of all connections are precomputed in advance and no resources are shared between them.

Shared backup path protection (SBPP) implies that redundant resources are not dedicated for rerouting of a particular connection, but are instead shared among a number of different working paths prior to failure occurrence. After the failure occurrence hits one of the working paths, that particular working path takes up previously shared redundant capacity and uses it for its spare path. Until and unless the shared spare capacity gets released, other paths will remain unprotected.

Restoration schemes use complex computations after the failure occurs and they are clearly slower than protection methods that use precomputed backup paths and channels. Another factor in favor of protection methods are simple control protocols that allow service recovery that is transparent to the user due to very low delay, compared to much larger delay induced by restoration algorithms and protocols for path-based methods. In the following we will mainly focus on protection methods.

Hierarchically, survivability techniques are classified as follows:

- protection
  - dedicated
    - link
    - path
    - channel
    - segment
  - shared

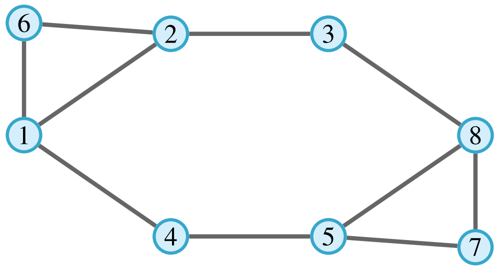


Figure 4.1: Example network used to illustrate the differences between link- and path-based schemes.

        * link
        * path
        * channel
        * segment
- restoration
    - link
    - path

To illustrates the difference between link and path-based schemes, consider the network given in Figure 4.1 and path between nodes 1 and 8 going over nodes 2 and 3. If the link from node 1 to node 2 fails, link-based scheme will result in path 1-6-2-3-8 being used. On the other hand, a path-based scheme will result in path 1-4-5-8 being used until the repair happens.

Many studies have been done on comparing the efficiency of both schemes [3, 112, 113, 114], and they have shown that path-based schemes require up to 19% less redundant capacity to be present in the network compared to link-based schemes.

4.1.2 Automatic Protection Switching in Point-to-Point Systems

Synchronous Optical Networks (SONET) and synchronous digital hierarchy (SDH) define three types of Automatic Protection Switching (APS) architectures: one-plus-one (1+1), one-for-one (1:1) and M-for-N (M:N) [61, 115]. Some literature also considers 1:N as a separate case from M:N; we will consider it as a special case where M is equal to 1.

In 1+1 protection architecture a spare path exists for every working path and the system establishes a diverse route from end to end. The traffic the network has to send is transmitted and received using both paths. One of the received signals is selected at the receiver end. In case of a fiber transmitting one of the signals gets cut, the receiver switches to another signal immediately without exchanging any additional control messages. However, the SDH APS signaling bytes (K1 and K2) are used to indicate the switch has been done. We should also note that 1+1 architecture works in non-revertive mode. This mode of operation implies that when the repair of the original path is done, there is no automatic switching back. The two paths are considered equivalent. Since both paths are used at the same time, 1+1 requires 100% extra capacity in the network and no low priority or best-effort traffic can be transmitted over the extra capacity while it is unused.

In 1:1 protection architecture the traffic is transmitted and received only using one of the paths, working or spare. This is the main difference to 1+1 architecture. Initially, both the transmit and receive ends switch to using the working path. In case of a failure occurrence is detected by the receive end, both ends switch to using the spare path to recover from the failure. Since only one of the paths is used at the time, the other path can be used for low priority (preemptible) traffic. That is, if both paths are failure-free, working path will be used for the high priority traffic and spare path will be unused. Therefore, preemptible traffic can be transmitted over the spare path.

Upon failure of the working path, the high priority traffic will switch to spare path and the preemptible traffic will be lost. However, unlike 1+1, the 1:1 protection architecture operates in revertive mode. This means when the failure of the working path is repaired, the transmitter and receiver will switch to using it again and spare path becomes available for failure recovery or low-priority traffic. Already mentioned SDH K1 and K2 signaling bytes are used for APS signaling in 1:1 protection. These bytes carry the message concerning the failure and are used to trigger and coordinate the process of recovery.

In M:N protection architecture the ideas of 1:1 protection are generalized. The protection resources are shared among working paths in a way that M spare paths protects N working paths. Since $M < N$, in case of multiple time-overlapping failures, there will be not enough spare paths to protect all the working paths. Therefore, the working paths carrying traffic with the highest priority will have their traffic switched to spare paths, and the rest of the traffic will be lost. Like 1:1, this architecture works in revertive mode and can carry best-effort or low priority traffic when spare paths are unused.

Aside from APS in point-to-point networks, ring architectures are also used for protection both in SDH and WDM networks [14, 116]. Examples of such ring architectures include self-healing rings used in SDH [61, 117] and also ring covers, cycle double covers, and p-cycles used in mesh networks [3, 118, 119].

4.1.3 Dedicated Backup Path Protection in Mesh Networks

When using DBPP, traffic for each of the connections is sent from the source node to the destination node using two disjoint paths, the working and spare path. In case of a failure occurrence no signaling is required since traffic is bridged at the source node to both working and spare path. Detection of the failure occurrence at the receiving node triggers a protection switch from the working to the spare path in a same way as 1+1 APS in SDH or WDM networks. DBPP is therefore the simplest mechanism of path-based protection available in mesh networks.

The two paths (working and spare) can be link-disjoint, node-disjoint, shared risk link group-disjoint or any combination of these [120, 121, 122]. If the requirement is for the network to be able to continue functioning in presence of link failures, then link-disjoint paths are the option. If there is additional requirement that spare paths avoid failing along with working paths in case of node failures, then link and node-disjoint paths will be used. Many algorithms have been developed for finding link and node-disjoint paths [123, 124, 125].

Finally, in presence of shared risk link groups (SRLGs), which introduce correlation between failures, one can have two logically distinct links fail at the same time. If there is a requirement to avoid such a case, shared risk link group-disjoint paths are to be used. We will discuss correlated failures and SRLGs in more detail in Chapter 5.

4.1.4 Shared Backup Path Protection in Mesh Networks

SBPP scheme precomputes disjoint working and spare paths for each connection in a same way as DBPP. In DBPP spare path of each connection gets dedicated redundant capacity. However, in SBPP, redundant capacity is shared among multiple spare paths which can not all be used at the same time [126, 127, 128, 129]. This also implies signal is not bridged on backup path as in case of DBPP. In effect, the spare capacity is soft-reserved, and node switching elements are only configured to use it in case of failure, remaining otherwise unused. Sharing redundant network capacity lowers spare capacity requirements on the network.

Sharing redundant capacity between spare paths of two connections works well when the two connections have disjoint working paths [56], as this implies that a single failure will not cause both connections to fail at the same time. Connections failing at the same time would imply both of them have to use the shared spare capacity, which is not possible. More generally, the level of disjointedness in terms of link, node, and SRLG along with recovery requirements (recovery from link, node, and SRLG failures) determines how much redundant capacity can be shared in the network.

In terms of required capacity, SBPP is more efficient than DBPP. However, SBPP requires signaling messages to be exchanged after a failure occurrence to configure the switching elements to use the previously soft-reserved capacity. This requirement introduces delays due to signaling and configuration, so SBPP is slower than DBPP.

Just like M:N APS in SDH, SBPP also operates in revertive mode. Since used spare capacity is released as soon as the repair of the working path happens, the traffic disruption in case of failure of another working is minimal.

4.1.5 Link or Span-Based Protection

When a failure occurs and it is detected, link- or span-based protection is utilizing optical cross-connects (OXCs) to attempt a rerouting of the lightpaths through alternate circuits around the failed links or nodes. If failure affects more than one working path passing through the same link or span, the rerouting can happen over the different circuits [56].

Finally, in some architectures link and path-based recovery schemes are combined. For example, if link-based recovery fails to recover from failure (i.e. no spare channel is available between OXCs), end-to-end path-based protection is triggered to set up a spare path. More generally, multi-layer recovery approaches are used in IP-over-WDM networks; if lower layer fails to recover from a failure, higher layer is triggered. Two layers attempting to recover without coordination can result in race condition and end in failure to recover. Therefore, multi-layer recovery has to be coordinated [93, 130, 131, 132].

4.2 Basics of Network Reliability

Resilience is defined as ability of the network in providing and maintaining an acceptable level of service in spite various faults and challenges ([133] and references therein). ResiliNets architectural framework [134] offers a set of fundamental principles and strategies for mitigation of impact of network failures. ResiliNets framework divides resilience disciplines into two categories: challenge tolerance and trustworthiness.

Challenge tolerance recognizes detrimental events or conditions that result in operationally degraded communication network. On the other hand, trustworthiness considers measurable characteristics such as dependability (a term including both reliability and availability), security and performability. Obviously, terms challenge tolerance and trustworthiness are related. The relation is two-fold; robustness, which is defined as the ability and measure of networks to remain trust worthy in face of challenges, and complexity, which arises from mechanisms that improve resilience and must be managed. Namely, increase in complexity due to implementation of additional mechanisms for resilience improvement can in fact result in decreased resilience.

Challenge tolerance is further divided into survivability, disruption tolerance, and traffic tolerance. Survivability includes fault tolerance, which tolerates only a few random failures, and also adds tolerance to many correlated or targeted failures [133, 135, 136]. Traffic tolerance studies challenges against normal traffic (such as distributed denial of service attacks [137]) and also unusual legitimate traffic, one example being flash crowds [138]. Finally, disruption tolerance deals with challenges in mobile wireless communication, which is outside the scope of this thesis.

Challenge tolerance of networks can be increased using ResiliNets strategy named $D^2R^2 + DR$ that consists of real-time mechanisms and long-term mechanisms. Real-time mechanisms (D^2R^2) are defense, detection, remediation, and recovery. Long-term mechanisms (DR) are diagnosis and refinement. The real-time mechanisms in the ResiliNets strategy have the purpose of bringing the network service level to acceptable level upon failure. On the other hand, long-term steps are done to improve service level with the evolution of the network over time. We now describe each of the mechanisms in details.

Defence mechanism are the initial step for ensuring the resilience of network in real-time. They can be passive or active. Passive mechanisms mainly consist of structural improvement of the network. Two of them are: including redundant spare components in the network in order to attain tolerance to failures and increasing physical link diversity in terms of geographical locations to avoid multiple correlated failures [134, 139]. Active mechanisms are applied at run time; one example would be a firewall that filters unwanted network traffic. Detection is necessary to recognize penetration of defensive measures.

After detection of penetration and resulting condition, the effects of detrimental event or condition should be remediated. Remediation should be done in a way that provides the highest level of network service possible under the circumstances and with the resources that remain available. For example, in case of a cable cut, the spare cables can provide limited capacity so end-to-end communication continues to function. Recovery implies restoring the network operation to the original and normal state [134, 140].

Diagnosis, one of the two long-term mechanisms, covers fault localisation and root cause analysis. Root cause analysis implies finding out what the meaning of failure alarms [141]. When the faults are identified using root cause analysis, the refinement of the network can happen. Refinement improves defence, detection, remediation, and recovery (D^2R^2) for given and predicted future network challenges.

4.3 Classification of Network Challenges

In this section we describe and categorize network challenges into very broad groups. We continue to follow the taxonomy of ResiliNets architectural framework [134].

The following seven categories of challenges are defined:

1. **Geographically correlated failures due to large-scale disasters**. Communication network components can be affected by large-scale natural disasters such as earthquakes, volcano eruptions, hurricanes, and tsunamis. Service failures that can be observed in such cases are geographically correlated, due to areal impact of disasters. Finally, large-scale disasters do not have to be natural; humans can also be the cause of disasters in case of power blackouts, electromagnetic pulse weapons etc.
2. **Sociopolitical and economic challenges**. Deliberate human activity through social, political and economic challenges can also be a threat to resilient communication. For example, we can consider network outages due to political decisions, terrorist attacks, and legal battles between Internet service providers that force one of them to stop using its

infrastructure with the aim of increasing market share of the other.

3. **Dependent failures**. Each network layer considers layer below it (if any) as infrastructure and layer above it (if any) as a service. Should the layer below fail, layers above it are faced with a challenge to keep the service unaffected. For example, if a logical link provide to IP layer by the optical layer gets torn down, IP will be faced with the challenge of rerouting traffic over other logical links provided by optical layer. Furthermore, if IP routing is then unable to find new paths, the transport layer will see it as a challenge to enabling end-to-end communication. Finally, dependencies in communication infrastructure result in failure of services using the infrastructure.
4. **Human errors**. Human action can also cause network failures in a non-malicious way. For example, misconfiguration of network components such as IP routing daemon becomes a challenge to transport layer working on top of it. Additionally, catastrophic failures can be a result of insufficiently educated personnel working in network operations, planning or design stages.
5. **Malicious attacks**. Deliberate endeavors to disrupt network operation, example being targeted attacks on network hardware and software, are challenges to communication network. Damage can become much larger if the attack targets Internet routing and control protocols due to possibility of global impact.
6. **Unusual traffic**. Legitimate unusual traffic, example being already mentioned flash crowds on the Internet, is a challenge to communication network. This kind of challenge varies depending upon the characteristics of the specific network. For example, an unpopular website that gets covered on Slashdot may go down due to order of magnitude or even more increase in traffic it has to handle [142]. Such a website might have been designed for its present user base, but not for order of magnitude or two more.
7. **Environmental challenges**. Real world is the environment where communication occurs. Therefore, communication is unavoidably challenged by real world phenomena such as mobility impairments of communication nodes in case of a mobile network, object that become signal barriers and result in weakly connected wireless channels, and potentially high variance in communication delays.

4.4 Path Disjointedness

Network design covers construction of the network from square one. Many decisions are to be made, including selection of node positions during node placement and deciding on what links will be established between nodes and provide connectivity required by network services. Network design differs depending on network type (backbone, metro, or access) due to differences in topological structure of various types. Researchers have been studying network design in the past decades and concluded it falls in the category of NP-hard problems [134].

Network optimization comes after network design and can be done in multiple ways. One approach is to fix the number of edges and rewire nodes. Another approach is adding new links to existing network with the goal of improving graph connectivity. Network optimization objectives are cost, capacity, reliability, and performance [143]. Cost increases with increasing number of nodes or links and also with increasing their capacity. Graph metrics such as betwenness, closeness, average degree, and graph diversity [144, 145] can be used as a measure of graph connectivity. We will consider path disjointedness metrics here; algebraic connectivity [146] can also be used as a basis for optimization [134, 147], but is outside the scope of this work.

4.5 Performance Evaluation of Network Resilience

Performance evaluation of computer networks inherently interdisciplinary field of study, as it relies on knowledge from various disciplines such as telecommunications, computer science, physics and applied mathematics. We will narrow our discussion of performance evaluation of network resilience to evaluation using analytical models and simulation models. Aside from using models, network resilience can also be studied via real world experimentation testbeds, examples of which are GENI [148] and FIRE [149].

4.5.1 Analytical Models

Reliability is defined as the probability of being in the working state for the entire duration of a specified time interval under defined environmental conditions [140, 150, 151]. Analytical computation of reliability uses certain metrics which we will define now. Mean time to failure (MTTF) is the average (mean) time that a component is operational before a failure occurs. Mean time to repair (MTTR) is the average (mean) time that it takes to repair a failed component. Finally, mean time between failures (MTBF) is the average (mean) time between two failures of a component [140, 152]. The relation between MTTF, MTTR and MTBF is

$$MTBF = MTTF + MTTR.$$

Failure rate λ is expressed in failures in time (FIT). The relation between MTTF and λ is [3, 133, 140]

$$MTTF = \frac{1}{\lambda}.$$

Repair rate μ is related to MTTR as

$$MTTR = \frac{1}{\mu}.$$

For constant λ, reliability R is a function of time t and given with a formula [3, 140]

$$R(t) = e^{-\lambda t}.$$

Unreliability Q is the complement of reliability. For constant λ unreliability is defined as

$$Q(t) = 1 - R(t) = 1 - e^{-\lambda t}.$$

Availability is the probability of a component being in the working state at a random time point. If we measure component working and failed time over a particular time interval, we can define availability as

$$A = \frac{\text{working time}}{\text{working time} + \text{failed time}}.$$

Should the time interval become sufficiently large, the fraction on the left becomes equal to

$$A = \frac{MTTF}{MTTF + MTTR},$$

which the relation we will use to compute availability when given MTTF and MTTR. Unavailability is the complement of availability

$$U = 1 - A = \frac{MTTR}{MTTF + MTTR}.$$

Reliability and availability might at first look confusingly similar, but are indeed not the same metric. Unlike reliability, availability does not require the component to be in the working state during the entire time interval before the time point. In other words, availability is related to instant of time probability of failure-free operation, while reliability is related to failure-free operation over time interval. To give an example of both, we will consider the following two systems: domain name system (DNS) server and online gaming server.

DNS is based on query and response model; a client who desires to know the IP associated with domain name will send a query to server and server will send back a response containing the answer. In this case, the client is only concerned with DNS working (i.e. being able to respond to query) during the short period of time when the query is sent by the client. Thus, a designer will optimize the DNS server for availability, while the reliability of the server will not be as important.

Online gaming server is an example of the opposite requirement. A client who connects to online gaming server expects the server to be working over a time interval (e.g. a game session duration). Therefore, a designer will optimize the online gaming server for reliability. In general, for a given system one can choose to optimize for reliability or availability (or a certain combination of both), depending on the service requirements placed on the system.

Availability and reliability give quality of service assessment for system components and subsystems. However, they are inherently limited to describing the components and subsystems in terms of binary states [153, 154]. Multi-state systems in general, and degradable systems in particular, do not entirely fit this model. Specifically, degradable system can provide an acceptable level of service in presence of failures. In networks many components can fail and, depending on the particular network configuration, level of service can remain acceptable in spite of failures.

4.5.2 Network Availability

General systems might degrade or partially fail during operation. Such possibility requires use to make a clear distinction between system up and down states. To say a network is down when only one link fails in network with sufficient spare capacity to reroute all connections would be clearly wrong. On the other extreme, to say a network is up when at least one path is up does not provide a very useful insight into network availability as perceived by the user. Therefore, the term network availability is a non-specific term that encompasses various measures such as minimal or average path availability.

We will define and use the following two measures of network availability:

- s,t-availability, defined as the minimum of all connection (in case of optical networks, logical channel) availabilities,
- g-availability, defined as the probability that all connections (logical channels) in the network are in working state at a random time.

Since availability values are often very close to 1, it is easier to do comparisons of simulation results based on the order of magnitude difference in unavailability. Therefore, to ease the evaluation of the effect of various simulation scenario settings on network availability, we additionally define s,t-unavailability and g-unavailability as complements of s,t-availability and g-availability (respectively).

4.5.3 Simulation Models

Complex scenarios concerning network analysis can be tough fit for analytical approach. Simulations are an alternative approach that can works for complex scenarios where analytical models lack versatility. However, an approach based on simulation involves modeling network performance under challenges, which certainly is not a trivial task [133, 155, 156]. Therefore, simulation models are created in a way that simplifies as much as possible.

To illustrate simplification process, we assume one wants to use simulation to study network application performance in relation to network bandwidth and delay, and that applications use client-server model. One will then model application traffic in a very detailed way and also TCP/IP stack. However, since underlying network is considered only in terms of bandwidth and delay, one can use simple point-to-point links with those two attributes and avoid modeling data link layer and physical layer characteristics such as framing, inter-frame spacing, signal propagation and loss etc.

4.5.4 Models and Network Experiments

Both the analytical and simulation models can aid us in understanding the impact of network challenges on network performance and improving it. They are non-exclusive and can be used for mutual verification of results. Such verification helps to improve model quality by correcting modeling errors and increasing the level of detail. For example, say one implements a resilience model in a network simulator. While using simulation approach to evaluate network availability, one finds that the results obtained using the model deviate significantly from analytical results in cases where using both analytical and simulation approach is possible. Then one has to review the implementation, and possibly revise the model until one gets good fit.

As we briefly mentioned already, analytical and simulation approach are not the only two approaches for evaluating network resilience. Testbeds are used for real-world evaluation of network reliability [133, 157]. The cost of hardware devices required to set up a small-scale testbed can be very low. However, simulations are much cheaper than testbeds, perfectly reproducible, and easily debuggable [66, 158]. Therefore, simulations can be used to preselect designs and configurations for testbed-based experimentation; namely, one can start with dozens of possibilities, and use simulation to reduce it to just a few which show the best performance and will be further studied on real-world testbeds.

4.6 Chapter Conclusions

We described protection and restoration recovery schemes, and also link- and span-based schemes. We specifically discussed the advantages of path-based schemes, and studied DBPP and SBPP. Both approaches provide a link-disjoint spare path for each working path in the network. DBPP scheme is simpler than shared protection and offer better performance in case of multiple failures, but also require more spare bandwidth than SBPP scheme. Better performance of DBPP scheme is a result of dedicated spare path resources for each working path, which is not the case with SBPP scheme. The advantages and simplicity of DBPP are the reasons we will use in the following chapters.

Chapter 5

Correlated Failures of Network Links

5.1 Introduction and Motivation

Shared risk link group (SRLG) [159] is a group of links in a network that share a physical location. This can be a cable, a duct or an exit at a node. (Example of the last case is shown in Figure 5.1.) All links contained in the SRLG have a probability of being damaged in case of physical damage to one link in the part contained in the SRLG. Such physical damage introduces a situation where a multiple logical failures in the network occur due to a single physical failure, and is therefore in general more damaging than two uncorrelated time-overlapping single link failures.

SRLG is called coincident if links contained in it are incident to a common node. For example, SRLG shown in Figure 5.1 is a coincident SRLG, since links $2-3$ and $2-4$ have node 2 in common. If the links contained in SRLG are not incident to a common node, it is called non-coincident SRLG. SRLG of either type is called general SRLG.

AT&T, the largest American provider of fixed telephony services, claims from experience that a link in the network may belong to over 100 SRLGs, each corresponding to a separate fiber group. In addition, in a large network it is very hard to maintain accurate SRLG information [160]. Common approach is to avoid SRLGs in the path routing stage, namely by making working and spare paths SRLG-disjoint. If the network has enough capacity, this leads to complete restorability in case of failure [161]. However, such a restorability increases costs

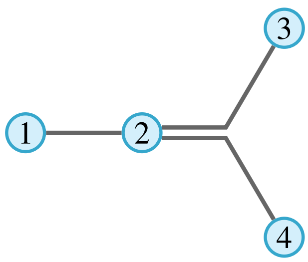


Figure 5.1: SRLG example with two cables sharing a common exit at a node.

and also is not always a necessity, as there is a possibility of dropping best-effort traffic in case of failure. Furthermore, common approaches rarely take SRLG weights (length, number and capacities of links) into account when doing route computation.

In this chapter we evaluate the effect of physical lengths of SRLGs on network availability. Since the failure correlation induced by SRLGs makes analytical computation of availability hard except in trivial cases, we use Monte Carlo method utilizing a new model implemented by network simulator ns-3 [70] to obtain results.

We expect that physically longer SRLGs will have higher failure probability, and more negatively impact logical channel and network availability, and we are interested in finding out how much. We would also like to compare impact on network availability of coincident and non-coincident SRLGs of the same length. Since SRLG elimination is in general costly, it is useful to know the impact of partial elimination (shortening the physical length of SRLG). While the exact results are hard to obtain analytically (except in non-trivial cases) due to the model complexity, it is possible to model an optical network containing SRLGs and use Monte Carlo simulation to get approximate results.

The chapter is organized as follows: in Section 5.2 we briefly cover related work, in Section 5.3 we present our model of optical transport network, in Section 5.4 we describe parts of the model which are related to availability analysis, in Section 5.5 we compare the results obtained by simulation to results using analytical methods, in Section 5.6 we present the case study. We analyze simulation precision in Section 5.7 and in Section 5.8 we conclude with some directions and plans for future work.

5.2 Related Work

While the protection and restoration of lightpaths in case of correlated component failures in optical networks has been studied by researchers for a long time, very few works are concerned with estimating the probability range and effects of such failures. One of the earliest papers explaining why correlated failures should be considered is [162]. Correlated failures are called dependent in that paper, which presents arguments against the assumption that all failures are independent. Lam and Li [163] study the correlation between link failures in communication network and propose an event-based reliability model. In the proposed model dependent (correlated) failures are the effect of independent (uncorrelated) events. Single failures of components occur with certain probabilities and cause failures of other components sharing the common equipment.

SRLG introduces correlation between link failures [159, 161, 164], since it is a set of links that share a common physical resource, such as conduit, cable or duct. Usual assumption is that correlation between failures is deterministic, implying that failure of one link in a SRLG always causes failure of all other links contained in it. In reality, this is not necessarily the case, so probabilistic models in which the links contained in the SRLG are damaged with a certain probability in case of failure have been studied in [165, 166]. Lapčević et.al. studied the impact of correlation between failures (including SRLGs) on network availability, and concluded that it is significant [165].

Various approaches to path provisioning, and more specifically routing and wavelength assignment (RWA) in optical networks containing SRLGs have been studied [167, 168, 169, 170] with the common goal of avoiding the failure of working and spare path at the same time. RWA problem can be expressed as an integer linear program. Since it is in general NP-hard, heuristics are often used [115, 171]. Lee and Mondiano [166] developed more general probabilistic SRLG framework for studying correlated failures, and formulated a problem of finding paths with minimum failure correlation as a non-integer linear program. An approach utilizing colored graphs (graphs containing colored vetrices or edges) for modeling shared risk resource groups, an unifying concept for both SRLGs and shared risk node groups has been proposed in [172, 173, 174]. Multicast routing utilizing path protection in presence of SRLGs has also been studied [175].

Somewhat related to our work are the papers studying geographically correlated failures [176, 177], which do not address SRLGs specifically, but focus on a broader set of link failures due to accidents and attacks affecting regions. A comprehensive network reliability framework is proposed in [140] and its implementation in network simulator is described in [95]. The framework includes SRLGs as a special case, but does not consider their effects specifically.

Our work expands on prior research by also taking into account the length of SRLGs in the network and comparing coincident and non-coincident SRLGs in terms of the effect on network availability.

5.3 Optical Network Modeling

In our previous work [6], we analyzed existing simulation models for optical transport network and found that none of the existing implementations fit the requirements, so we developed our own model basing on the infrastructure provided by network simulator ns-3 [70]. We named our software PWNS (acronym for Photonic/Prototype WDM Network Simulator, name picked in the beginning of development). Taking into account the feature functionality of ns-3 network simulator at the time, we had to identify the specific areas where it was to be extended to support simulating optical WDM network. We considered the differences between the networks that have existing models in ns-3, which operate almost entirely in the electronic domain, and the optical WDM networks, which operate in both the optical and electronic domain. We opted for model based on components since it is easier to develop, test, verify and validate, and also because the implementation of feature functionality can happen iteratively, first implementing a feature and then testing the implementation. In addition, code re-usability inherent in object-oriented design reduces the time needed to develop a similar model.

Components of ns-3 are modules, which consist of one or more C++ classes which together make one or more models of real world communication devices, communication channels, network protocols etc. Abstract base classes used implemented by every model of a physical network in ns-3 are `NetDevice` and `Channel`. `NetDevice` describes a network interface card at a network node; `Channel` interconnects two or more network cards and contains delays, losses etc. Models of complex networks (e.g. WiFi, WiMAX, LTE) frequently separate PHY layer of the network card from its MAC layer to allow combining various MAC devices and PHY devices and facilitate code reuse.

Building upon the work done in described in Chapter 3 Section 3.5 and [6], we added support for availability analysis [7] which is described in Section 5.4. The entire model code has since undergone a significant refactoring to support evolving requirements of our research, and also to make usage and further development easier. We describe the current version in the text that follows.

Our optical network model implementation consists of models for common optical transport network components: edge network devices (class `WdmEdgeNetDevice`), core network devices (classes `WdmOxcNetDevice`, `WdmMuxNetDevice` and `WdmDemuxNetDevice`), physical interfaces (classes `WdmInputPhy` and `WdmOutputPhy`) and channels (class `WdmUnidirectionalChannel`). Detailed description of these classes can be found in [6].

In addition to models for optical network components, our implementation also supports analysis of availability. For the present requirements of our research, we implemented models for additional physical objects and software entities.

5.3.1 Control Plane, Lightpaths and Logical channels

`LightPath` is a class modeling a lightpath passing through one or more network devices, physical interfaces and fibers. In case any of these becomes faulty, `LightPath` instance is notified about it and it changes its state from working to failed. Upon repair of a previously failed network device, physical interface or fiber, it is also notified and changes its state back to working.

`OtnLogicalChannel` is a class modeling a logical channel that has working and spare lightpath. In case of failure of working path, spare path is activated, and channel remains in working state. In case of failure of both paths, channel changes its state to failed. Upon repair of at least one path, logical channel switches to it and changes its state back to working.

`OtnCentralControlPlane` is a class that manages logical channels present in the network, establishing and tearing down channels on demand. We opted for centralized control paradigm instead of distributed, due to the centralized nature of a network simulation. The model uses

DPP scheme, and uses Dijkstra shortest path algorithm to find both working path and spare path when creating logical channels.

5.3.2 Helper Classes

`OtnHelper` contains helper functions that ease installation and interconnection of optical network devices, channels, cables and shared risk link groups, and also set up the control plane for the network.

5.4 Optical Availability Analysis

Network availability A is a probability that a repairable system will be in operating state at a random moment in time. Service providers and network operators frequently aim for "five nines" (99.999%) availability, which translates to less than 5.26 minutes of network outage per year.

Network availability can be computed by analytical and simulation methods. Analytical method uses mean time to failure (MTTF) and mean time to repair (MTTR) of a component (for example, a link or a device at a node) to compute the component availability. Network availability computation takes into account series or parallel relationship between the components in the network. However, the relationship between components can be complex in presence of failure dependencies. Failure dependencies are neither serial nor parallel relationship, which makes analytical computation of availability hard. For example, in case a SRLG contains two cables, there is a probability 0.7 that a failure occurring in part of a cable contained in SRLG will affect both cables, meaning that, on average, 70% of failures will affect both cables, and 30% of failures only one cable [165, 178].

Monte Carlo simulation can be used for network availability estimation. To do so, network model implemented in a discrete event simulator has to support handling failure and repair events. Then, Monte Carlo simulation uses random numbers to generate times to failure and times to repair for components in the network, based on their MTTF and MTTR values. Failure and repair events are handled by the model, and it is possible to make failure (or a repair) of a particular component affect other components in some way. As a consequence, this makes modeling of complex relationships such as failure dependency possible.

Unavailability is a complement of availability. Since availability values are often very close to 1 (or 100%), it is easier to do comparisons of availability results for varying network parameters based on the order of magnitude difference in unavailability.

5.4.1 Failure-repair Model Description

All the classes mentioned in Section 5.3 are derived from ns-3's base object class named `Object`. We wanted to avoid adding failed and working state functionality to this base class, since it would likely be unused in most of the other classes which are not interested in modeling object state. Instead, we opted to implement failed and working state of an object by using multiple inheritance. Multiple inheritance is used with care to avoid the diamond problem, specifically, the class providing object failed and working state functionality (named `StartStopFunctionality`) does not derive from the `Object` class*.

The classes modeling objects that understand the notion of failed and working state derive from both their base class, that is subclass of `Object` and class providing failed and working state functionality. Therefore, this approach does not alter anything on existing classes in ns-3 simulator and researcher creating or extending a model has to explicitly use failed and working state functionality.

Building upon this functionality, `FailureRepairModel` class is provided. This class is expected to be aggregated to an object which allows state changes from failed to working and from working to failed. In order to achieve this aggregation in the most generic way, a functionality provided by ns-3 named bound callback† is used [70]. When aggregated to an object, failure-repair model does changing of object state either a certain number of times or for certain amount of time specified by the user, whichever comes first. Duration of failed and working state is decided by random variables specified by the user.

5.4.2 Physical Entities

`OpticalFiberCable` is a class modeling a physical cable containing one or more fibers. It has a physical length, measured in meters or kilometers, and it contains information about positions of zero or more unrepaired cable failures. It is possible to configure the probability that a cable failure will affect any of the fibers in the cable.

`SharedRiskLinkGroup` is a class modeling a group containing parts of two or more cables that share a physical location. If the part of the cable that gets hit by a failure is contained in the SRLG, other cables in the same group will also fail at a certain probability. Our model allow configuration of this probability, ranging from 0 to 1. In case it is 0, the cables failures will occur uncorrelated to each other, while in case it is 1, each failure affecting SRLG will cause failures of all cables contained in it.

In addition, it is possible to configure the model in a way that a cable cut affecting part contained in the SRLG affects only some of the cables in the same SRLG.

*In some programming languages such a class is called a mixin.

†A functor is an object that can be called as it was an ordinary function. Bound callback is a specific type of functor that allows providing ("binding") some of the function parameters that will be used in the call.

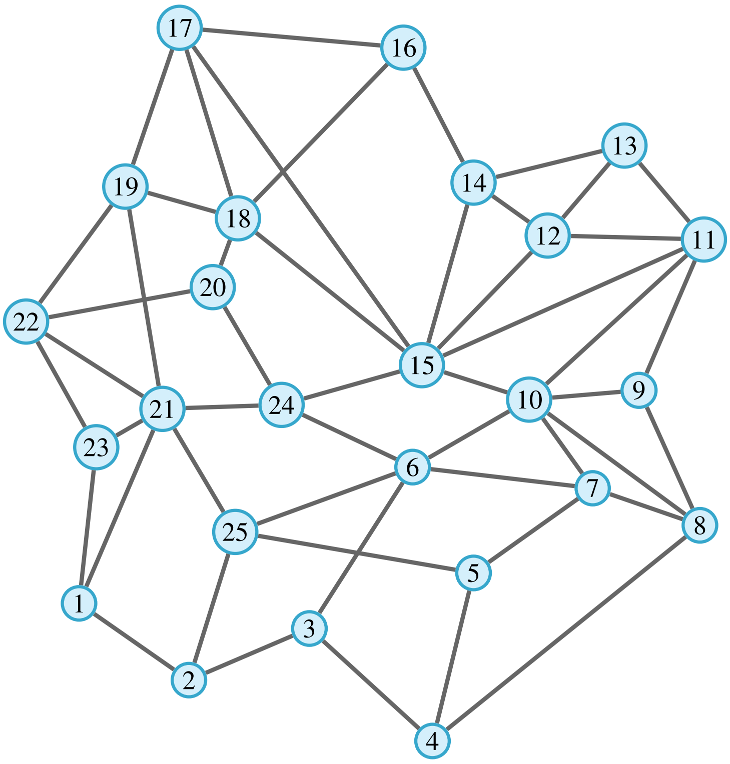


Figure 5.2: Test network topology containing 25 nodes and 50 spans [168].

5.4.3 Data Collection Entities

`FailureRepairTimeTracker` is a class used for tracking uptime and downtime of an object. It can be used for any object that exposes sources for tracing failure event and repair event, but is most commonly used for tracing uptime and downtime of lightpaths and logical channels. It provides interface for getting object uptime and downtime information, as well as computation of availability and unavailability.

`OtnLogicalChannelTracker` is a class that uses multiple `FailureRepairTimeTrackers` to track uptime and downtime of logical channels in the network. It is used for obtaining network availability results from the simulations[‡].

5.5 Analytical Computation of Network Availability and Comparison to Simulation Results

For the evaluation we use the network with 25 nodes and 50 links that can be seen in Figure 5.2. The link length is taken to be Euclidean distance between nodes, resulting in mean link length of 129.40 km with the standard deviation of 47.63 km. The total length of cables in the network is 6470.38 km.

[‡]We expect to eventually generalize these classes for usage by other models, basing on Simulation Automation Framework for Experiments, which will eventually be included in ns-3 [179].

5.5.1 Comparison of Simulation and Analytical Results

Logical channel is considered to be in working state if at least one of working or spare path is in working state, and in failed state otherwise. Path is considered to be in working state if all the links it contains are in working state, or equivalently, none of the links contained in the path are in failed state.

Cables are considered to have failure rate of 114 FIT per kilometer, which translate to mean time to failure (MTTF) of approximately 1000 years per kilometer, which equals 8 760 000 hours [180]. We take mean time to repair (MTTR) to be 6 hours. Finally, we take the nodes to be ideal (have availability equal 1).

We did not specifically consider optical amplifier failures. However, one could simply include optical amplifiers in computation by considering the cable lengths to be larger. Namely, as optical amplifier is considered to have 2850 FIT [181], it has the same failure rate as 25 km of cable. As we assume one amplifier per 100 km of cable, a cable 150 km long has one optical amplifier and will have the same failure rate as the cable that has 175 km.

To compare analytical availability results to results obtained by using Monte Carlo simulation, we evaluate the availability of the following two logical channels:

- logical channel between nodes 1 and 4, having working path going over nodes 2 and 3, and spare path going over nodes 21, 25, and 5 (shown in Figures 5.3 and 5.2);
- logical channel between nodes 8 and 18, having working path going over nodes 10 and 15, and spare path going over nodes 7, 6, 24 and 20 (shown in Figure 5.2 along with the rest of the network).

We denote availability of logical channel between nodes i and j by A^{lch}_{i-j}, availability of working (spare) path between nodes i and j by A^{work}_{i-j} (A^{spar}_{i-j}), and availability of physical link between nodes i and j by A_{i-j}.

Using the analytical approach we obtain availabilities for logical channels as follows.

$$
\begin{aligned}
A^{lch}_{1-4} &= A^{work}_{1-4} + A^{spar}_{1-4} - A^{work}_{1-4} \cdot A^{spar}_{1-4} = \\
&= A_{1-2} \cdot A_{2-3} \cdot A_{3-4} + A_{1-21} \cdot A_{21-25} \cdot A_{25-5} \cdot A_{5-4} - \\
&- A_{1-2} \cdot A_{2-3} \cdot A_{3-4} \cdot A_{1-21} \cdot A_{21-25} \cdot A_{25-5} \cdot A_{5-4} \\
&= 0.999999911055623
\end{aligned}
$$

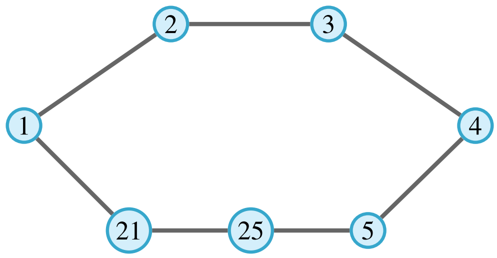


Figure 5.3: Part of the test network used by working and spare paths between nodes 1 and 4 (case without SRLGs).

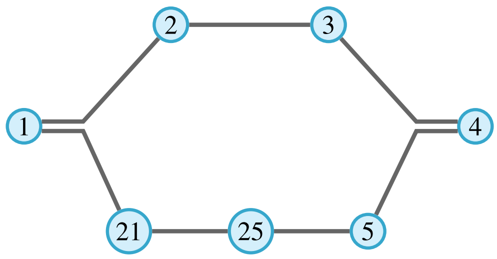


Figure 5.4: Part of the test network used by working and spare paths between nodes 1 and 4 (case with SRLGs).

$$\begin{aligned}
A^{lch}_{8-18} &= A^{work}_{8-18} + A^{spar}_{8-18} - A^{work}_{8-18} \cdot A^{spar}_{8-18} = \\
&= A_{8-10} \cdot A_{10-15} \cdot A_{15-18} + A_{8-7} \cdot A_{7-6} \cdot A_{6-24} \cdot \\
&\cdot \; A_{24-20} \cdot A_{20-18} - A_{8-10} \cdot A_{10-15} \cdot A_{15-18} \cdot A_{8-7} \cdot \\
&\cdot \; A_{7-6} \cdot A_{6-24} \cdot A_{24-20} \cdot A_{20-18} = \\
&= 0.999999902632511
\end{aligned}$$

We now consider the case in which working and spare paths contain coincident SRLGs. Due to complexity of analytical computation for arbitrary failure correlation between cables contained in SRLG, we study the specific case with failure correlation equal to 1. For analytical approach, such failure correlation implies that one computes availability of an SRLG as it was a single cable. Furthermore, the SRLG is in series availability structure with the rest of the cables in the path.

We take two coincident SRLGs to be present at links incident to source and termination node of logical channels, each 5 km long. Specifically,

- logical channel between nodes 1 and 4 passes through SRLG that contains parts of links $1-2$ and $1-21$ and through SRLG that contains parts of links $3-4$ and $5-4$ (shown in Figures 5.3 and 5.2),
- logical channel between nodes 8 and 18 passes through SRLG that contains parts of links $8-10$ and $8-7$, and through SRLG that contains parts of links $15-18$ and $20-18$

(shown in Figure 5.2 along with the rest of the network).

By A_i^{srlg} we denote the availability of SRLG at node i, and by A_{i-j}^* we denote the availability of the part of the cable between nodes i and j that is not contained in the SRLG.

$$
\begin{aligned}
A_{1-4}^{lch} &= A_{1-4}^{work} + A_{1-4}^{spar} - A_{1-4}^{work} \cdot A_{1-4}^{spar} = \\
&= A_1^{srlg} \cdot (A_{1-2}^* \cdot A_{2-3} \cdot A_{3-4}^* + A_{1-21}^* \cdot A_{21-25} \cdot \\
&\cdot \; A_{25-5} \cdot A_{5-4}^* - A_{1-2}^* \cdot A_{2-3} \cdot A_{3-4}^* \cdot A_{1-21}^* \cdot \\
&\cdot \; A_{21-25} \cdot A_{25-5} \cdot A_{5-4}^*) \cdot A_4^{srlg} = \\
&= 0.999993065984851
\end{aligned}
$$

$$
\begin{aligned}
A_{8-18}^{lch} &= A_{8-18}^{work} + A_{8-18}^{spar} - A_{8-18}^{work} \cdot A_{8-18}^{spar} = \\
&= A_8^{srlg} \cdot (A_{8-10}^* \cdot A_{10-15} \cdot A_{15-18}^* + A_{8-7}^* \cdot A_{7-6} \cdot \\
&\cdot \; A_{6-24} \cdot A_{24-20} \cdot A_{20-18}^* - A_{8-10}^* \cdot A_{10-15} \cdot \\
&\cdot \; A_{15-18}^* \cdot A_{8-7}^* \cdot A_{7-6} \cdot A_{6-24} \cdot A_{24-20} \cdot \\
&\cdot \; A_{20-18}^*) \cdot A_{18}^{srlg} = \\
&= 0.999993057590914
\end{aligned}
$$

By doing 1000 runs[§] of Monte Carlo simulation having 10^9 hours of simulated time per run, we get the availabilites shown in Table 5.1. We set the requirement for both standard deviation and absolute difference between analytical result and simulation result mean to be lower than 10^{-5}, which is the threshold for "five nines" availability. Since the standard deviation is in order of magnitude of 10^{-8} in case without SRLGs and 10^{-7} in case with SRLGs, we consider the number of runs was sufficient. Comparing simulation mean and analytical results gives absolute difference that is five orders of magnitude below 10^{-5} (four orders of magnitude in case with SRLGs). Therefore, we consider the model used in the simulation validated, and have reasonable confidence that it is suitable for general use.

[§]For doing multiple runs of a single simulation scenario, our model uses high-level interface provided by ns-3 (description can be found in [74]). For the purpose of pseudorandom number generation, ns-3 provides built-in MRG32k3a [87] generator. MRG32k3a provides $1.8 \cdot 10^{19}$ independent sequences of random numbers, each containing $2.3 \cdot 10^{15}$ subsequences. Each subsequence has period $7.6 \cdot 10^{22}$.

Table 5.1: Difference between simulation and analytical results.

Logical channel between nodes	**Simulation availability result**	**Standard deviation of availability**	**Absolute difference between simulation and analytical result**
Nodes 1 and 4 (no SRLGs)	0.999 999 910 399	1.916×10^{-8}	6.56×10^{-10}
Nodes 8 and 18 (no SRLGs)	0.999 999 903 363	1.988×10^{-8}	7.30×10^{-10}
Nodes 1 and 4 (with SRLGs)	0.999 993 075 031	2.1123×10^{-7}	9.047×10^{-9}
Nodes 8 and 18 (with SRLGs)	0.999 993 060 931	2.0268×10^{-7}	3.340×10^{-9}

5.6 Case Study

For the evaluation we use three networks. In addition to the network already shown, we use two more: one with 20 nodes and 40 links, and other with 30 nodes and 60 links, that can be seen in Figures 5.5 and 5.6.

5.6.1 Scenario Description

We evaluate the scenario where all pairs of nodes have bidirectional logical channels. As the test network has 20 (25, 30) nodes, 190 (300, 435) bidirectional channels are established. We use SRLG-aware routing that sets up working and spare paths for each logical channel that are both link and SRLG-disjoint if possible, and link-disjoint otherwise. We use DBPP scheme (described in detail in Subsection 4.1.3).

Logical channel is considered to be in working state if at least one of working or spare path is in working state, and in failed state otherwise. Path is considered to be in working state if all the links it contains are in working state, or equivalently, none of the links contained in the path are in failed state.

Cables are considered to have failure rate of 114 FIT per kilometer, which translate to MTTF of approximately 1000 years per kilometer, which equals 8 760 000 hours [180]. We take MTTR to be 6 hours. Finally, we take the nodes to be ideal (have availability equal 1).

We used the s,t- and g-availability as measures of network availability.

We simulate the scenarios with 20, 30, 40, 60, 80 SRLGs present in the network, each containing two cables. For scenarios utilizing 20 node and 30 node networks we also consider

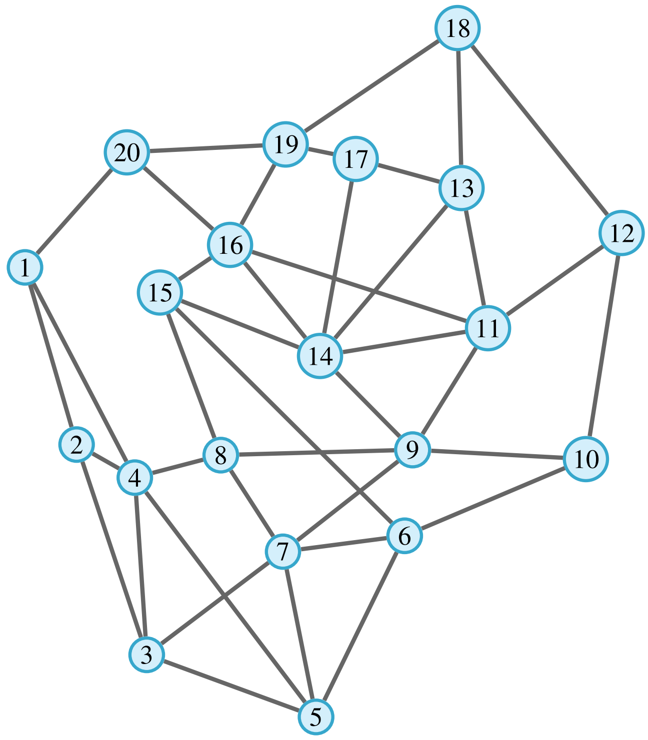


Figure 5.5: Test network topology containing 20 nodes and 40 links [3]. The link length is again taken to be Euclidean distance between nodes, resulting in mean link length of 131.02 km with the standard deviation of 43.98 km. The total length of cables in the network is 5240.65 km.

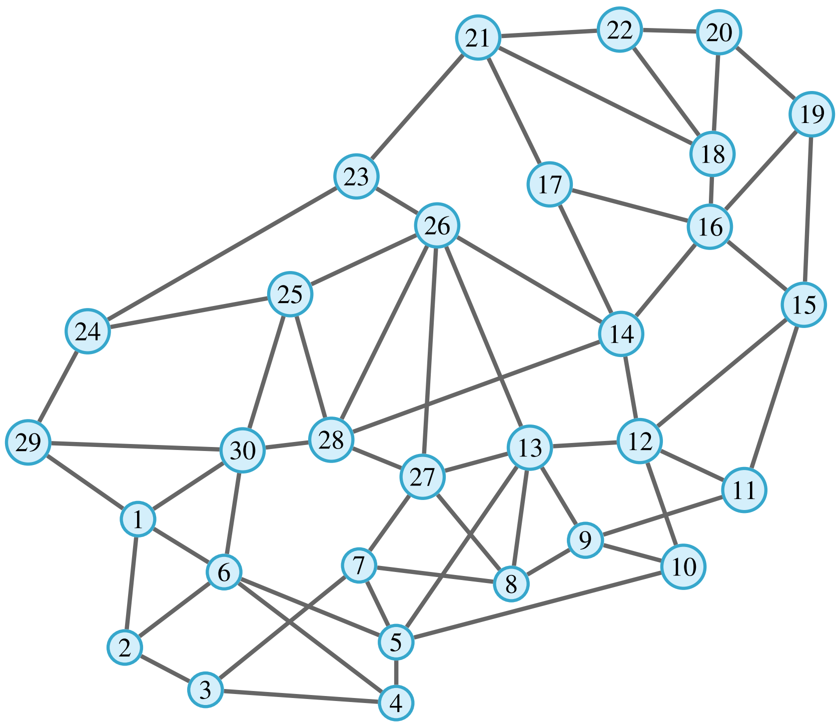


Figure 5.6: Test network topology containing 30 nodes and 60 links [3]. The link length is again taken to be Euclidean distance between nodes, resulting in mean link length of 118.57 km with the standard deviation of 44.18 km. The total length of cables in the network is 7113.95 km.

scenarios with 90 and 120 SRLGs. For each of these numbers of SRLGs in the network we simulate scenarios with SRLG mean lengths of:

- 1.0 km, 2.0 km, 3.0 km, 4.0 km and 5.0 km (for 25 node network),
- 0.5 km, 1.0 km, and 2.0 km (for 20 and 30 node networks).

For 20 node and 30 node networks we simulate two scenarios: one scenario having only coincident SRLGs and other containing general SRLGs.

Based on real world data presented in [165, 178], we set failure correlation between cables contained in the same SRLG to be 0.7. The consequence of this failure correlation is that, on average, 70% of the time a failure of a part of cable contained in SRLG will affect both cables, and the rest of the time will affect only one. Also, we assume that upon repair, both cables will be repaired in the common part.

We evaluate the availability of the 20 node and 30 node networks in terms of both s,t-unavailability and g-unavailability by doing 3000 runs[¶] of Monte Carlo simulation having 10^9 hours of simulated time per run for each scenario described. For 25 node network we did 1200 runs with same amount of simulated time.

In Section 5.7 we evaluate the reasons for choosing this number of runs.

5.6.2 Simulation Results and Discussion

We first turn our attention to 25 node network. The unavailabilities obtained by Monte Carlo simulation are shown in Figure 5.7, along with "Five nines" availability threshold line.

Simulation results show that linear increase in mean SRLG length causes linear increase both in s,t-unavailability and g-unavailability for all numbers of SRLGs evaluated. We used the least squares method for obtaining line coefficients that fit simulation results. The coefficients obtained are shown in Table 5.2. This results in sum of residuals is in each case below y values by seven or more orders of magnitude, so we consider the line to fit our results well.

The unavailabilities obtained by Monte Carlo simulation are show in Figures 5.8 and 5.9 for scenarios using 20 node and 30 node networks, along with "five nines" availability threshold line.

We can again observe that increasing SRLG length does increase unavailability. We also observe that while doubling the number of SRLGs in the network in effect approximately doubles the unavailability, doubling the length does not increase it by such a large margin. This can, be explained by longer paths in presence of more SRLGs. Namely, since routing algorithm creates SRLG-disjoint working and spare paths (if such can be found), higher number of SRLGs will

[¶]For doing multiple independent runs of a single simulation scenario, our model uses high-level interface provided by ns-3 (description of this interface can be found in ns-3 manual, which is available online at `http://www.nsnam.org`). For the purpose of pseudorandom number generation, ns-3 provides built-in MRG32k3a generator. MRG32k3a provides $1.8 \cdot 10^{19}$ independent sequences of random numbers, each containing $2.3 \cdot 10^{15}$ subsequences. Each subsequence has period $7.6 \cdot 10^{22}$.

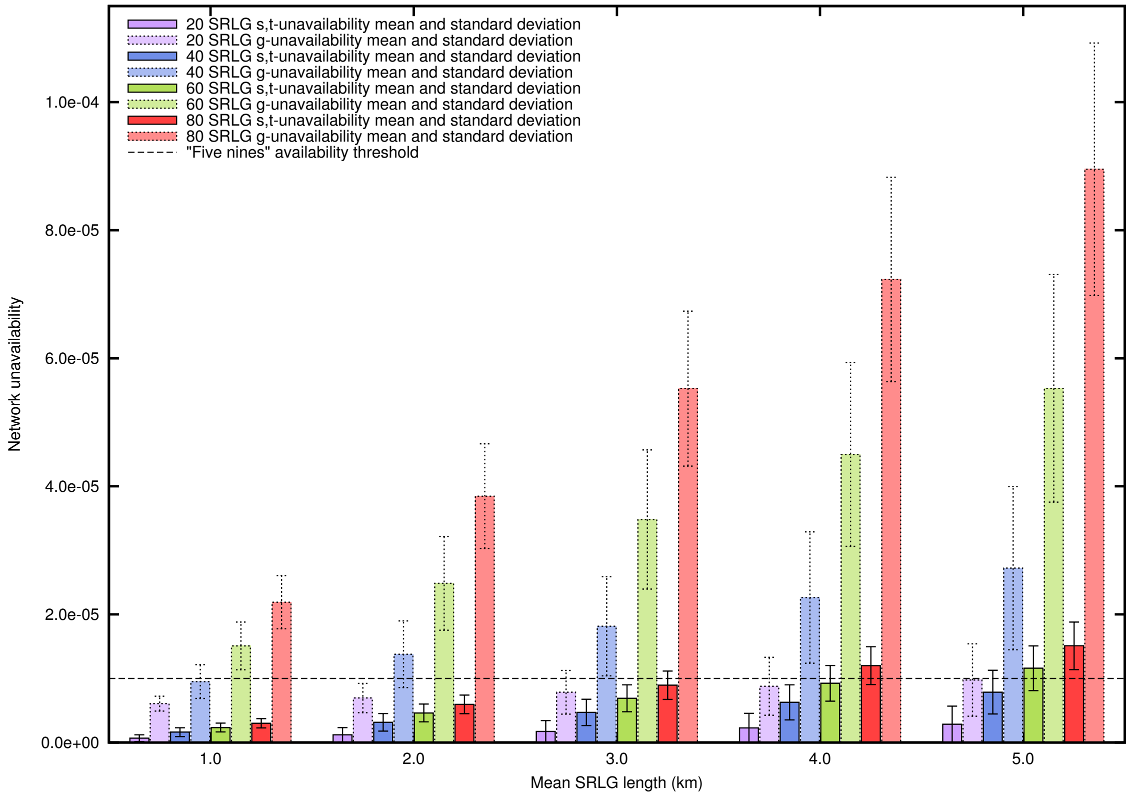


Figure 5.7: Monte Carlo simulation results for network unavailability for varying number of SRLGs and varying mean of SRLG physical length.

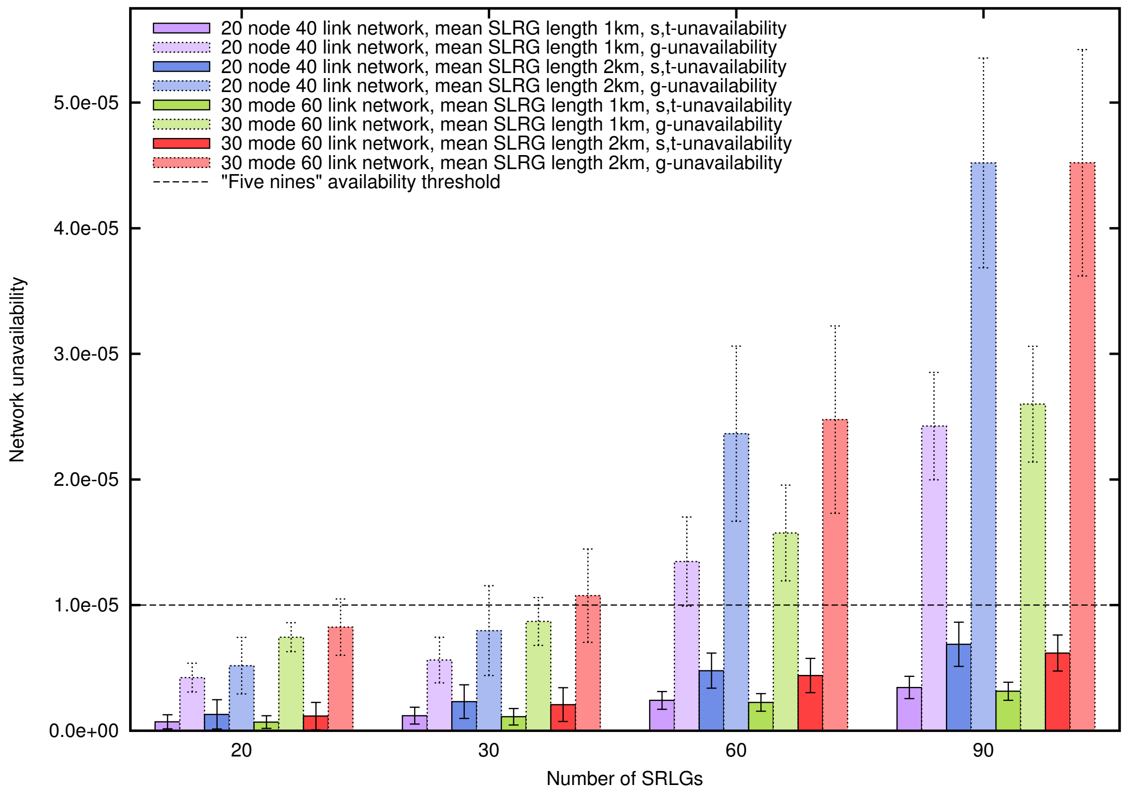


Figure 5.8: Simulation results for network unavailability: comparison of 20 node 40 link network and 30 node 60 link network with coincident SRLGs.

Table 5.2: Line coefficients obtained using least squares method.

Simulation case	**Line slope**	**Line y-intercept**	**Sum of residuals**
20 SRLG s,t-unavailability	5.403×10^{-7}	1.268×10^{-7}	9.891×10^{-16}
20 SRLG g-unavailability	9.203×10^{-7}	5.117×10^{-6}	4.03×10^{-15}
40 SRLG s,t-unavailability	1.56×10^{-6}	4.06×10^{-8}	1.314×10^{-15}
40 SRLG g-unavailability	4.428×10^{-6}	4.973×10^{-6}	4.063×10^{-14}
60 SRLG s,t-unavailability	2.315×10^{-6}	-8.769×10^{-9}	3.231×10^{-15}
60 SRLG g-unavailability	1.006×10^{-5}	4.832×10^{-6}	1.147×10^{-13}
80 SRLG s,t-unavailability	3.025×10^{-6}	-7.743×10^{-8}	1.01×10^{-14}
80 SRLG g-unavailability	1.691×10^{-5}	4.772×10^{-6}	1.641×10^{-13}

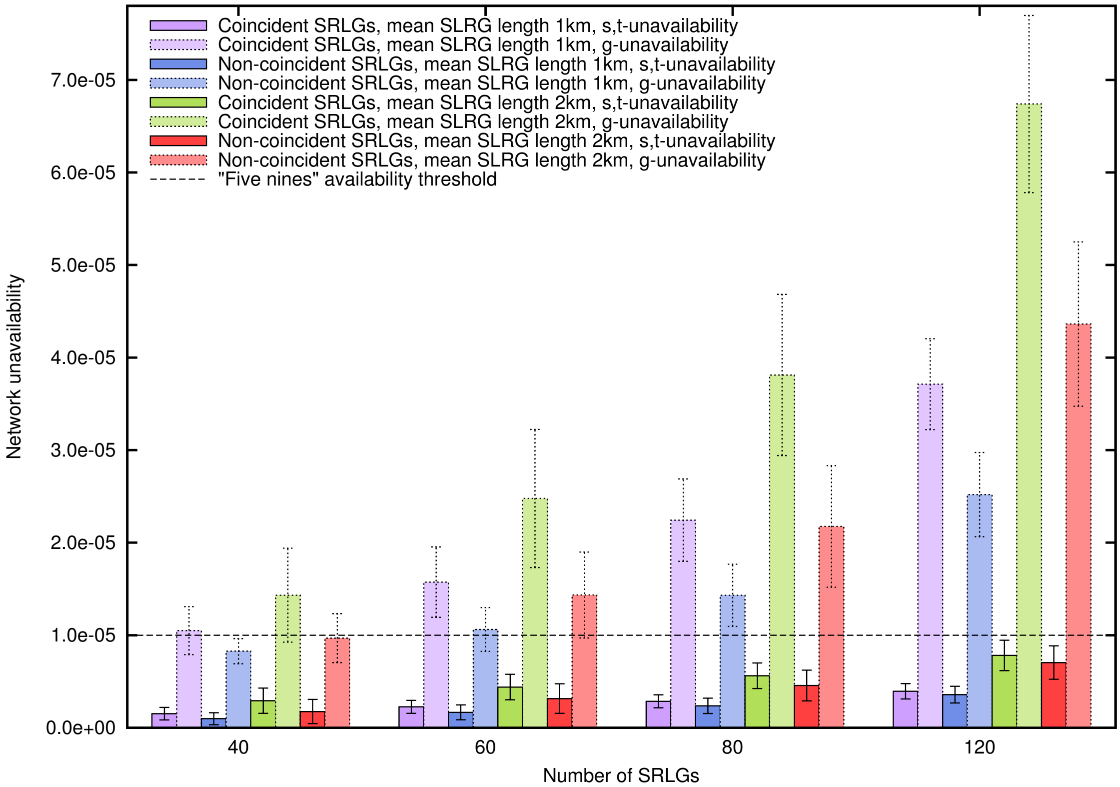


Figure 5.9: Simulation results for network unavailability: comparison of coincident SRLGs and general SRLGs on 30 node 60 link network.

result in some logical channels having longer paths to fulfill SRLG-disjointedness requirement.

We can observe that impact on network unavailability of higher number and length of SRLGs is slightly higher in the network with 30 nodes and 60 links, which can be explained by larger number of logical channels established compared to a network with 20 nodes and 40 links. A result of a larger number of logical channels established is that, on average, each SRLG failure affects more paths.

What might seem unexpected at first is that coincident SRLGs of the same length more negatively impact network availability then general SLRGs. This can be explained by two factors. Firstly, we can observe that routing algorithm creates longer paths in presence of coincident SRLGs. Namely, a coincident SRLG at a particular node will result in at least some of the spare paths originating or terminating at that node being longer to fulfill SRLG-disjointedness requirement. In case an SRLG is not coincident, it is less likely that it will be required to take longer path to fulfill SRLG-disjointedness requirement when creating working and spare path of the same logical channel. Secondly, in case a routing algorithm made a path that is only link-disjoint but not SRLG-disjoint a failure of non-coincident SRLG is less likely to affect both working and spare path of the same logical channel than a failure of coincident SRLG.

It is also worthy noting that standard deviation also increases with increase in number and mean length of SRLGs. Such an increase in standard deviation can be explained by larger variety of scenarios that can happen. Namely, in case with more and longer SRLGs, the probability that failure of a cable in the network will hit a SRLG also increases, but still does not equal 1. Therefore, a SRLG will be hit more often, but still not by every failure that occurs.

5.7 Simulation Precision

More generally, variance in results in the same order of magnitude as mean can be explained by the fact that both metrics we use, s,t- and g-unavailability, depend on the state of all components in the network.

To choose the appropriate number of Monte Carlo simulation runs one has to do to get results, we have to look at the change of resulting mean and variance value over the increasing number of runs. We define the change of value as the absolute difference between current and previous value divided with previous value.

In all the scenarios we simulated for the case study using 20 node and 30 node networks described earlier, the change of mean and variance value is below 5% after 600 runs, and below 1% after 2800 runs. This finding, along with model validation, gives us reasonable confidence in results we obtained.

5.8 Chapter Conclusions

We expected that physically longer SRLGs will more negatively impact logical channel and network availability. To evaluate this, we developed a new model of optical network components and implemented it in network simulator ns-3. We validated our model in specific cases by comparing Monte Carlo simulation results for availability analysis to analytical results. Results fulfilled our expectations.

We wanted to evaluate the impact of SRLG properties on optical network availability. The exact network availability results for complex cases are hard to obtain analytically, so we used Monte Carlo simulation to get results and evaluate different possible improvements. To do so, we developed a new model of optical network components with support for evaluation of availability, implemented it in network simulator ns-3, and validated it against analytical results. Simulation results for the case study show that increase in length of SRLGs increases unavailability, but less than increase in number of SRLGs. Also, non-coincident SRLGs increase network unavailability less than coincident SRLGs.

In more complex cases other factors impacting network availability could be studied. Physical factors such as geographical location, altitude and urbanization level of the area can be taken into account as well as network characteristics such as routing and wavelength assignment resulting from traffic demands.

Since elimination of all SRLGs is costly, partial elimination in terms of shortening physical length and reducing the number of SRLGs can be a viable alternative. While the exact network availability results are hard to obtain analytically for non-trivial cases, it is possible to use Monte Carlo simulation to get approximate results and evaluate different possible improvements.

Finally, in spirit of free open source software, we plan to write detailed documentation describing the features our model and submit our code for review and inclusion as a part of ns-3 source code distribution.

Chapter 6

Impact of Correlated Failures on Various Topology Models

6.1 Introduction and Motivation

Researchers in the telecommunications field often need to assess new algorithms and protocols over realistic topologies. So far, they have widely used topologies that are either regular, e.g. tree, mesh, for analytic studies of algorithmic performance, or synthetic randomly generated ones in the case of running simulations. Even more, reference topologies [182, 183] or instances of real topologies [184, 185, 186, 187] are employed wherever available, since telecom operators are usually reluctant to share such information for business competitiveness and security reasons (e.g. to aggravate physical-layer attacks).

However, as real-world topology data are becoming more and more available, the structural and geographic properties of telecommunication networks are analyzed in order to characterize and model such topologies, mainly making use of graph theory tools. Despite the engineers' overriding role in the case of networks, emergent and unplanned topological traits usually appear in both the logical [188] and the physical level [189]. It has been found that the physical topologies can rarely be described by traditional patterns such as star, bus, ring, hierarchical or full mesh graphs and thus a variety of approaches from complex network theory have been discussed lately on the formation of appropriate network models. Recently, Çetinkaya et al. evaluated the fitness of geographical graph generators for modeling physical level topologies [190]. They evaluated four geographical graph models (Gabriel, geometric, population-weighted geographical threshold, Waxman) and drew to the conclusion that while none of these models capture the structure of real networks perfectly, though Gabriel graphs best capture grid-like structure of physical level topologies.

It is natural to expect that the details of the underlying network topology have an impact on the availability of network services. Especially, when moving from small to larger networks, beyond increasing the length of end-to-end path, there is evidence that shared risk link groups (SRLGs) will more probably be present and negatively impact availability. (A shared risk link group is a structure containing two or more logically disjoint links that share a physical location and are subject to failing at the same time.) In particular, Segovia, Calle and Villa analyzed the network availability for six different physical network topologies [191], differing in number of nodes and links, average node degree, network diameter, link length and other indices. They inferred that large topologies have very different average availability values from smaller topologies, and that difference in availability in smaller topologies could not be observed.

Meanwhile, there has been considerable research on the impact of SRLGs on network availability. Doucette et al. studied capacity requirements in the network in presence of SRLGs, and proposed a design model that included elimination of known SRLGs within budget limits and covering others with additional capacity [161]. We previously analyzed the impact of SRLG length variation on network availability using a specific test topology, and concluded that unavailability increases linearly with increasing SRLG length [7].

Building upon the work described above, in this chapter we compare six physical topology models in terms of resulting network and logical channel availability. We specifically evaluate availability in presence of SRLGs against the scenario where no SRLGs are present. While failure dependency – inherent in SRLGs – makes analytical computation of availability complicated, we make use of Monte Carlo simulation utilizing optical network availability model [7] implemented by network simulator ns-3 [70, 192] to obtain results. The model we use is described in Sections 3.5 and 5.3

We expect that there will be a significant difference in network availability for different topology models, and that the impact of SRLGs on different topologies will also considerably vary. We furthermore anticipate being able to correlate impact of SRLGs with certain topological properties.

This chapter is organized as follows: in Section 6.2 we describe the topology models we have chosen for this study, in Section 6.3 we refer to the topology implementation details and statistical properties and in Section 6.4 we briefly cover basics of availability analysis in the field of optical networks. Finally, in Section 6.5 we present the case study and the simulation results, while in Section 6.6 we conclude with some directions and plans for future work.

6.2 Network Topologies

The recent appearance of geographic graph generators allows the creation of several realistic synthetic graphs for extensive simulation studies. Such graph models generate topologies that fairly fit the observed real-world non-trivial topological features that are neither purely regular nor purely random. The most well-established physical level models are the Random Geometric

Graph model [193], the Gabriel Graph model [194], the Relative Neighborhood Graph model [195], the K-Nearest Neighbor Graph model [196], the Waxman model [197] and the Spatial Barabási-Albert (or Preferential Attachment) model [198, 199, 200], additionally to many others, less popular, such as the Geographical Threshold Graph model [201, 202], the Transit-Stub [203], the KU-LoCGen [204], the HINT [205], and so on [206]. However, the above topology generators do not take into account network design objectives and constraints such as minimizing the latency, dimensioning the links, adding redundancy or minimizing the network budget. Instead, their main objective is to be realistic in terms of fitting the properties of observed real networks, so they serve different purpose than algorithms for optimized physical topology generation, e.g. [207].

Apart from the inherent graph-theoretic interest when studying spatial graph generation, the evaluation of such topologies under failure scenarios always can provide critical information about the network behavior and moreover contributes to understanding the network availability. Regarding the SRLG related literature, the usage of synthetic graph topologies is somehow narrow and rather limited to the Waxman and Barabási-Albert models. Particularly, in [208] the authors presented an IP fast reroute mechanism for SRLG failures in routing protocols without global topology information. Through simulations on both Waxman and Barabási-Albert topologies, they confirmed that their mechanism can achieve a repair coverage close to 100% for different SRLG size. Furthermore, for the experiment setup in [209], in which a tool for network fault diagnosis was presented, the authors used either Waxman or Barabási-Albert as a physical connectivity pattern. As well, the authors in [184] used synthetic network topologies based on the Waxman model, together with some publicly available real topologies, for their evaluation methods of IP fast reroute schemes. Likewise, in [185] where the fault localization problem was considered, the authors employed Waxman-based topologies along with real-world topologies in their extensive simulations with the intention of demonstrating the effectiveness of the proposed monitoring technique. Besides, in [186] the performance of the proposed fast reroute scheme was validated under a variety of real and synthetic Waxman topologies. Similarly, real and Waxman graphs, jointly with 2-level hierarchical graphs and purely random graphs were used in [187] for experiments on efficient load balancing under a wide range of failure scenarios.

6.2.1 Random Geometric Graph Model

A random geometric graph is a random undirected graph drawn on a bounded region, e.g. the unit square or on any d-dimensional Euclidean space. It is generated as follows [193]. First n nodes are placed (independent and identically distributed) uniformly at random on the region. Consequently for some specific distance threshold parameter r, nodes i and j are connected if and only if the distance between them is at most r:

$$d(i,j) \leq r \tag{6.1}$$

where $d(i,j)$ is the Euclidean distance between the two nodes i and j. Modeling random networks in this way is simple and easy to implement, and sometimes a more realistic alternative to the classical random graph models of Erdős and Rényi [210].

6.2.2 Gabriel Graph Model

The Gabriel graphs are named after K.R. Gabriel, who introduced them in a paper with R.R. Sokal in 1969 [194]. In this connection scheme, two nodes are connected directly if and only if there are no other nodes that fall inside the circle (or sphere in three dimensions) associated with the diameter that has the two nodes as endpoints. Mathematically, two nodes i and j, from a set of n nodes, are connected if the square of the distance between them is less than the sum of the squared distance between each of these points and any other point k. That is an undirected graph is constructed by adding edges between nodes i and j if for all nodes k, $k \neq i,j$, where d expresses the Euclidean distance:

$$d(i,j)^2 \leq d(i,k)^2 + d(j,k)^2 \tag{6.2}$$

The Gabriel graphs are useful in modeling graphs with geographic connectivity that resemble grids [194]. These synthetic graphs when compared to AT&T, Level 3, Sprint, and other physical networks, were found to most closely capture the grid-like structure and at the same time achieve the smallest cost among all of the graph models considered in [190]. Moreover, in [211] Bell Atlantic confirmed the Gabriel graph model of their wire centers in Pennsylvania to be remarkably similar to the topology of their inter-office network.

6.2.3 Relative Neighborhood Graph Model

In computational geometry, the relative neighborhood graph is a subgraph of the Gabriel graph. It is an undirected graph created by connecting two nodes i and j, from a set of n nodes, by an edge whenever there does not exist a third node k that is closer to both i and j than they are to each other [195]. In other words an edge is formed between i and j if and only if there is no other node in the interior of the intersection (lune) of the two circles, one with center at i and the other centered at j, with the same radius $d(i,j)$. Formally, the relative neighborhood graph of a set of nodes in the plane is defined as follows: Two nodes i and j define an edge when for all nodes k, $k \neq i, j$, where d expresses the Euclidean distance:

$$d(i,j) \leq \max\{d(i,k), d(j,k)\} \tag{6.3}$$

6.2.4 K-Nearest Neighbor Graph Model

The k-nearest neighbor graph is a graph in which two nodes i and j, from a set of n nodes, are connected by an edge, if the distance between i and j is among the k-th smallest distances from i to all other nodes [196]. The resulting set of edges represents the outcome of the k nearest neighbors query for each node. The directions of the edges may be ignored to lead to an undirected graph.

6.2.5 Waxman Graph Model

The Waxman topology model incorporates location information into random graphs and was introduced by Waxman [197] as a geographic model for the growth of a computer network. In this model the n nodes of the network are uniformly distributed in the plane and they are connected based on a probability derived from the geographical distance between the nodes, in contrast to the Erdős–Reńyi model where the probability is fixed [210]. The probability to have an edge between nodes i and j is given by:

$$P(i,j) = \alpha e^{\frac{-d(i,j)}{\beta L}} \tag{6.4}$$

where $\alpha, \beta \in \langle 0, 1]$, $d(i,j)$ is the Euclidean distance from i to j, and L denotes the maximum distance between any two nodes. An increase in the parameter α increases the edge density, while an increase in β yields a larger ratio of long edges to short edges. The output of this model is an undirected graph with a higher probability for edges between two nodes that are close compared to two nodes further apart.

6.2.6 Spatial Barabási-Albert Graph Model

The Barabási-Albert model generates scale-free networks using a preferential attachment mechanism [212]. It implements the key concept that highly connected vertices are likely to become even more connected. Each new node in this evolving model is connected to a number of existing nodes with a probability proportional to the number of links that the existing nodes already have.

Starting from the Barabási-Albert model, authors in [198] developed a spatial version of the model. In this spatial model, the network grows until n nodes have been created. For a fixed integer $m \geq 1$, each new node is given m links on arrival. These new connections are not chosen uniformly; the new node attaches itself to an existing node with a probability that is proportional to the latter's connectivity, as suggested by Barabási and Albert [212]. Furthermore, since the cost of connecting two nodes increases with geographical distance, the probability that the new node connects to the already connected node is inversely dependent on their distance. Hence, well-connected nodes tend to become even better connected with a bias towards less distant nodes as the network evolves. The probability that the new node i connects to node j is:

$$P(i,j) = \frac{k_j}{\sum_j k_j} \frac{1}{(d(i,j))^a} \tag{6.5}$$

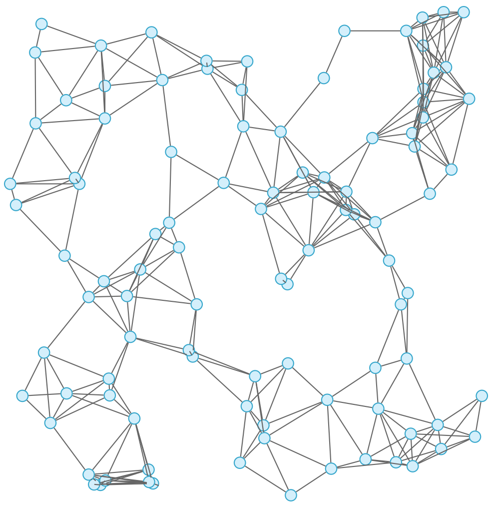
where k is the degree of the node, d is the Euclidean distance and $\alpha \geq 0$ is a parameter for controlling the distance effects. The probabilities are to be normalized such that the sum of all probabilities adds up to one.

This model leads to undirected graphs that take into account the effect of geographical distance and in the same time are characterized by the presence of few nodes with a large number of links (called hubs), while most nodes only have few ones.

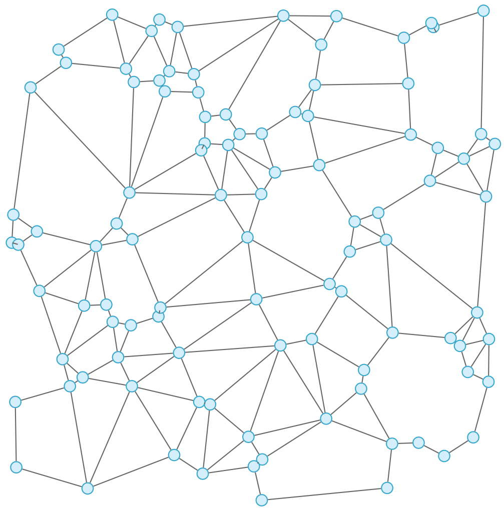
6.3 Implementation and Statistical Properties of the Chosen Models

In this study, we choose a 1000×1000 (i.e. kilometers) square plane as the 2-dimensional Euclidean space where we place $n = 100$ nodes (independent and identically distributed) uniformly at random. Two out of the six models are parameterless (the Gabriel and the Relative Neighborhood models) using only node locations as input, while the rest require at least one parameter. We choose these parameters upholding biconnectivity as a prerequisite. A biconnected graph is a connected graph (no isolated nodes) that if any node or edge were to be removed, the graph will remain connected*. This property is valuable in maintaining a graph with a two-fold redundancy, to avoid disconnection upon the deletion of a single node/edge. On the grounds of this redundancy property, the use of biconnected graphs is very essential in the field of networking and especially SRLG related studies. Simultaneously, we select the parameters' values to minimize the total wiring of the graph, which is another realistic assumption for constructing networks in the physical level. In particular, for the Random Geometric model $r = 150$, for the k-Nearest Neighbor model $k = 3$, for the Waxman model $\alpha = 0.6$ and $\beta = 0.3$, for the Spatial Barabási-Albert model $m = 2$ and as derived by the empirical analysis in [198] the value for $\alpha = 3$. We generate and test families of 100 networks of each of the above models. In Figure 6.1 typical topologies of each model are observed.

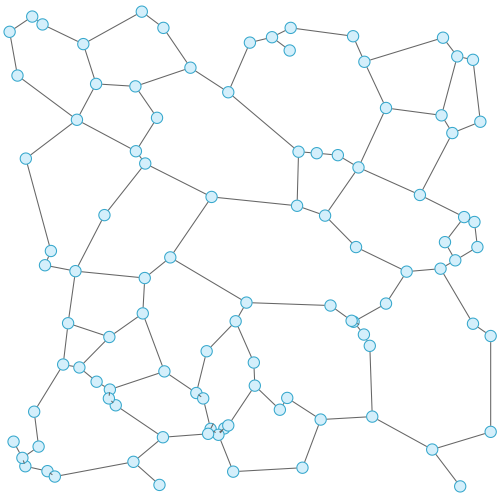
*When measuring the biconnectivity on the Relative Neighborhood graphs, leaf nodes are not considered as articulation points since when a leaf is deleted from a graph, the rest of the graph remains connected.



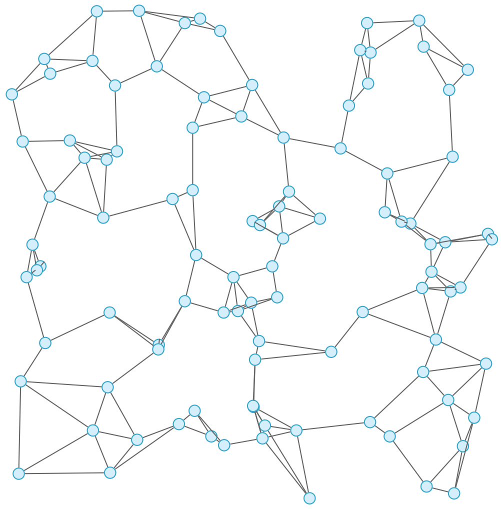
(a) Random Geometric Graph model



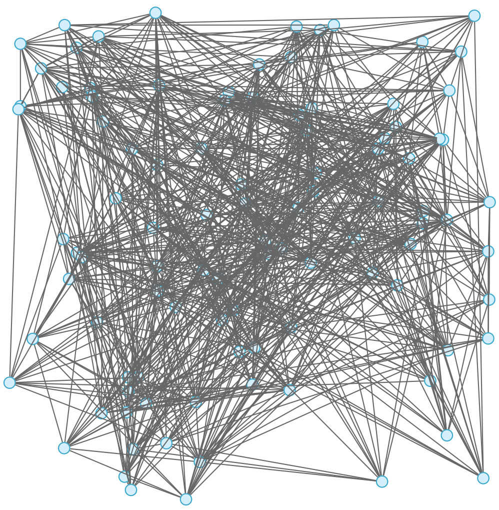
(b) Gabriel Graph model



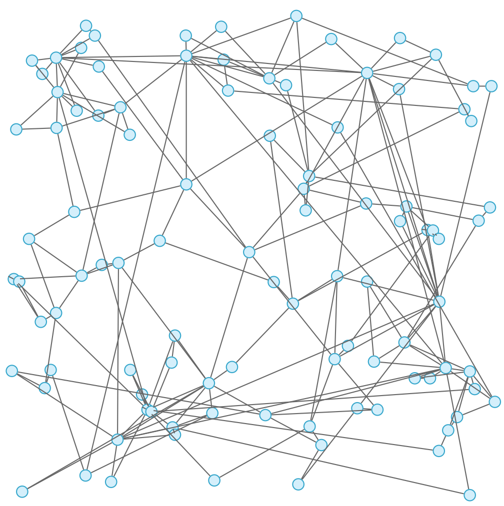
(c) Relative Neighborhood Graph model



(d) 3-Nearest Neighbor Graph model



(e) Waxman Graph model



(f) Spatial Barabási-Albert Graph model

Figure 6.1: Visualizations of instances of the six topology types.

Table 6.1: Basic statistical properties for the six topologies studied (nodes=100, plane=1000 × 1000). Standard deviation appears in the parentheses.

Topology model	Num. of edges	Avg shortest path (hops)	Diam.	Avg clustering coefficient	Total wiring (km)	Avg link length (km)	Mean node degree	Min node degree	Max node degree	Alg. connectivity
Random Geometric Graph	306.1 (20.9)	5.8 (0.4)	10.6 (1.6)	0.6 (0)	30 577.2 (2149.8)	99.9 (1.6)	6.1 (0.4)	2.0 (0)	11.7 (1.5)	0.06 (0.02)
Gabriel Graph	180.5 (6.3)	6.0 (0.2)	11.1 (1.7)	0.2 (0)	17 813.2 (930.2)	98.7 (2.9)	3.6 (0.1)	2.0 (0.2)	6.5 (0.6)	0.07 (0.01)
Relative Neighborhood Graph	120.4 (3.1)	8.5 (0.5)	16.2 (2.1)	0 (0)	9757.5 (532.9)	81.0 (2.8)	2.4 (0.1)	1.0 (0)	3.9 (0.2)	0.03 (0.01)
3-Nearest Neighbor Graph	189.8 (4.0)	7.8 (0.6)	14.0 (2.2)	0.5 (0)	16 530.9 (679.6)	87.1 (2.8)	3.9 (0.1)	3.0 (0)	6.6 (0.7)	0.03 (0.01)
Waxman Graph	943.2 (42.4)	1.9 (0)	2.8 (0.4)	0.2 (0)	355 024.3 (19 025.9)	376.4 (12.1)	18.9 (0.9)	7.7 (1.4)	31.9 (2.8)	6.74 (0.96)
Spatial Barabási-Albert Graph	197.0 (0)	3.4 (0.1)	5.7 (0.6)	0.3 (0.1)	36 247.2 (2031.7)	184.0 (10.3)	3.9 (0)	2.0 (0.2)	18.1 (3.3)	0.29 (0.04)

The basic statistical properties of such topologies are referred here: the average shortest path, the diameter, the average clustering coefficient, the degree (mean, minimum, maximum). The average shortest path or average geodesic path length is defined as the average number of steps along the shortest paths for all possible pairs of network nodes. The diameter of a network is the length (in number of edges) of the longest shortest path between any two nodes in the network. The average clustering coefficient is defined as the average of all n local clustering coefficients C_i, where $C_i = \frac{\text{number of triangles connected to node } i}{\text{number of triples centered on node } i}$. In particular, a triangle indicates that two neighbors of a node are also connected by an edge, while the number of triples indicates the number of permitted edges between the neighbors of a node. The degree of a node is the number of edges directly connected to the node. The total wiring is defined as the sum of edge lengths, while the average link length is defined as the ratio of the summation of all edge lengths to the number of edges, both measured in kilometers here. The algebraic connectivity measures how difficult it is to break the network into islands or individual components [213] and is defined as the second smallest Laplacian eigenvalue. The larger it is, the greater the robustness of a topology against both node and link removal.

In Table 6.1, the basic statistical properties for the six topologies under study are presented. All values are rounded to the nearest tenth decimal, while the standard deviation appears in the parentheses. What turns out notably significant is that the Relative Neighborhood graphs, along with the 3-Nearest Neighbor and the Gabriel graphs show a considerably lower cost in terms of total wiring. The Gabriel graphs have already been found to closely capture the grid-like structure of physical-level networks and at the same time achieve a feasible cost [190]. These three models which have an advantage in total wiring are also the best in terms of average link length, but the worst in the average shortest path and diameter properties. Even though, the main difference among these three models is that the 3-Nearest Neighbor demonstrates a quite higher average clustering coefficient. Although the rest three models are more common in the literature as synthetic topology generators, they produce graphs with high or extremely high total wiring, i.e. Waxman, and a variety of property values. The Waxman graphs appear to have a very low average shortest path and diameter due to their high number of edges, while the Random Geometric graphs show the highest average clustering coefficient. Last, the Spatial Barabási-Albert graphs result in low average shortest path and diameter while maintaining a relatively low mean node degree, compared to the aforementioned two models.

Unfortunately, the diversity in the values of statistical properties (i.e. number of edges, total wiring) may raise potential concerns about performing a legitimate comparison. However, this is both reasonable and unavoidable since each model has – by definition – specific limitations and not all its attributes can be controlled concurrently. For instance, we cannot impose the generation of fewer edges on the Waxman model without letting the existence of isolated nodes. Respectively, in the Random Geometric model we cannot produce a biconnected graph with a lesser value in r, than the one already assigned. This is more evident in the parameterless models, where – by default – we are unable to control the output traits. In short, holding the same number of nodes, retaining biconnectivity and then minimizing cost (where applicable), are the requirements for inclusion in the comparison, albeit we still observe extremely diverse values in some models (e.g. Waxman), which are kept in our analysis due to their prevalence in the related literature.

6.4 Optical Network Availability Analysis

Network availability is a probability that a repairable system will be in operating state at a random moment in time. It can be computed by both analytical and simulation methods. The analytical method uses component mean time to failure (MTTF) and mean time to repair (MTTR) to compute the network availability, by considering the availability of logical channels established in the network. Availability of logical channels can be computed by computing availability of paths they use, which can furthermore be reduced to considering availability of network components in the path. Analytical method relies on serial and parallel relationships between components of a path or paths used by a logical channel, but the relationship among components can become complex in presence of failure dependencies, which are neither serial nor parallel relationships.

Unavailability is defined as a complement of availability. Since availability values are usually very close to 1 (or 100%), it is much easier to compare availability results based on the order of magnitude in unavailability difference.

Monte Carlo simulation can be used for the estimation of network availability. Particularly, it uses random numbers to generate times to failure and times to repair for components in the network, based on their MTTF and MTTR values. Failure and repair events are then handled by the component model implemented in a network simulator. It is possible to make a failure (or a repair) in a particular component to affect other components in a certain way. More specifically, this makes it possible to model complex neither serial nor parallel relationships such as failure dependency.

The network model we use is implemented by discrete event network simulator ns-3. More details of the model and its implementation can be found in [7].

6.5 Case Study

For the evaluation of network availability we use 100 instances for each of the six physical topology models described above, totaling to 600 different physical topologies.

6.5.1 Scenario Description

We evaluate scenarios where all pairs of nodes have bidirectional logical channels, each having working and spare path. As each test network has 100 nodes, 4950 bidirectional logical channels are established. When routing logical channel working and spare path, DBPP scheme is used (more details are given in Subsection 4.1.3). A more detailed traffic model based on either population or other geographical properties could as well be used instead of the full mesh logical channel scheme. Since we use synthetic topologies, such a model would require another randomly generated parameter – or set of parameters – to be introduced. This in turn would affect the results, and therefore make the correlations between topological properties and availability less evident. Additionally, although the effect of node failures could be explored as well, it is considered beyond the scope of this work and therefore the network nodes are assumed to be fully reliable.

We take cables having failure rate of 310 FIT per kilometer (1 FIT = 1 failure in 10^9 hours), which includes fiber and inline amplifier failures [214]. We assume MTTR to be 12 hours and nodes in the network to be ideal[†] (have availability equal 1). We further consider that once a failure of a cable occurs, then all contained fibers will also fail.

Logical channel is considered to be up if at least one of the paths it uses is so, while otherwise it is considered down. A path is regarded to be up if all the contained links are in working state, or in other words, none of the contained links in the path is in a failed state.

We use s,t- and g-availability as two measures of network availability.

The SRLG model which has been used is the one described in [7]. In particular, this model assumes that each SRLG contains parts of two or more cables sharing a physical location. If the cable part contained in SRLG fails, there is a specific probability that the other cables are damaged too. Notably, this probability is set to be 0.7 [178]. It is additionally assumed that all cables are repaired in the common part upon repair.

We simulate the scenarios with no SRLGs and 200 SRLGs present in the network. In the case where SRLGs are present in the network, their length is normally distributed with mean 3.0 km, and each SRLG contains two cables. We take all SRLGs to be coincident, meaning that cables contained in SRLG share a common node. We use SRLG-aware routing that sets up working and spare paths for each logical channel which are both link and SRLG-disjoint if possible, and only link-disjoint otherwise.

For each topology instance we conduct 20 runs of Monte Carlo simulation lasting 10^9 hours of simulated time, resulting in 2000 simulation iterations done per physical topology model for each scenario.

[†]Our model allows configuration of MTTF and MTTR for optical network components contained in nodes. Failures of network components could be considered as well as link failures. However, such consideration falls outside the scope of this work.

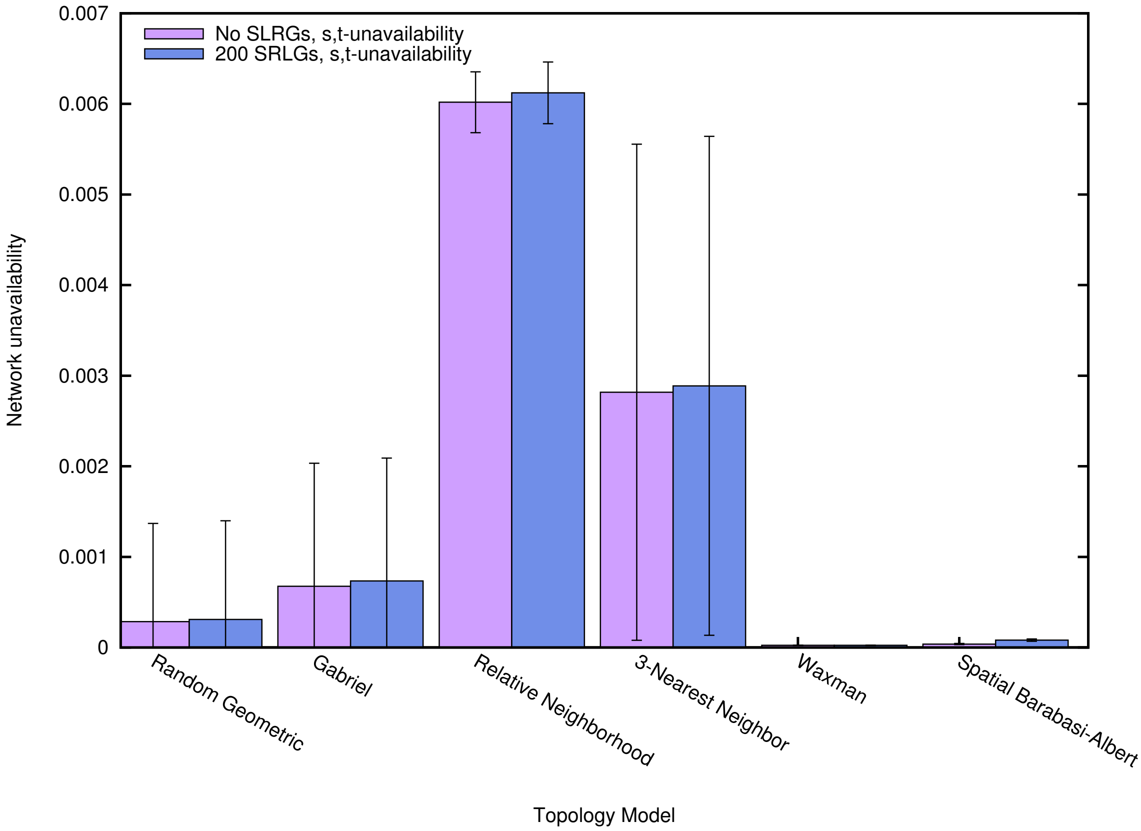


Figure 6.2: Simulation results s,t-unavailability: comparison of scenarios with no SRLGs to scenarios with SRLGs present in the network.

6.5.2 Simulation Results and Discussion

Simulation results presented in Figures 6.2 and 6.3 are obtained by computing mean value and standard deviation on 2000 runs for each topology model and each scenario, as well. It is obvious that there is a significant difference both in g- and s,t-unavailabilities among these models.

To begin with, the Relative Neighborhood model has the highest unavailability among the models presented here. This fact can be fairly perceptible given the presence of leaves in the graph, and also by the zero value in the average clustering coefficient. In addition, this model produces graphs with lower number of edges than other models, which results in inability to find link-disjoint spare paths for some logical channels. This inability could also explain negligible difference in g-unavailability in presence of SRLGs; if it is not possible to find a backup path for some logical channel that is link-disjoint, it will certainly not be possible to find one that is both link- and SRLG-disjoint. On the other hand, the increase in s,t-unavailability can be explained by the increase in average backup path length in presence of SRLGs for those logical channel whose SRLG-disjoint paths could be found.

Continuing, the Random Geometric model has the lowest g-unavailability and is among the lowest with regard to s,t-unavailability. Larger number of edges generally induces a larger

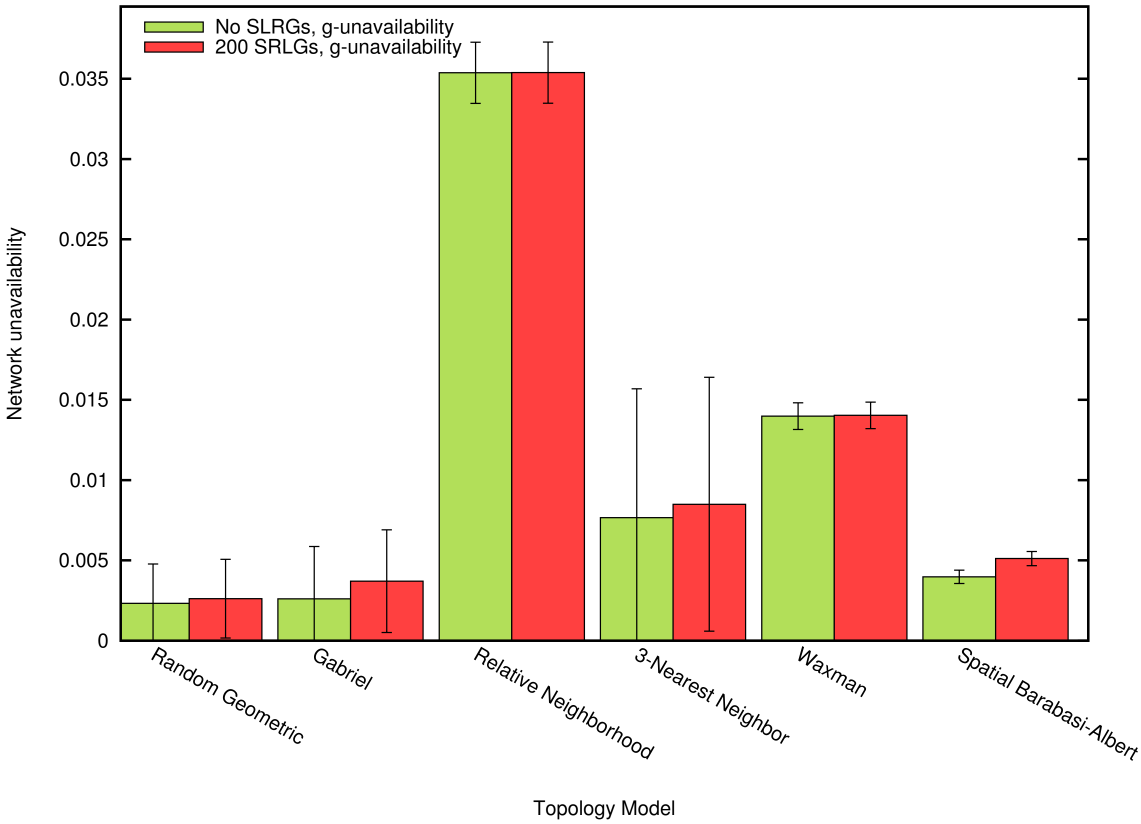


Figure 6.3: Simulation results *g*-unavailability: comparison of scenarios with no SRLGs to scenarios with SRLGs present in the network.

number of possible backup paths. In parallel, a larger number of possible backup paths results in links being shared by fewer number of backup paths set up when establishing logical channels. This in turn results in a single failure on average affecting lower number of logical channels, which results in low value for s,t-unavailability. Relatively high standard deviation can be explained by randomness inherent in the model.

The Gabriel model is similar to the Random Geometric in terms of g-unavailability, but at a much lower number of edges and total wiring. This is reflected in the increase of s,t-unavailability, since the number of possible backup paths in the Gabriel model is lower compared to the Random Geometric model. We can additionally observe that lower number of edges compared to that of the Random Geometric model leads to a more negative SRLGs impact on g-unavailability, due to the fact that SRLG-disjoint backup paths are more unlikely to exist on average.

For the Waxman model, the relatively low s,t-unavailability can be explained by many possible paths due to having almost an order of magnitude more edges than other models. Once again, we deem it necessary to emphasize that despite this last finding, still, the Waxman model is frequently common in literature and thus it is included here for comparison aims. Besides, the relatively high g-unavailability could be attributed to large total wiring and therefore more failure occurrences in time, affecting some of the logical channels. Furthermore, we can observe that in the Waxman model SRLGs have a negligible effect on unavailability; due to many possible paths between two nodes, it is very likely that SRLG-disjoint paths can be found. Negative effect of SRLG failure resulting in two concurrent logical link failures is still present, however.

The 3-Nearest Neighbor model shows much lower s,t-unavailability to Relative Neighborhood, albeit on the same order of magnitude, which is plausibly expected due to the larger number of edges. There is also an even greater improvement in terms of g-unavailability, which however results in noticeable impact of SRLGs. As also with Random Geometric model, high standard deviation can be explained by randomness that is inherent in the model.

Moreover, the Spatial Barabási-Albert model indicates very good performance in terms of both s,t- and g-unavailability, as well. Similarly to the Waxman model, the relatively high total wiring results in a very low s,t-unavailability but this does not hold also for g-unavailability. Additionally, the observation about the effect of SRLGs on the Gabriel model does also hold for the Spatial Barabási-Albert.

Besides, and since the Gabriel model has been found to most closely fit real physical networks [190], the usage of different topology models in availability related experiments could lead to availability miscalculation. Thus, the utilization of models such as the Waxman, the Spatial Barabasi-Albert or the Random Geometric would underestimate the s,t-unavailability, while the usage of models such as the Relative Neighborhood or the 3-Nearest Neighbor would contrary result in an overestimation of this metric. Regarding the estimation of g-unavailability,

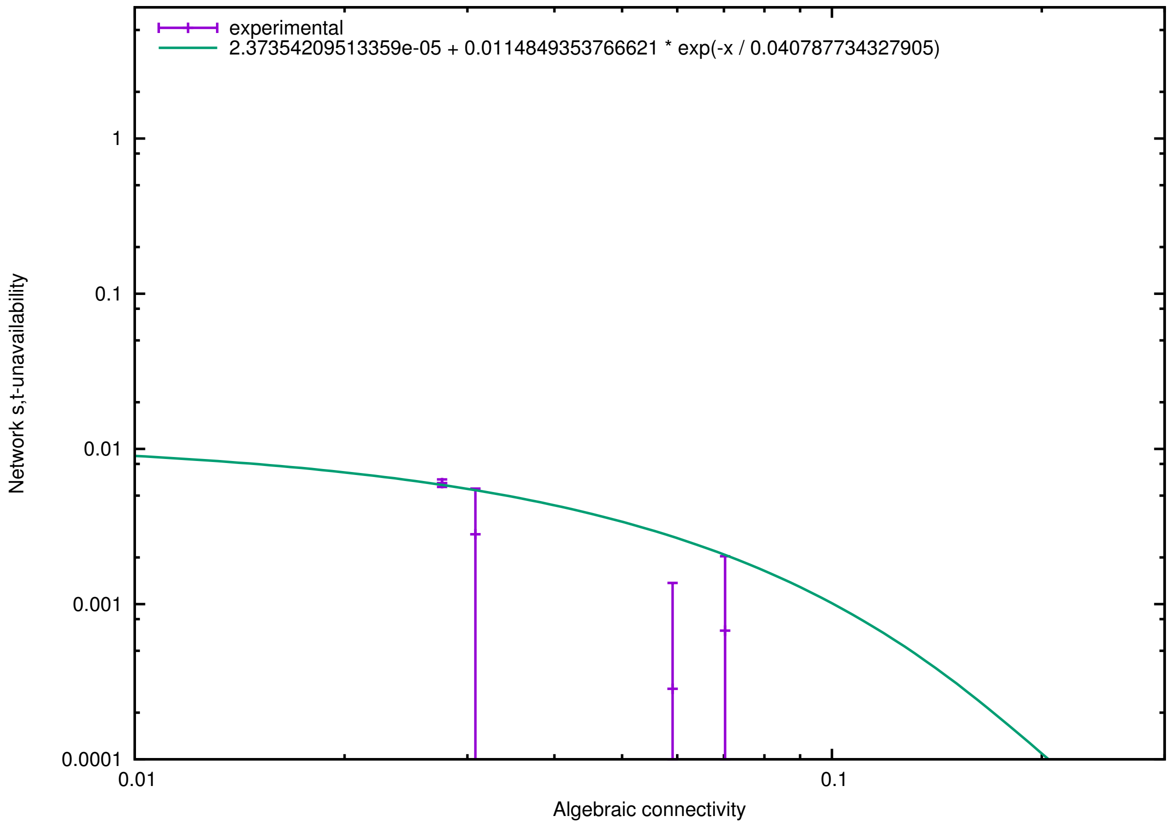


Figure 6.4: Simulation results s,t-unavailability: comparison of scenarios with no SRLGs to scenarios with SRLGs present in the network.

the usage of the Random Geometric model would probably underrate it, while the usage of any other model, among the considered ones, would on the other hand exaggerate its value.

Finally, we observe correlation between algebraic connectivity and s,t-unavailability. Interestingly, we find that s,t-unavailability is subject to exponential decay, since it decreases at a rate proportional to the value of the corresponding topology algebraic connectivity. It has been again shown in literature that algebraic connectivity is informative and predictive of graph robustness, with a direct (although non trivial) relationship [213, 215]. The details of our measurement are shown in Figure 6.4. Although the fit statistics indicate a strong correlation, we should note the usage of only six data points and suggest the need for further research to draw profound conclusions.

6.6 Chapter Conclusions

In this chapter we implemented and used six different physical topology models for investigating their influence on optical telecommunication network availability. We anticipated observing an apparent difference in availability of logical channels and a significant difference in the impact of SRLGs on network availability for the considered topology models. Eventually, the results actually did fulfill the above expectations. On top of that, the findings elaborated above also indicate a coupling between particular topological metrics and optical network availability, albeit not a trivial one. Average shortest path and diameter appear to have a critical effect on s,t-availability, while regarding g-availability it turns up to be influenced by the average shortest path and diameter in combination with the total wiring as well. Nonetheless, any possible correlation between those metrics and network availability seems to be non-trivial and no definitive conclusion has so far been reached about it.

In terms of future work, it would be quite intriguing to further examine and explain the correlations between a richer set of topological metrics and availability measurements. Since it is unlikely that such correlations are trivial ones, let alone including the presence of SRLGs, we are convinced that this research direction will eventually contribute to a better understanding of the network availability determinants. Finally, the challenging study of effective (in terms of network availability) network topology construction based on such results is a direction which warrants further attention and research.

Chapter 7

Shared Risk Link Group-aware Optimization of Routing and Wavelength Assignment

7.1 Introduction and Motivation

In case a working lightpath goes down due to a component failure, a spare lightpath is used until the working is repaired. Routing of working and spare lightpaths is a non-trivial problem and, combined with wavelength assignment, it can be shown to be NP-complete [216]. Many heuristics for solving this problem have been developed over the years, and many special case optimizations have been made.

A particularly interesting special case optimization is routing and wavelength assignment (RWA) for working and spare lightpath in presence of shared risk link groups (SRLGs), groups of links that have a common physical location [159]. Example of a SRLG where two cables share a common exit at a node can be seen in Figure 7.1. Due to shared location, be it a cable, duct or bridge crossing, SLRGs are prone to failing at the same time due to single physical damage. In effect, multiple seemingly unrelated logical failures can occur, for example two link-disjoint (but not SRLG-disjoint) lightpaths can fail at the same time. Therefore, an algorithm for RWA should be designed to avoid common shared risk link groups in working and spare lightpaths, to prevent them from failing at the same time due to a common physical force. Since link- and SRLG-disjoint paths might not exist or be possible to set up in a network, maximum disjoint paths are usually a reasonable substitute. However, finding maximum link disjoint paths is NP-hard problem [216]. Furthermore, it is known that greedy algorithms for it are performing as well as (much more complex) heuristics.

The rest of this chapter is organized in the following sections: the application of maximum disjoint path algorithms in RWA is described in Section 7.2, approach to improving RWA

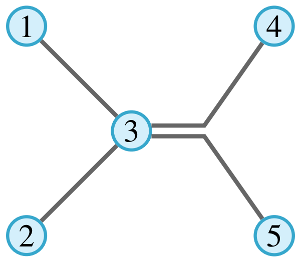


Figure 7.1: Example SRLG containing two cables (3–4 and 3–5) that share an exit at a particular network node. For comparison, there is no SRLG containing cables 1–3 and 2–3.

algorithm is presented in Section 7.3, and case study simulations are done in Section 7.4. Afterwards, we conclude in Section 7.5.

7.2 Routing and Wavelength Assignment in Presence of Shared Risk Link Groups

We now turn our attention to RWA of logical channels in the network. Specifically, we will consider RWA of working and spare paths required for establishment of each channel. Routing requires link- and SRLG-disjointness of working and spare paths. On the other hand, wavelength assignment requires a common unused wavelength on each of the links the path traverses for both paths. The first requirement can be relaxed to maximum disjointness, if completely disjoint paths do not exist. The second requirement can be relaxed if optical network contains support for wavelength conversion, so only an unused wavelength on each link is required, i.e. the wavelength does not need to be the same one in all the links used by the path.

7.2.1 Shared Risk Link Group Disjoint Paths

Despite the fact a network might offer many options for routing working and spare paths for of a particular channel, generally not all of them need to be link- and SRLG-disjoint or satisfy certain limit of path length.

To illustrate link- and SRLG-disjointness, we turn our attention to network shown in Figure 7.2 which offers four possible paths between nodes 1 and 8. Out of those four we need to pick one for working path and one for spare path. One option would be to route two paths as $1-2-3-8$ and $1-4-5-8$. We can see that links $1-2$ and $1-4$, and also $3-8$ and $5-8$ each have a common SRLG, see these two paths despite being link-disjoint share two SRLGs. If instead of $1-2-3-8$ one picks $1-6-2-3-8$ as working path and spare path remains unchanged, only one common SRLG remains between two paths. Finally, paths $1-6-2-3-8$ and $1-4-5-7-8$ are both link- and SRLG-disjoint. Despite the fact that they are longer, the requirement to avoid simultaneous failure of working and spare path is quite often more

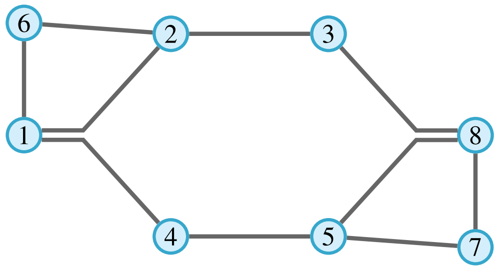


Figure 7.2: Example network used to illustrate the concepts of link- and SRLG-disjointedness.

significant than increase in path length.

RWA problem in presence of shared risk link groups can generally be written as an integer linear program, and software solvers can be applied [166]. However, due to NP-completeness of RWA problem, relaxation techniques and heuristics are commonly used.

7.2.2 Routing and Wavelength Assignment Algorithm

An approach to RWA in presence of SRLGs was studied by Li et.al. [217, 218]. The terminology used in the paper, in particular the SRLG model, is very different from the one we use. The idea behind the algorithm presented in the paper is that one increase path length by SRLG length for each SRLG contained in the path. Since the algorithm described in their work is not directly usable on our model, we will use an adaptation that goes as follows:

1. Compute the working path using Dijkstra shortest path algorithm. Let the set of links used by working path be L_w, and let $S(L_w)$ be a set of all links that contain at least one common SRLG with a link in L_w.
2. To route the spare path, remove from graph links in L_w (links used by working path) and also remove links in $S(L_w)$ (links with commons SRLG with working path). If possible route the spare path using Dijkstra shortest path algorithm and exit with success.
3. Let $s = |S(L_w)|$. Then there are $\binom{s}{r}$ r-subsets of links in $S(L_w)$, and 2^s subsets total. Let $i = 1, 2, \ldots, s$. In $i-th$ step do the following:
   (a) Select next $(s-i)$-subset of links in $S(L_w)$. If all $(s-i)$-subsets have been tried, increment i and continue.
   (b) Remove from graph links in L_w and $s-i$ links selected in subset. Compute the spare path p_{spare} using Dijkstra shortest path algorithm. If the path computation was successful, compute its weight by using the formula

$$weight(p_{spare}) = length(p_{spare}) + length_{srlg}(p_{spare})$$

   where $length_{srlg}()$ is the total length of cables contained in SRLGs on path.
4. Route the spare path by selecting the lowest weight one.

This algorithm will be used as a base for comparison. The maximum SRLG-disjoint path algorithm was also studied by Shao et.al. [219]. The algorithm described in the paper is usable on our model and goes as follows:

1. Compute the working path using Dijkstra shortest path algorithm. Let the set of links used by working path be L_w, and let $S(L_w)$ be a set of all links that contain at least one common SRLG with a link in L_w.
2. To route the spare path, remove from graph links in L_w (links used by working path) and also remove links in $S(L_w)$ (links with commons SRLG with working path). If possible compute the spare path using Dijkstra shortest path algorithm and exit with success.
3. Let $s = |S(L_w)|$. Then there are $\binom{s}{r}$ r-subsets of links in $S(L_w)$, and 2^s subsets total. Let $i = 1, 2, \ldots, s$. In $i-th$ step do the following:
   (a) Select next $(s-i)$-subset of links in $S(L_w)$. If all $(s-i)$-subsets have been tried, increment i and continue.
   (b) Remove from graph links in L_w and $s-i$ links selected in subset. Compute the spare path using Dijkstra shortest path algorithm. If the path computation was successful, route the spare path using the computation result and exit with success.

It is reasonable to route the working path as shortest path since it is used most of the time. It is easy to see that algorithm ends either upon finding a maximum SRLG-disjoint and completely link-disjoint spare path or concluding no link-disjoint path exists.

Note also that it would be trivial to extend the algorithm to find maximum link-disjoint path in addition to maximum SRLG-disjoint path if there is such requirement. Combining link- and SRLG-disjointness with particular weight or coefficient assigned to each is also a possibility.

7.3 Improving Routing and Wavelength Assignment Algorithm

So far, we have used Monte Carlo simulation to obtain availability results in presence of SRLGs due to complex serial-parallel relationship induced by SRLGs. However, it is possible to give reasonable models of an SRLG using analytical approach. This analytical model can then be used for weighting SRLGs in the process of routing working and spare ligthpaths.

7.3.1 Modeling Correlated Failure Relationship

In Section 4.5 we explained the method for computing availability of cables of varying lengths. Assuming cable of length x with mean time to failure of $MTTF$ per kilometer and mean time to repair of $MTTR$, availability $A(x)$ of a cable is

$$A(x) = \frac{\frac{MTTF}{x}}{\frac{MTTF}{x} + MTTR}. \tag{7.1}$$

For cable of length x unavailability $U(x)$ is

$$U(x) = \frac{MTTR}{\frac{MTTF}{x} + MTTR}.$$

We will denote unavailability of two cables with lengths x and y with $U_2(x,y)$. Following the definition of series availability, we have

$$U_2(x,y) = U(x) \times U(y).$$

Since $A = 1 - U$, it follows that availability of two cables with lengths x and y is

$$A_2(x,y) = 1 - (1 - A(x)) \times (1 - A(y)),$$

which simplifies to

$$A_2(x,y) = A(x) + A(y) - A(x) \times A(y). \tag{7.2}$$

Intuitively, we could think of an SRLG with two cables of length x and correlated failure probability $p = 1$ as being a single cable. Since every failure will hit both cables, availability of the SRLG will be the same as that of one cable. Formally, that is

$$A_{SRLG}(x,1) = A(x).$$

With probability p decreasing towards zero, availability of the SRLG will increase. It will finally reach that of two cables in parallel relationship when $p = 0$. Written formally, this is

$$A_{SRLG}(x,0) = A(x) + A(x) - A(x) \times A(x) = 2A(x) - A(x)^2.$$

Considering SRLG as a particular form of two cables of length x and correlated failure probability p, and setting $A(y) = (1-p) \times A(x)$, we can compute availability of SRLG as

$$A_{SRLG}(x,p) = A(x) + (1-p) \times A(x) - (1-p) \times A(x) \times A(x). \tag{7.3}$$

7.3.2 Expressing Lowered Availability with Increased Path Length

We consider a path using n links $l_1, l_2, \ldots, l_n$ and passing through $n-1$ nodes. Assuming nodes to be ideal, we only consider link availabilities. Availability of link l_i can be computed from the length of the cable x_i that contains the link. Say the lengths are $x_1, x_2, \ldots, x_n$. Since all the cables must be working in order for the path to be working, we have series relationship and it follows that availability of path A_p is

$$A_p = A(x_1) \times A(x_2) \times \cdots \times A(x_n).$$

Intensity of cable failures in our model depends solely on cable length. We can approximate the availability of the path by considering it to contain only a single cable. This approach can be repeated $n-1$ times to get

$$\begin{aligned} A(x_1) \times A(x_2) \times \cdots \times A(x_n) &\approx A(x_1 + x_2) \times A(x_3) \times \cdots \times A(x_n) &(7.4)\\ &\approx A(x_1 + x_2 + \cdots + x_n). &(7.5)\end{aligned}$$

Let us note that approximation is done solely to simplify further equations in writing and reduce the amount of computation required; we could just as well derive the following results without approximating path availability.

We assume we have a logical channel with working path of length x and spare path of length y. Additional assumption is that working and spare paths have a common SRLG of length s with correlated failure probability p. We denote the availability of this logical channel by $A_{lc}(x, y, s, p)$.

Considering failures as uncorrelated, per equation 7.2 and simplification we did availability of logical channel is

$$A(x) + A(y) - A(x) \times A(y). \tag{7.6}$$

On the other hand, if we consider SRLG to be in series relationship with both cables, we get that availability equals

$$A_{SRLG}(s, p) \times (A(x-s) + A(y-s) - A(x-s) \times A(y-s)). \tag{7.7}$$

Equating 7.7 with 7.6 can be done in two ways. Presence of SRLG between working and spare path results in lowered availability of logical channel, which is equivalent to longer path. In other words, path length gets increased by *inc*, namely

$$\begin{aligned} A_{lc}(x, y, s, p) &= A(x) + A(y + inc) - A(x) \times A(y + inc) &(7.8)\\ &= A_{SRLG}(s, p) \times (A(x-s) + A(y-s) - A(x-s) \times A(y-s)). &(7.9)\end{aligned}$$

Solving this equation for *inc* using Maxima [220] yields

$$
\begin{aligned}
inc &= [(MTTR^4 \times s^3 + (MTTF \times MTTR^3 - MTTR^4 \times y) \times s^2 + \\
&+ ((-MTTF \times MTTR^3 \times y - MTTF^2 \times MTTR^2) \times p + MTTF^2 \times MTTR^2) \times \\
&\times s) \times x^2 + (-MTTR^4 \times s^4 + (2 \times MTTR^4 \times y + MTTF \times MTTR^3) \times s^3 + \\
&+ ((2 \times MTTF \times MTTR^3 \times y + 2 \times MTTF^2 \times MTTR^2) \times p - MTTR^4 \times y^2 - \\
&- 3 \times MTTF \times MTTR^3 \times y - MTTF^2 \times MTTR^2) \times s^2 + ((-MTTF \times \\
&\times MTTR^3 \times y^2 - 3 \times MTTF^2 \times MTTR^2 \times y - 2 \times MTTF^3 \times MTTR) \times p + \\
&+ MTTF^3 \times MTTR) \times s) \times x + (-MTTR^4 \times y - MTTF \times MTTR^3) \times s^4 + \\
&+ (MTTR^4 \times y^2 + MTTF \times MTTR^3 \times y) \times s^3 + ((2 \times MTTF^2 \times MTTR^2 \times y + \\
&+ 2 \times MTTF^3 \times MTTR) \times p + MTTF \times MTTR^3 \times y^2 - MTTF^2 \times MTTR^2 \times \\
&\times y - 2 \times MTTF^3 \times MTTR) \times s^2 + ((-MTTF^2 \times MTTR^2 \times y^2 - 2 \times MTTF^3 \times \\
&\times MTTR \times y - MTTF^4) \times p + MTTF^2 \times MTTR^2 \times y^2 + MTTF^3 \times MTTR \times \\
&\times y) \times s]/[((MTTF \times MTTR^3 \times p - 2 \times MTTF \times MTTR^3) \times s - MTTF^2 \times \\
&\times MTTR^2) \times x^2 + (-MTTR^4 \times s^3 + (-2 \times MTTF \times MTTR^3 \times p + MTTR^4 \times \\
&\times y + 3 \times MTTF \times MTTR^3) \times s^2 + ((MTTF \times MTTR^3 \times y + 2 \times MTTF^2 \times \\
&\times MTTR^2) \times p - MTTF^2 \times MTTR^2) \times s - MTTF^3 \times MTTR) \times x + MTTR^4 \times \\
&\times s^4 - MTTR^4 \times y \times s^3 + (-2 \times MTTF^2 \times MTTR^2 \times p - MTTF \times MTTR^3 \times \\
&\times y + 2 \times MTTF^2 \times MTTR^2) \times s^2 + ((MTTF^2 \times MTTR^2 \times y + MTTF^3 \times \\
&\times MTTR) \times p - MTTF^2 \times MTTR^2 \times y) \times s]
\end{aligned}
$$

We consider the usage of the value of *inc* below. In the following text, we refer to this approach to path length increase computation our original approach.

7.3.3 Simplification of Path Length Increase Computation

We consider the following approach to simplify path length increase computation. Equation 7.3 can be written as

$$A_{SRLG}(x, p) = A(x) + A(x) \times (1 - p) \times (1 - A(x)). \tag{7.10}$$

The first $A(x)$ in the sum contributes more to the $A_{SRLG}(x, p)$ than the following $A(x)$ that is a part of the product. Therefore, putting second $A(x)$ on the right side of equation 7.10 to be equal 1 simplifies the equation to

$$A_{SRLG}(x,p) \approx A(x) + (1-p) \times (1-A(x)). \tag{7.11}$$

We can use $A_{SRLG}(s,p)$ approximation from equation 7.11 and substitute it in equation 7.8. Solving the equation we get for *inc* yields

$$\begin{aligned}
inc \;=\; & [(MTTR^3 \times s^2 + ((-MTTR^3 \times y - MTTF \times MTTR^2) \times p + MTTF \times \\
\times\;& MTTR^2) \times s) \times x^2 + (-MTTR^3 \times s^3 + (2 \times MTTR^3 \times y + 2 \times MTTF \times \\
\times\;& MTTR^2) \times p \times s^2 + ((-MTTR^3 \times y^2 - 3 \times MTTF \times MTTR^2 \times y - 2 \times \\
\times\;& MTTF^2 \times MTTR) \times p + MTTF^2 \times MTTR) \times s) \times x + (-MTTR^3 \times y - \\
-\;& MTTF \times MTTR^2) \times s^3 + ((2 \times MTTF \times MTTR^2 \times y + 2 \times MTTF^2 \times \\
\times\;& MTTR) \times p + MTTR^3 \times y^2 - MTTF^2 \times MTTR) \times s^2 + ((-MTTF \times \\
\times\;& MTTR^2 \times y^2 - 2 \times MTTF^2 \times MTTR \times y - MTTF^3) \times p + MTTF \times \\
\times\;& MTTR^2 \times y^2 + MTTF^2 \times MTTR \times y) \times s]/[((MTTR^3 \times p - MTTR^3) \times \\
\times\;& s - MTTF \times MTTR^2) \times x^2 + ((MTTR^3 - 2 \times MTTR^3 \times p) \times s^2 + \\
+\;& (MTTR^3 \times y + 2 \times MTTF \times MTTR^2) \times p \times s - MTTF^2 \times MTTR) \times x + \\
+\;& MTTR^3 \times s^3 + (-2 \times MTTF \times MTTR^2 \times p - MTTR^3 \times y + MTTF \times \\
\times\;& MTTR^2) \times s^2 + ((MTTF \times MTTR^2 \times y + MTTF^2 \times MTTR) \times p - \\
-\;& MTTF \times MTTR^2 \times y) \times s]
\end{aligned}$$

In the following text, we refer to this approach to path length increase computation our simplified approach.

7.3.4 Improved Routing and Wavelength Assignment Algorithm

We can utilize the path length increase computation to improve availability of logical channels by selecting spare paths that are less likely to be hit by correlated failure. To illustrate the idea, say we are routing working and spare path of a logical channel. After routing working path, we can route the spare path as the one with the shortest length, ignoring the number and length of common SRLGs between working and spare path. Alternatively, we can route the spare path as the one with the fewest number of common SRLGs, ignoring the length. Finally, it is possible to combine both approaches.

To combine both approaches, we note first that number and length of common SRLGs between working and spare path can be expressed in terms of increase in path length. If we sum that increase with path length of the spare path, we get a single value for path weighting that we can use for comparing paths. More formally, the algorithm is as follows.

1. Compute the working path using Dijkstra shortest path algorithm. Let the set of links used by working path be L_w, and let $S(L_w)$ be a set of all links that contain at least one common SRLG with a link in L_w.
2. To route the spare path, remove from graph links in L_w (links used by working path) and also remove links in $S(L_w)$ (links with commons SRLG with working path). If possible route the spare path using Dijkstra shortest path algorithm and exit with success.
3. Let $s = |S(L_w)|$. Then there are $\binom{s}{r}$ r-subsets of links in $S(L_w)$, and 2^s subsets total. Let $i = 1, 2, \ldots, s$. In $i-th$ step do the following:
   (a) Select next $(s-i)$-subset of links in $S(L_w)$. If all $(s-i)$-subsets have been tried, increment i and continue.
   (b) Remove from graph links in L_w and $s-i$ links selected in subset. Compute the spare path p_{spare} using Dijkstra shortest path algorithm. If the path computation was successful, compute its weight by using the formula

$$weight(p_{spare}) = length(p_{spare}) + PWNSincrease(p_{spare})$$

   where $PWNSincrease()$ is computed using original or simplified approach for each SRLG common to working and spare path. Store the computed path as potential spare path along with its weight.
4. Route the spare path by selecting the lowest weight one.

We will refer to algorithm that uses original path length increase computation as original algorithm, and the one that uses simplified as simplified algorithm.

7.3.5 Example Spare Path Selection

We now consider a simple example to illustrate how the spare path selection algorithm works. Let's say that routing spare path between source and destination nodes results in the following possible spare paths:

- path p_1 of length 100 units with no common SRLGs with working path,
- path p_2 of length 50 units with a single common SRLG with working path, and SRLG gets scored as 30 units of distance increase,
- path p_3 of length 120 units with no common SRLGs with working path,
- path p_4 of length 30 units with two common SRLGs with working path, one SRLG gets scored as 60 units of distance increase, and another as 20 units.

The spare path will be picked among these four possible paths. Weighting the paths and sorting by resulting weight from the lowest to the highest gives the following list:

1. path p_2 with weight 80,

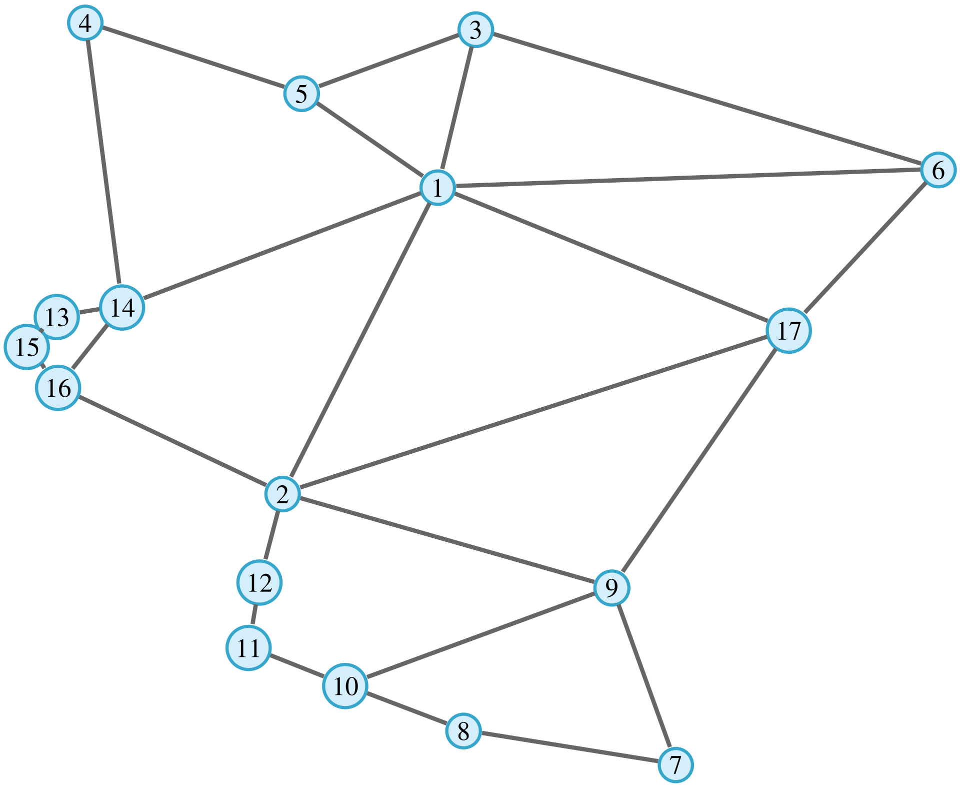


Figure 7.3: nobel-germany network from SNDlib [221].

2. path p_1 with weight 100,
3. path p_4 with weight 110,
4. path p_3 with weight 120.

The algorithm will now pick path p_2 as spare path for logical channel since it is the path with the lowest weight among possible spare paths.

7.4 Case Study

We now compare the base algorithm and the algorithm proposed by Shao et.al. to our algorithm using original and simplified computation of increase in length.

7.4.1 Scenario Description

We evaluate scenarios where all pairs of nodes have bidirectional logical channels established between them. Each logical channel has working and spare path, and DBPP scheme is used.

For comparison of the algorithms we use topologies from SNDlib [221]: nobel-germany (Figure 7.3) and germany50 (Figure 7.4). Node positions are specified using geographical coordinates, and Haversine formula is used for computing distance between each of the two nodes. Distances between nodes are used as link lengths where links exist.

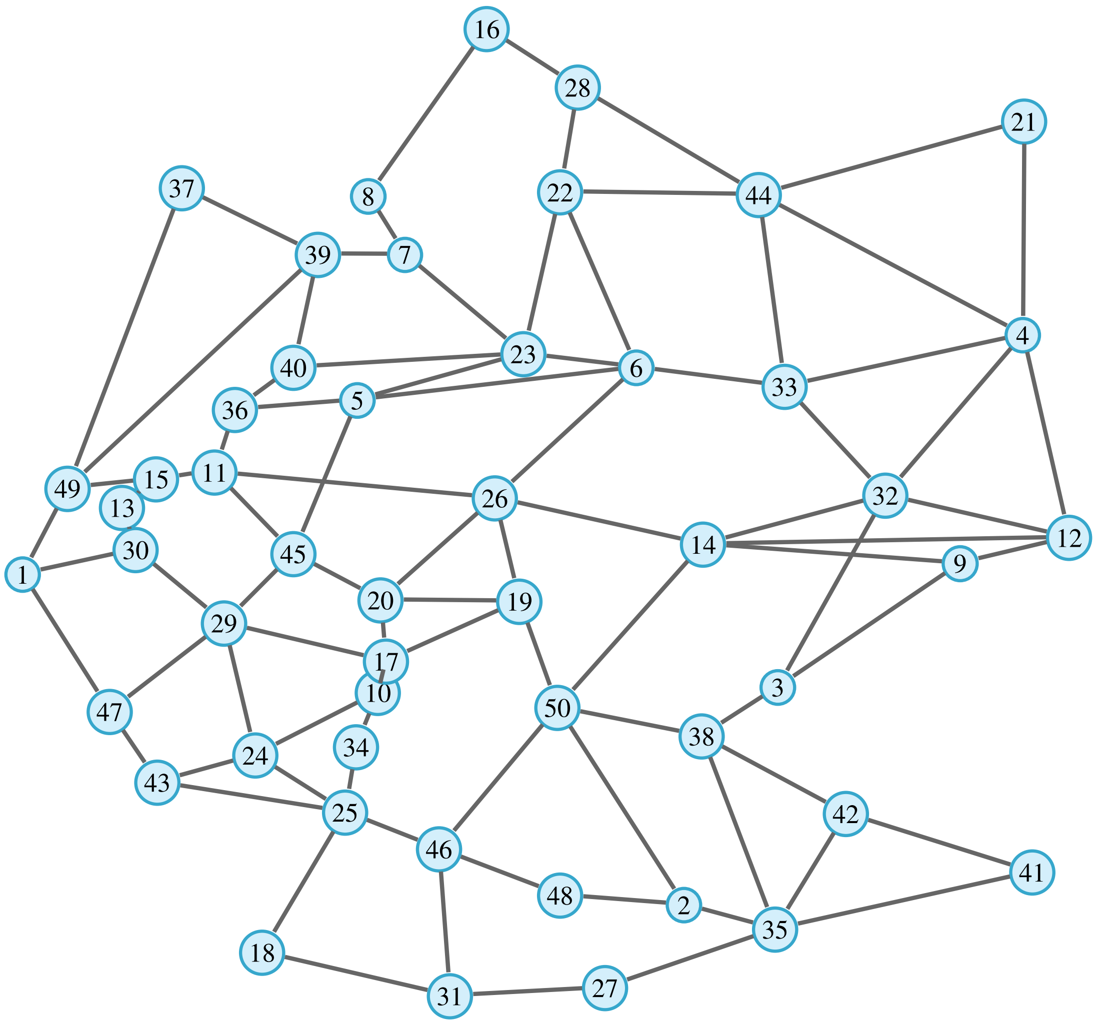


Figure 7.4: germany50 network from SNDlib [221].

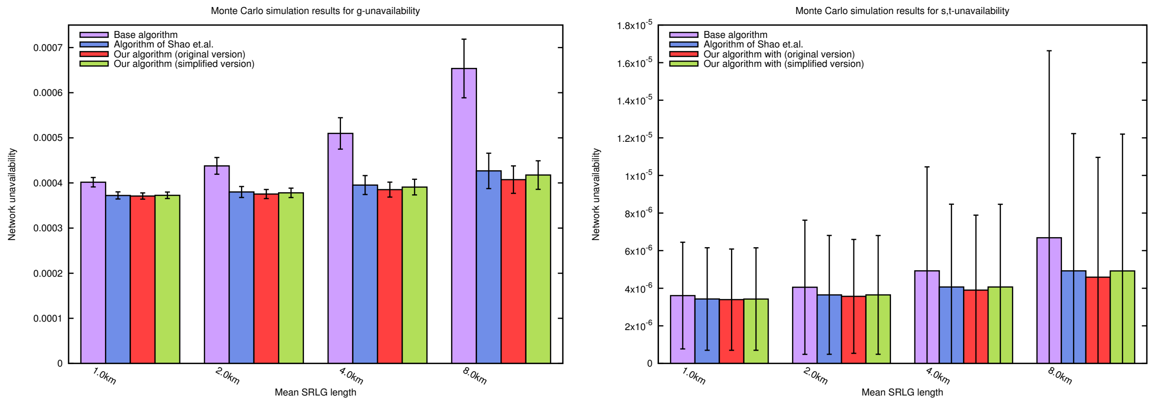


Figure 7.5: Availability results for germany50 network with 20 SRLGs.

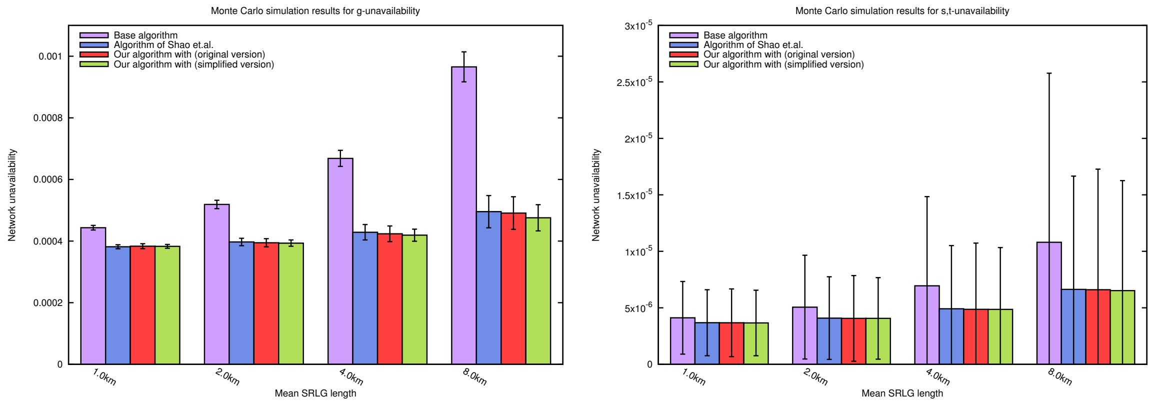


Figure 7.6: Availability results for germany50 network with 40 SRLGs.

For each network we do simulations with 20, 40, 60, and 80 SRLGs present in the network. For each of these numbers of SRLGs we do simulations with mean length of SRLGs set to 1, 2, 4, and 8 kilometers. For each combination of parameters we conduct 10 runs of Monte Carlo simulation lasting 10^9 hours of simulated time.

7.4.2 Simulation Results

Simulation results for g- and s,t-unavailability for large and dense germany50 network can be seen in Figures 7.5, 7.6, 7.7, and 7.8. When comparing our original algorithm and its simplified version, we can observe that for small number of SRLGs the original algorithm performs better. However, as the number of SRLGs increases from 20 to 40, 60, or even 80, the simplified algorithm gives lower unavailability results.

Sparse nobel-germany network shows similar results to germany50. Results are shown in Figures 7.9, 7.10, 7.11, and 7.12. In this network the simplified algorithm shows better results than original algorithm for all the cases we tested.

We can observe that either our original or simplified algorithm performs better or equal to the two algorithms we compared it with in all the cases we tested. The difference is larger on

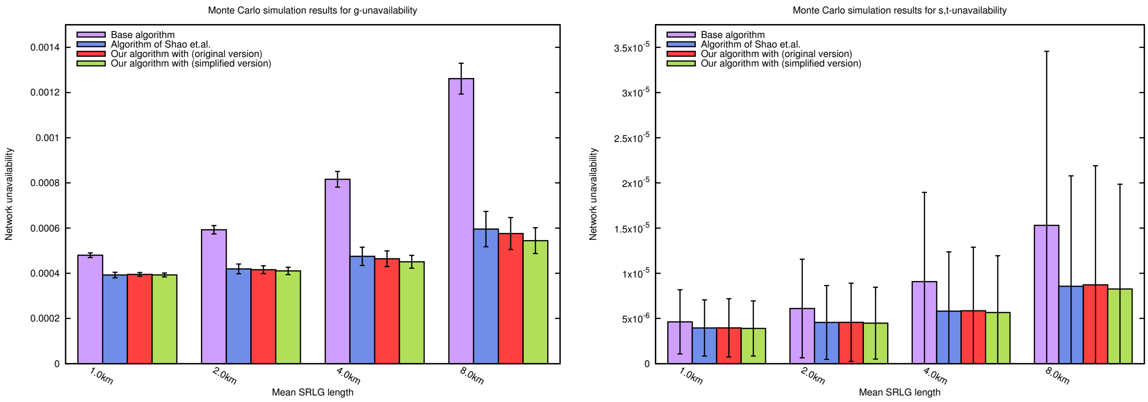


Figure 7.7: Availability results for germany50 network with 60 SRLGs.

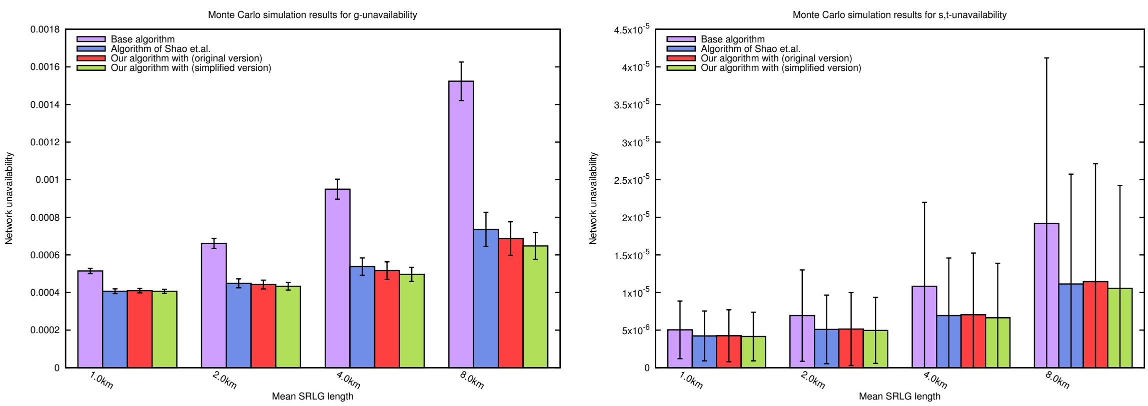


Figure 7.8: Availability results for germany50 network with 80 SRLGs.

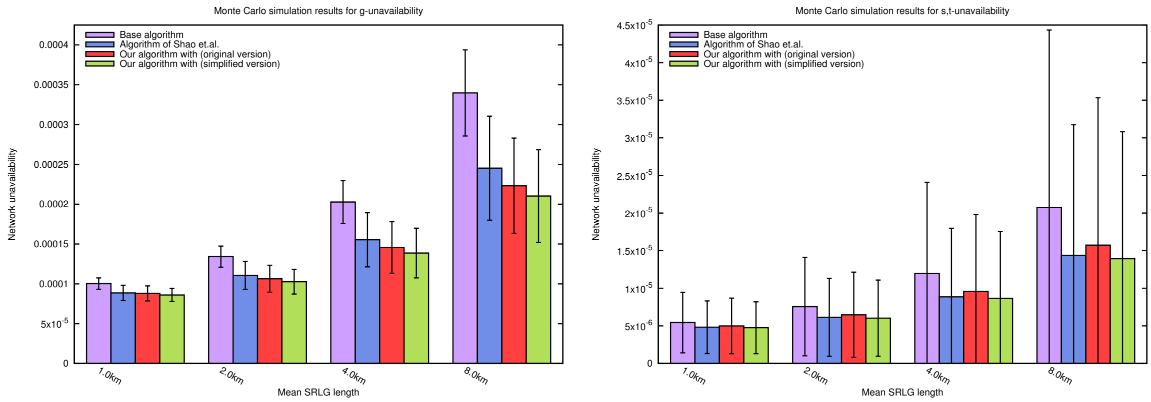


Figure 7.9: Availability results for nobel-germany network with 20 SRLGs.

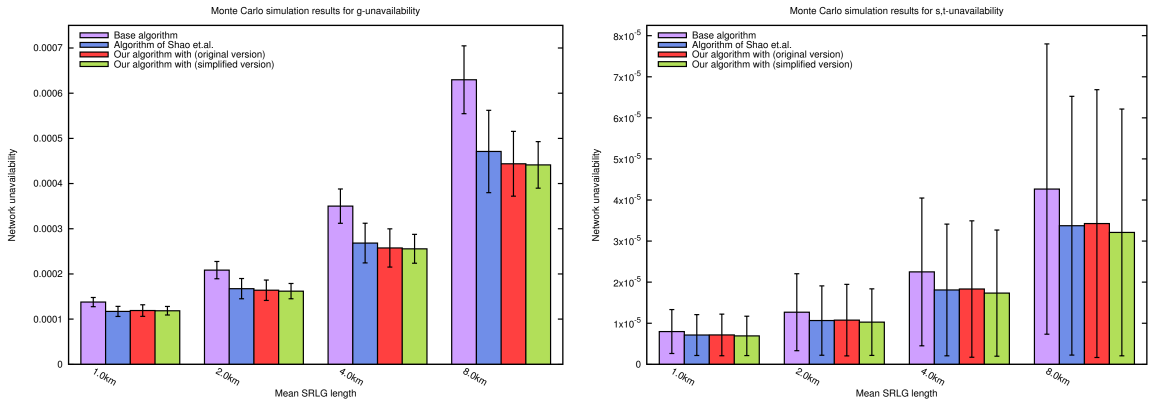


Figure 7.10: Availability results for nobel-germany network with 40 SRLGs.

scenarios with more or longer SRLGs.

7.5 Chapter Conclusions

In this chapter we described an improved algorithm for RWA of working and spare path in presence of SRLGs. The algorithm increases spare path length in presence of SRLGs common to working and spare path. Path length increase makes path less likely to be used; therefore the formula for increase is critical to algorithm performance. We presented the original approach to path length increase computation and also a simplified one.

We found that overall the algorithm performs comparable to or better than existing algorithms in the cases we tested. We measured the algorithm performance in terms of s,t-availability and g-availability. The simplified version of the described RWA algorithm unexpectedly performed better in some scenarios on germany50 network.

In terms of future work, it would be interesting to consider other possible simplifications of the algorithm and compare them in terms of resulting availabilities. Perhaps some simplifications can be found which do not result in decreased availability.

Another direction would be to consider more precise algorithm, by improving some of the

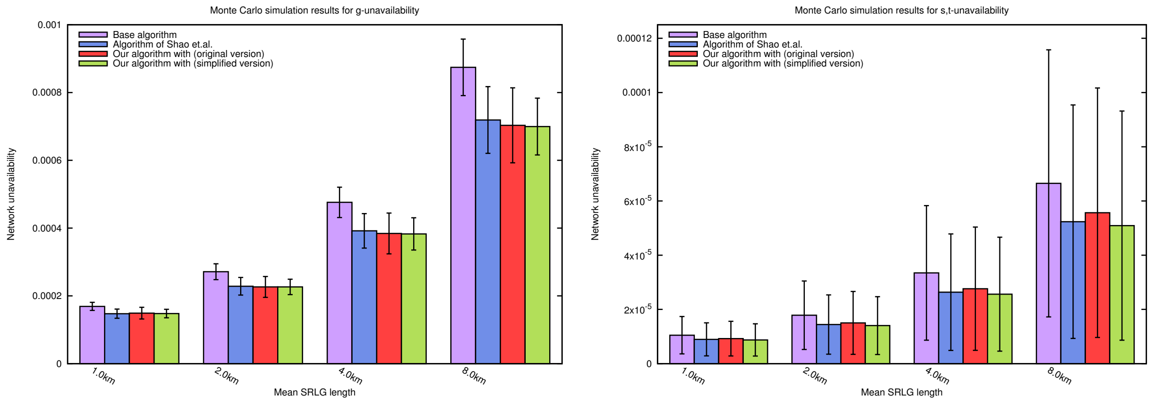


Figure 7.11: Availability results for nobel-germany network with 60 SRLGs.

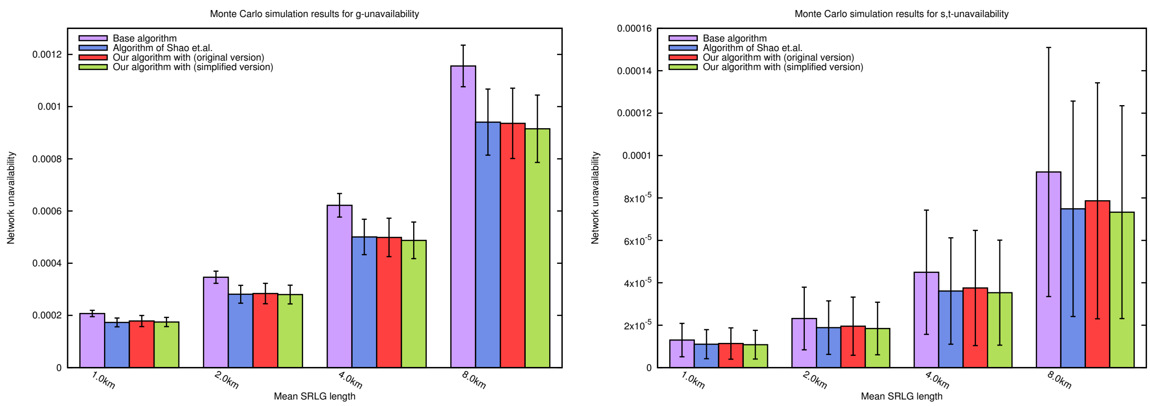


Figure 7.12: Availability results for nobel-germany network with 80 SRLGs.

approximations we used in deriving the formula for path length increase. It is possible that a better approximation will give better availability results.

We consider the problems of computation complexity and run time requirements of maximum SRLG-disjoint algorithms in Chapter 8 and propose an approach to run time reduction.

Chapter 8

Performance Optimization Using Heterogeneous Parallel Programming

8.1 Introduction and Motivation

Simulation methods are often employed for studies of network resilience [7, 95, 222]. In particular, Monte Carlo simulation can be used to give an estimate of network availability and comparison of different network scenarios using different RWA strategies. Monte Carlo simulation require many runs of the same scenario to give a good estimate, so reduction of simulation execution time becomes crucial. One approach is parallelization of suitable parts of simulation, utilizing multi-core central processing units (CPUs) and one or more graphics processing units (GPUs) on one or more compute nodes.

Horizontal scaling means adding more compute nodes to a computer cluster used for running simulations. Vertical scaling, on the other hand, implies adding resources to a single compute node in the cluster, meaning additional CPUs, GPUs, memory etc. When scaling is required to satisfy computation demands, one can utilize horizontal or vertical scaling, or combine both.

This chapter presents our approach to performance optimization of best-effort RWA algorithm using CUDA heterogeneous parallel programming platform enabling code to run on both GPU(s) and CPU(s). Part of the algorithm is moved to GPU for computation to reduce overall execution time. Meanwhile, CPU handles computations not suitable for the GPU. Our approach is based on extending models implemented by ns-3 network simulator [70] with GPU-enabled code, utilizing NVIDIA CUDA programming platform [223]. Compute clusters are becoming increasingly heterogeneous over time, with computation power divided over a number of different processors of vastly disparate computational features [224].

The chapter organized as follows: first we provide an overview related work in Section 8.2. We follow up with description of our approach to algorithm parallelization in Section 8.3. We do performance benchmarks in Section 8.4, and finally conclude along with possible directions for future work.

8.2 Related Work

The usage of GPUs for general purpose computing has been on the rise in recent years [225]. Many application domains of general purpose computing such as artificial intelligence, computational sciences, and various branches of engineering, have benefited greatly and expanded their scope significantly due to computational performance increase resulting from GPUs.

In domain of computer networks, usage of GPUs for IP routing has been studied by Han et al. [226] using custom PacketShader software. Benchmarks have shown that peak performance of NVIDIA GeForce GTX 480 consumer-grade GPU is roughly comparable to ten Intel Xeon X5500 processors. In effect, this result enables a well-designed PC-based router to forward IP packets at 40 Gbps.

8.2.1 Parallelization of Graph Search

Swenson and Riley provided an implementation of CUDA-enabled computation of Floyd-Warshall algorithm used for solving all pairs shortest path problems [227]. The goal was performance improvement of IP routing in ns-3 network simulator and therefore decrease of simulation run time. It was shown that CUDA-enabled routing reduced simulation run time compared to CPU-only Nix-vector routing [228] consistently by a factor over three.

Harish and Narayanan described the approach to parallelization of breadth-first search, single source shortest path and all pairs shortest path using CUDA [229]. They parallelized Dijkstra algorithm using two kernels and found a two orders of magnitude speedup in GPU-enabled code over the code that utilizes only the CPU.

8.2.2 Algorithms for Maximum Link and Shared Risk Link Group Disjoint Paths

Maximum edge (link) disjoint path problem is a variant of k-shortest path problem (in most applications $k = 2$). Say two (disjoint) shortest paths can not be found in a given network; one can ask for maximum link or shared risk link group disjoint paths instead. Algorithms have been studied for RWA in optical networks for many years [120, 216, 219, 230]. In particular, Oki et al. study RWA in presence of SRLGs, introducing the concept of weighted SRLGs. Two paths sharing many SRLGs have low probability to be selected as working and spare path pair, since weight of SRLGs contained on links is added to link cost. Shao et al. [219] present a custom maximum link disjoint path algorithm to RWA problem in optical telecommunication network in presence of SRLGs, taking a different approach than Oki et al. and using number of SRLGs as a metric independently of path length.

The algorithm is as follows:

1. Route the working path using Dijkstra shortest path algorithm. Let the set of links used by working path be L_w, and let $S(L_w)$ be a set of all links that contain at least one common SRLG with a link in L_w.
2. To route the spare path, remove from graph links in L_w (links used by working path) and also remove links in $S(L_w)$ (links with commons SRLG with working path). If possible route the spare path using Dijkstra shortest path algorithm and exit with success.
3. Let $s = |S(L_w)|$. Then there are $\binom{s}{r}$ r-subsets of links in $S(L_w)$, and 2^s subsets total. Let $i = 1, 2, \ldots, s$. In $i - th$ step do the following:
   (a) Select next $(s-i)$-subset of links in $S(L_w)$. If all $(s-i)$-subsets have been tried, increment i and continue.
   (b) Remove from graph links in L_w and $s-i$ links selected in subset. If possible route the spare path using Dijkstra shortest path algorithm and exit with success.

8.2.3 Simulation Models

We used PWNS, an extension of ns-3 network simulator intended for availability study of optical telecommunication network. Models for optical network components such as demultiplexers and multiplexers, fiber, edge devices, optical cross connects, path computation element and control plane are included [6], as well as models for network cables and SRLGs [7]. Component failure and repair events can be simulated; component uptime and downtime are tracked and used for availability estimation. The model is described in Sections 3.5 and 5.3.

We extended path computation element available in PWNS with support for CUDA-based Dijkstra shortest path finding, which is used in maximum disjoint path algorithm.

8.3 Maximum Disjoint Path Algorithm Parallelization Approach

Graphics processors began as general-purpose computing processors with programmable shaders on NVIDIA GeForce FX and AMD Radeon series of graphics cards in 2003 [231]. Three programming languages were used: NVIDA Cg, OpenGL Shading Language (GLSL), and High-level shading language (HLSL), part of Microsoft DirectX suite. Regardless of the requirement to significantly alter algorithms to fit them for the GPU, usage of GPUs for non-graphics computations started to grow and NVIDIA saw the potential in it. GeForce 8 series introduced an application programming interface (API) called Compute Unified Device Architecture (CUDA) intened for general purpose computing on the GPU [223].

8.3.1 Compute Unified Device Architecture

GPUs are very different from commonly used CPUs. GPUs are essentially single instruction, multiple data (SIMD) parallel processors, meaning they have many processing elements able to do the same operation on multiple data elements simultaneously. As we have seen in examples mentioned in Section 8.2, gains from using GPUs vary a lot depending on amount of data-level parallelism present in algorithm one is aiming to accelerate.

Roughly a year after the introduction of CUDA an open standard called OpenCL (short for Open Compute Language) appeared. OpenCL is very similar to CUDA both in application domain and syntax, but has not so far gained market share comparable to CUDA. In addition to the fact that CUDA appeared first, relative unpopularity of OpenCL is also due to lesser amount of literature and advanced programming tools compared to CUDA. While both standards are very similar, they are not compatible [232].

We picked CUDA for this work, and from now on we focus solely on it. CUDA is as an extension of programming languages C/C++ and Fortran. CUDA Application Programing Interface (API) enables programmer to use threads, grouped in blocks of threads. Threads can share memory if required, and thread synchronization mechanism is provided. On the other hand, blocks do not have these features, and execute independently of each other. CUDA programming model is particularly suited for multidimensional arrays. Functions written in CUDA intended for GPU execution are called kernels. When a kernel is called from the code, the number of blocks and threads used for execution is specified. This allows writing kernels once for data arrays of different shapes and sizes.

8.3.2 Algorithm Parallelization Approach

Due to many academic and open source efforts utilizing CUDA, a number of libraries with highly optimized versions of commonly used algorithms (such as reduction, transformation, and sorting) have appeared. However, due to our particular needs we describe below, we implemented our work in pure CUDA C/C++ without using any additional libraries.

To fit our problem into data-parallel framework, we opted for parallelizing the Dijkstra algorithm in maximum disjoint spare path routing stage. Algorithm described in Section 7.2 remains unchanged in stages 1 and 2. Stage 3 is done on the GPU in way that:

- CPU generates 2^s subset of links and stores them in an array, which is copied to GPU.
- GPU kernel is called in $\frac{2^s}{512}$ blocks with 512 threads in each block.*

*Early GeForce and Tesla cards support a maximum of 512 threads per block. Later cards allow 1024 threads per block; regardless, we opted for 512 threads per block to gain wider compatibility, since we had no particular

  - Each thread takes its subset from the array of subsets stored in GPU memory, and stores a copy of the graph in statically allocated array contained in per-thread local memory.
  - Each thread does Dijkstra shortest path algorithm on graph stored in per-thread local memory. If the shortest path is found, it is stored in global memory.
- Array of paths that were found is copied back from GPU.

To simplify the implementation, we also convert link weights to integer. To contain decimals, prior to conversion we multiply the weights by 1000. In our test networks links weights (lengths) are in order of magnitude of 100 (i.e. kilometers), and sum of lengths of all links is in order of magnitude of 1000. Multiplied by 1000, this gives numbers in order of magnitude 10^6 which is way below 10^9 order of magnitude of 32-bit integer maximum value.

One might be concerned here by the amount of memory used for graph copies. However, used memory consistently remained under 1 GB for all scenarios we tested. Since modern entry level domain GeForce GPUs come with over 1 GB of video memory, we did not consider this to be a big issue. However, GPU memory usage can be reduced further by utilizing dynamic instead of static memory allocation for storing per-thread arrays representing graphs.

8.4 Performance Measurements

Our testing and benchmarking system consists of AMD FX-6100 6-core CPU and NVIDIA GeForce GTX 480 GPU. Since we work only with integers, neither 64-bit floating point precision nor extremely large amounts of GPU memory are required for our implementation. Therefore, consumer grade GeForce GPUs work just as well as more expensive professional grade Teslas and Quadros.

For the performance benchmarking we use three networks: 20 nodes and 40 links (Figure 5.5), 25 nodes and 50 links (Figure 5.2), and 30 nodes and 60 links (Figure 5.6). All three networks were first used by Grover et al. [3, 168].

We evaluate performance using the scenario where bidirectional logical channels are established between all pairs of nodes. For the the test networks, 20, 25 and 30 nodes implies 190, 300 and 435 bidirectional logical channels established. We benchmark using scenarios with 20, 40 and 80 SRLGs existing in the network. We assume each SRLG contains two cables.

Program execution time of CPU and GPU versions of the algorithm for 20 node 40 link network is shown in Figure 8.1. We can see that even GPU performance is consistently better, despite large variance in magnitude of difference. If we compare 80 SRLG scenario, GPU computation time is only 3 seconds, which is 7 times better than CPU computation time of 21 seconds. However, in case of 100 SRLGs, GPU takes 39 seconds and CPU takes 72 seconds,

requirement to increase number of threads per block.

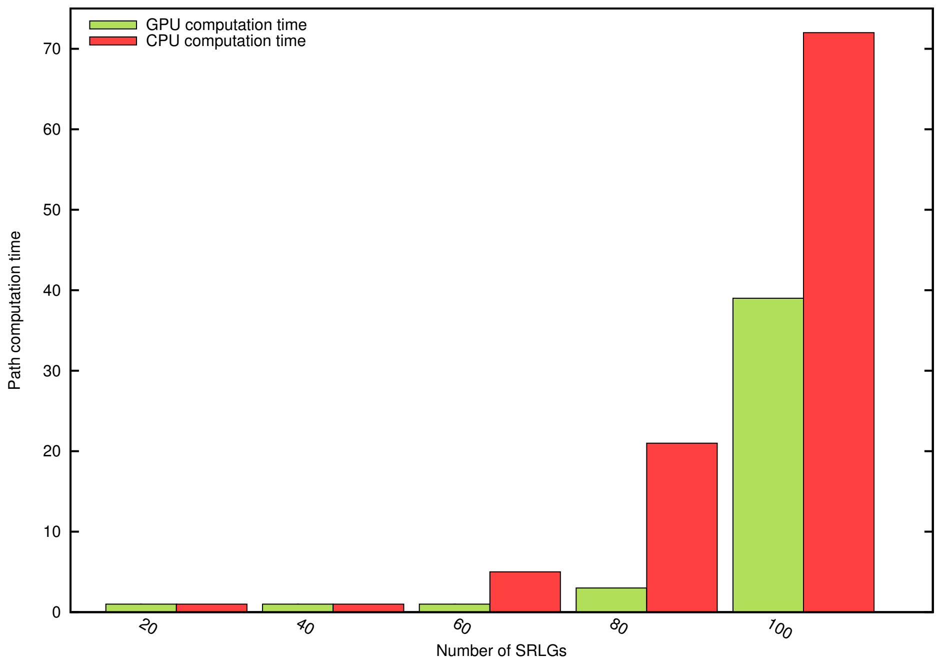


Figure 8.1: Performance measurements for 20 node 40 link topology.

so the difference isn't nowhere as large.

Program execution time of the algorithm for 25 node 50 link network is shown in Figure 8.2. Here we can see that speed is much more consistent, and ends up at nearly 10 times in 100 SRLG case with CPU execution taking 107 and GPU taking only 11 seconds.

Finally, we take a look at 30 node 60 link network results in Figure 8.3. Up to 80 SRLGs GPU is consistently faster, coming again up to 10 times in scenarios 60 and 80 SRLGs. However, for 100 SRLGs scenario this is not the case, and speedup is around 1.5 times.

8.5 Chapter Conclusions

We presented an approach to optimization of maximum shared risk link group-disjoint path algorithm by offloading a part of algorithm to GPU for execution. We believe this approach to be future-proof, considering the increasing heterogeneity of compute components inside computer systems over time, each chip suited for different kind of work. We found the optimization approach we took improving performance very significantly, and decreasing simulation run time. Increasing number of SRLGs has shown an expected impact on performance; on average, more SRLGs increases the number of subsets the algorithm has to process. However, to give definitive performance assessment and select code "hotspots" for optimization further study will be required. Specifically, we are interested in evaluating larger scenarios with more nodes and links in the network.

Our future work will be focused on further optimizing the implementation by increas-

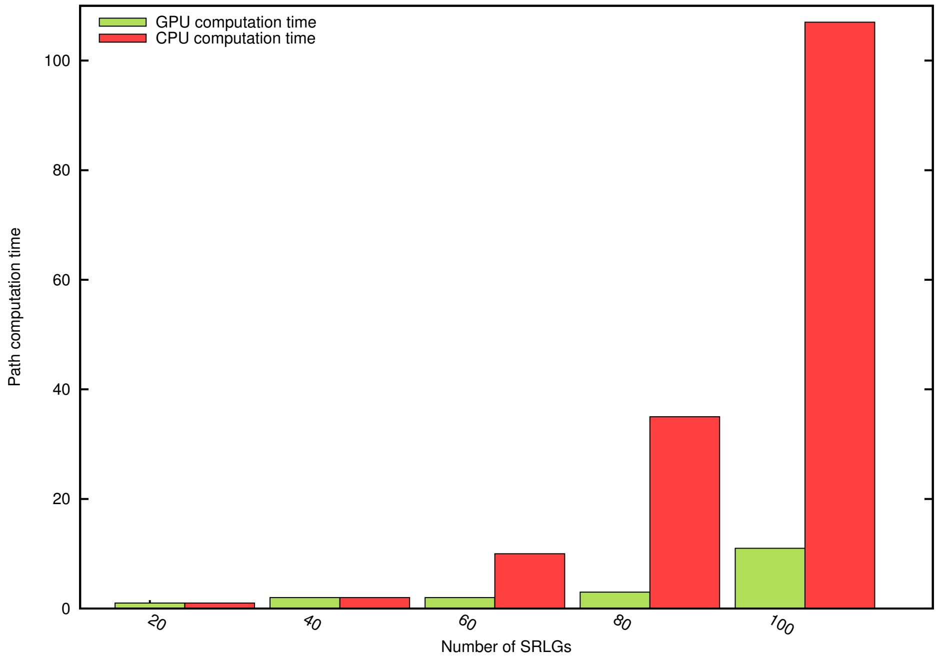


Figure 8.2: Performance measurements for 25 node 50 link topology.

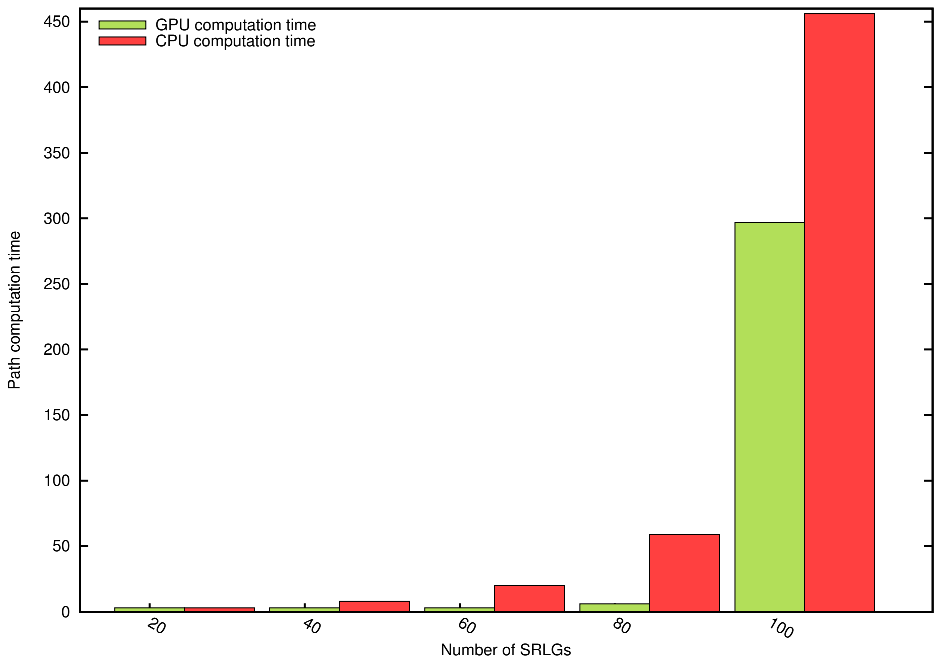


Figure 8.3: Performance measurements for 30 node 60 link topology.

ing amount of parallelism and decreasing memory usage. Dynamic parallelism available on NVIDIA Kepler and subsequent chips, which we have not utilized so far to ensure broader compatibility, is potentially useful for increasing amount of parallelism.

With GPUs making their way into embedded hardware such as NVIDIA Tegra and Adapteva Parallella, it could be possible to use GPUs also for routing in control plane of the optical telecommunication network. Considering the performance and energy efficiency of the GPUs, usage of them in real world optical network control plane is an interesting research direction for the future research work.

Chapter 9

Conclusion

The work presented in this thesis fits in the intersection of optical networks, network modeling and simulation methods network, and network reliability studies. Chapters 2, 3, and 4 provided the overview of selected topics related to our work from each of these large areas. Recent studies within these areas relevant to our work were also referred to.

In particular, Chapter 3 surveyed available optical network simulators. Concluding that none of the available simulators fulfill requirements of our research, we decided to create a new model and implement it in network simulator ns-3. We described our approach to creating a new model of optical telecommunication network components and implementation of a simulator named Photonic WDM Network Simulator. This simulator is open source and available to wider optical network research community for further development.

In Chapter 5 we described an availability model that can be used for network simulation in presence of correlated failures. The results obtained using simulation were validated against results obtained using analytical approach. We then studied the impact of correlated failures of network links on availability, and found out that longer shared risk link groups more negatively impacted network availability. This study and similar studies can help when deciding which shared risk link groups should be shortened or eliminated.

In Chapter 6 we expanded our study of impact of correlated failures on network availability. We used synthetic instances resulting from six different topology models and observed that impact of shared risk link groups can considerably vary depending on underlying network topology. During network planning, these results can help in deciding which topology should be used for the network if shared risk link group can not be avoided. In the process of network evolution, such results can help deciding where to invest to modify the topology over time towards one that better mitigates the negative impact of shared risk link groups.

In Chapter 7 we presented an approach to improving routing and wavelength assignment algorithm of working and spare path in presence of shared risk link groups. The main difference between our approach and existing approaches is the weighting scheme of spare path. The comparison with existing algorithms shows that using our algorithm results in comparable or better network availability than existing algorithms. These results can be applied to improve spare path selection and reduce network outage in presence of correlated failures.

In Chapter 8 we utilized GPU-based parallelization of routing and wavelength algorithm presented in the previous chapter. The resulting parallelized algorithm had improved performance and reduced simulation run time compared to serial one. Routing and wavelength algorithms that consider shared risk link groups show high run-time complexity, hence such parallelization is necessary for them to be feasible in real-world networks.

Bibliography

[1] Miya, T., Terunuma, Y., Hosaka, T., and Miyashita, T., "Ultimate low-loss single-mode fibre at 1.55 μm", Electronics Letters, vol. 15, no. 4, 1979, pp. 106–108.

[2] Castells, M., The information age: Economy, society, and culture. Volume I: The rise of the network society. Blackwell, 1996.

[3] Grover, W. D., Mesh-based survivable networks: options and strategies for optical, MPLS, SONET, and ATM Networking. Prentice Hall, 2004.

[4] Odlyzko, A. M., "Internet traffic growth: Sources and implications", in ITCom 2003. International Society for Optics and Photonics, 2003, pp. 1–15.

[5] Hrvatska agencija za poštu i elektroničke komunikacije (HAKOM). (2010) Porast tržišta elektroničkih komunikacija u republici hrvatskoj u 2010. godini, [Online]. Available: http://www.hakom.hr/default.aspx?ID=787

[6] Miletic, V., Mikac, B., and Dzanko, M., "Modelling optical network components: A network simulator-based approach", in Telecommunications (BIHTEL), 2012 IX International Symposium on. IEEE, 2012, pp. 1–6.

[7] Miletic, V., Mikac, B., and Dzanko, M., "Impact evaluation of physical length of shared risk link groups on optical network availability using monte carlo simulation", in Network and Optical Communications (NOC), 2013 18th European Conference on and Optical Cabling and Infrastructure (OC&I), 2013 8th Conference on. IEEE, 2013, pp. 249–255.

[8] Miletic, V., Maniadakis, D., Mikac, B., and Varoutas, D., "On the influence of the underlying network topology on optical telecommunication network availability under shared risk link group failures", in Design of Reliable Communication Networks (DRCN), 2014 10th International Conference on the. IEEE, 2014, pp. 1–8.

[9] Miletic, V., Subic, T., and Mikac, B., "Optimizing maximum shared risk link group disjoint path algorithm using nvidia cuda heterogeneous parallel programming platform",

in Telecommunications (BIHTEL), 2014 X International Symposium on. IEEE, 2014, pp. 1–6.

[10] Green, P. E., "An all-optical computer network: Lessons learned", Network, IEEE, vol. 6, no. 2, 1992, pp. 56–60.

[11] Maier, M., Optical switching networks. Cambridge University Press, 2008.

[12] Knight, J., Birks, T., Russell, P. S. J., and Atkin, D., "All-silica single-mode optical fiber with photonic crystal cladding", Optics letters, vol. 21, no. 19, 1996, pp. 1547–1549.

[13] Suzuki, K., Kubota, H., Kawanishi, S., Tanaka, M., and Fujita, M., "Optical properties of a low-loss polarization-maintaining photonic crystal fiber", Optics Express, vol. 9, no. 13, 2001, pp. 676–680.

[14] Wu, T.-H., and Lau, R. C., "A class of self-healing ring architectures for sonet network applications", Communications, IEEE Transactions on, vol. 40, no. 11, 1992, pp. 1746–1756.

[15] Ballart, R., and Ching, Y.-C., "SONET: Now it's the standard optical network", Communications Magazine, IEEE, vol. 27, no. 3, 1989, pp. 8–15.

[16] Cavendish, D., "Evolution of optical transport technologies: from SONET/SDH to WDM", Communications Magazine, IEEE, vol. 38, no. 6, 2000, pp. 164–172.

[17] Tucker, R. S., Eisenstein, G., and Korotky, S. K., "Optical time-division multiplexing for very high bit-rate transmission", Lightwave Technology, Journal of, vol. 6, no. 11, 1988, pp. 1737–1749.

[18] Brackett, C. A., "Dense wavelength division multiplexing networks: Principles and applications", Selected Areas in Communications, IEEE Journal on, vol. 8, no. 6, 1990, pp. 948–964.

[19] Feldman, R. D., Harstead, E., Jiang, S., Wood, T. H., and Zirngibl, M., "An evaluation of architectures incorporating wavelength division multiplexing for broad-band fiber access", Lightwave Technology, Journal of, vol. 16, no. 9, 1998, pp. 1546–1559.

[20] Mulugeta, T., and Rasras, M., "Silicon hybrid (de)multiplexer enabling simultaneous mode and wavelength-division multiplexing", Optics Express, vol. 23, no. 2, 2015, pp. 943–949.

[21] Chang, S. H., Chung, H. S., Ryf, R., Fontaine, N. K., Han, C., Park, K. J., Kim, K., Lee, J. C., Lee, J. H., Kim, B. Y. *et al.*, "Mode-and wavelength-division multiplexed

transmission using all-fiber mode multiplexer based on mode selective couplers", Optics Express, vol. 23, no. 6, 2015, pp. 7164–7172.

[22] Mukherjee, B., "Wdm optical communication networks: progress and challenges", Selected Areas in Communications, IEEE Journal on, vol. 18, no. 10, 2000, pp. 1810–1824.

[23] Saleh, A. A., "Transparent optical networking in backbone networks", in Proc. OFC, vol. 2000, no. 3, 2000.

[24] Cantrell, C., "Transparent optical metropolitan-area networks", in Proc. IEEE LEOS, vol. 2, 2003, pp. 608–609.

[25] Callegati, F., Casoni, M., Corazza, G., Raffaelli, C., Chiaroni, D., Masetti, F., and Sotom, M., "Architecture and performance of a broadcast and select photonic switch", Optical Fiber Technology, vol. 4, no. 3, 1998, pp. 266–284.

[26] Pieris, G. R., and Sasaki, G. H., "Scheduling transmissions in WDM broadcast-and-select networks", IEEE/ACM Transactions on Networking (TON), vol. 2, no. 2, 1994, pp. 105–110.

[27] Brackett, C. A., Acampora, A. S., Sweitzer, J., Tangonan, G., Smith, M. T., Lennon, W., Wang, K.-C., and Hobbs, R. H., "A scalable multiwavelength multihop optical network: A proposal for research on all-optical networks", Lightwave Technology, Journal of, vol. 11, no. 5, 1993, pp. 736–753.

[28] Chatterjee, S., and Pawlowski, S., "All-optical networks", Communications of the ACM, vol. 42, no. 6, 1999, pp. 74–83.

[29] Ramaswami, R., and Sivarajan, K. N., "Routing and wavelength assignment in all-optical networks", IEEE/ACM Transactions on Networking (TON), vol. 3, no. 5, 1995, pp. 489–500.

[30] Mokhtar, A., and Azizoğlu, M., "Adaptive wavelength routing in all-optical networks", IEEE/ACM Transactions on Networking (TON), vol. 6, no. 2, 1998, pp. 197–206.

[31] Banerjee, D., and Mukherjee, B., "A practical approach for routing and wavelength assignment in large wavelength-routed optical networks", Selected Areas in Communications, IEEE Journal on, vol. 14, no. 5, 1996, pp. 903–908.

[32] Zang, H., Jue, J. P., Mukherjee, B. *et al.*, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks", Optical Networks Magazine, vol. 1, no. 1, 2000, pp. 47–60.

[33] Iannone, E., and Sabella, R., “Optical path technologies: A comparison among different cross-connect architectures”, Lightwave Technology, Journal of, vol. 14, no. 10, 1996, pp. 2184–2196.

[34] Tzanakaki, A., Zacharopoulos, I., and Tomkos, I., “Optical add/drop multiplexers and optical cross-connects for wavelength routed networks”, in Transparent Optical Networks, 2003. Proceedings of 2003 5th International Conference on, vol. 1. IEEE, 2003, pp. 41–46.

[35] Jeong, G., and Ayanoglu, E., “Comparison of wavelength-interchanging and wavelength-selective cross-connects in multiwavelength all-optical networks”, in INFOCOM’96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE, vol. 1. IEEE, 1996, pp. 156–163.

[36] Desurvire, E., Simpson, J. R., and Becker, P., “High-gain erbium-doped traveling-wave fiber amplifier”, Optics Letters, vol. 12, no. 11, 1987, pp. 888–890.

[37] Koga, M., Watanabe, A., Kawai, T., Sato, K., and Ohmori, Y., “Large-capacity optical path cross-connect system for WDM photonic transport network”, Selected Areas in Communications, IEEE Journal on, vol. 16, no. 7, 1998, pp. 1260–1269.

[38] Jinno, M., Takara, H., Kozicki, B., Tsukishima, Y., Sone, Y., and Matsuoka, S., “Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies”, Communications Magazine, IEEE, vol. 47, no. 11, 2009, pp. 66–73.

[39] Benjamin, D., Trudel, R., Shew, S., and Kus, E., “Optical services over the intelligent optical network”, Communications Magazine, IEEE, vol. 39, no. 9, 2001, pp. 73–78.

[40] Simeonidou, D., Nejabati, R., Arnaud, B. S., Beck, M., Clarke, P., Hoang, D., Hutchison, D., Karmous-Edwards, G., Lavian, T., Leigh, J. *et al.*, “Optical network infrastructure for grid”, in Grid Forum Draft, GFD-I, vol. 36, 2004.

[41] Phillips, M. R., and Ott, D. M., “Crosstalk due to optical fiber nonlinearities in WDM CATV lightwave systems”, Journal of Lightwave Technology, vol. 17, no. 10, 1999, p. 1782.

[42] Ramaswami, R., Sivarajan, K., and Sasaki, G., Optical networks: a practical perspective. Morgan Kaufmann, 2009.

[43] Kovacevic, M., and Acampora, A., “Benefits of wavelength translation in all-optical clear-channel networks”, Selected Areas in Communications, IEEE Journal on, vol. 14, no. 5, 1996, pp. 868–880.

[44] Elmirghani, J. M., and Mouftah, H. T., “Technologies and architectures for scalable dynamic dense WDM networks”, Communications Magazine, IEEE, vol. 38, no. 2, 2000, pp. 58–66.

[45] Mukherjee, B., Banerjee, D., and Ramamurthy, S., “Some principles for designing a wide-area WDM optical network”, Networking, IEEE/ACM Transactions on, vol. 4, no. 5, 1996, pp. 684–696.

[46] Chlamtac, I., Ganz, A., and Karmi, G., “Lightpath communications: An approach to high bandwidth optical WAN’s”, Communications, IEEE Transactions on, vol. 40, no. 7, 1992, pp. 1171–1182.

[47] Sahasrabuddhe, L. H., and Mukherjee, B., “Light trees: optical multicasting for improved performance in wavelength routed networks”, Communications Magazine, IEEE, vol. 37, no. 2, 1999, pp. 67–73.

[48] Antoniades, N., Yoo, S., Bala, K., Ellinas, G., and Stern, T. E., “An architecture for a wavelength-interchanging cross-connect utilizing parametric wavelength converters”, Lightwave Technology, Journal of, vol. 17, no. 7, 1999, pp. 1113–1125.

[49] Subramaniam, S., Azizoglu, M., and Somani, A. K., “All-optical networks with sparse wavelength conversion”, Networking, IEEE/ACM Transactions on, vol. 4, no. 4, 1996, pp. 544–557.

[50] Iness, J., and Mukherjee, B., “Sparse wavelength conversion in wavelength-routed WDM optical networks*”, Photonic Network Communication, vol. 1, no. 3, 1999, pp. 183–205.

[51] Durhuus, T., Joergensen, C., Mikkelsen, B., Pedersen, R. J. S., and Stubkjær, K., “All optical wavelength conversion by SOA’s in a mach-zehnder configuration”, IEEE photonics technology letters, vol. 6, no. 1, 1994, pp. 53–55.

[52] Durhuus, T., Mikkelsen, B., Joergensen, C., Danielsen, S. L., and Stubkjaer, K. E., “All-optical wavelength conversion by semiconductor optical amplifiers”, Lightwave Technology, Journal of, vol. 14, no. 6, 1996, pp. 942–954.

[53] Džanko, M., Mikac, B., Amaya Gonzalez, N., Zervas, G., and Simeonidou, D., “Availability analysis of optical cross-connect implemented by architecture on demand”, in 14th International Conference on Transparent Optical Networks (ICTON), 2012.

[54] Dzanko, M., Mikac, B., and Miletic, V., “Availability of all-optical switching fabrics used in optical cross-connects”, in MIPRO, 2012 Proceedings of the 35th International Convention. IEEE, 2012, pp. 568–572.

[55] Džanko, M., Mikac, B., Miletić, V., Amaya, N., Zervas, G., and Simeonidou, D., "Analytical and simulation availability models of roadm architectures", in International Conference on Telecommunications 2013, 2013.

[56] Bouillet, E., Path routing in mesh optical networks. John Wiley & Sons, 2007.

[57] Maeda, M. W., "Management and control of transparent optical networks", Selected Areas in Communications, IEEE Journal on, vol. 16, no. 7, 1998, pp. 1008–1023.

[58] Mei, Y., and Qiao, C., "Efficient distributed control protocols for WDM all-optical networks", in Computer Communications and Networks, 1997. Proceedings., Sixth International Conference on. IEEE, 1997, pp. 150–153.

[59] Glitho, R. H., and Hayes, S., "Telecommunications management network: vision vs. reality", Communications Magazine, IEEE, vol. 33, no. 3, 1995, pp. 47–52.

[60] Bischoff, M., Huber, M. N., Jahreis, O., and Derr, F., "Operation and maintenance for an all-optical transport network", Communications Magazine, IEEE, vol. 34, no. 11, 1996, pp. 136–142.

[61] Vasseur, J.-P., Pickavet, M., and Demeester, P., Network recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS. Elsevier, 2004.

[62] Wehrle, K., Güneş, M., and Gross, J., Modeling and tools for network simulation. Springer, 2010.

[63] Breslau, L., Estrin, D., Fall, K., Floyd, S., Heidemann, J., Helmy, A., Huang, P., McCanne, S., Varadhan, K., Xu, Y. *et al.*, "Advances in network simulation", Computer, vol. 33, no. 5, 2000, pp. 59–67.

[64] Guizani, M., Rayes, A., Khan, B., and Al-Fuqaha, A., Network modeling and simulation: a practical perspective. John Wiley & Sons, 2010.

[65] Fall, K., "Network emulation in the vint/ns simulator", in Computers and Communications, 1999. Proceedings. IEEE International Symposium on. IEEE, 1999, pp. 244–250.

[66] Lacage, M., "Experimentation tools for networking research", Ph.D. dissertation, Ecole doctorale Stic, Université de Nice Sophia Antipolis, 2010.

[67] Bhide, N. M., and Sivalingam, K. M., "Design of OWns: optical wavelength division multiplexing (WDM) network simulator", in proceedings of First SPIE Optical Networking Workshop. Citeseer. Citeseer, 2000.

[68] Wen, B., Bhide, N. M., Shenai, R. K., Sivalingam, K. M. *et al.*, “Optical wavelength division multiplexing (wdm) network simulator (owns): architecture and performance studies”, SPIE Optical Networks Magazine, 2001, pp. 16–26.

[69] McCanne, S., Floyd, S., and Fall, K., “The lbnl network simulator”, Software on-line: http://www. isi. edu/nsnam, 1997.

[70] Henderson, T. R., Lacage, M., Riley, G. F., Dowell, C., and Kopena, J., “Network simulations with the ns-3 simulator”, SIGCOMM demonstration, 2008.

[71] Lackovic, M., Inkret, R., and Mikac, B., “An approach to an IP over WDM transmission network modelling”, in Transparent Optical Networks, 2002. Proceedings of the 2002 4th International Conference on, vol. 1. IEEE, 2002, pp. 82–85.

[72] Varga, A. *et al.*, “The OMNeT++ discrete event simulation system”, in Proceedings of the European Simulation Multiconference (ESM’2001), vol. 9. sn, 2001.

[73] Riley, G. F., and Henderson, T. R., “The ns-3 network simulator”, in Modeling and Tools for Network Simulation. Springer, 2010, pp. 15–34.

[74] ns 3 Project, ns-3 Manual, version 3.22, [Online]. Available: https://www.nsnam.org/docs/release/3.22/manual/singlehtml/index.html

[75] Lacage, M., “Network experimentation and simulation with ns-3”, in Trilogy Future Internet Summer School. Trilogy consortium, 2009.

[76] Wetherall, D., and Lindblad, C. J., “Extending Tcl for dynamic object-oriented programming”, in Proceedings of the Tcl/Tk Workshop, vol. 95. Citeseer, 1995.

[77] Kim, K., “Integration of OMNeT++ hybrid TDM/WDM-PON models into INET framework”, in OMNeT++ Workshop, 2011.

[78] desRivieres, J., and Wiegand, J., “Eclipse: A platform for integrating development tools”, IBM Systems Journal, vol. 43, no. 2, 2004, pp. 371–383.

[79] Lacage, M., and Henderson, T., “Yet another network simulator”, in Proceeding from the 2006 workshop on ns-2: the IP network simulator. ACM, 2006, p. 12.

[80] ns 3 Project, ns-3 Model Library, version 3.22, [Online]. Available: https://www.nsnam.org/docs/release/3.22/models/singlehtml/index.html

[81] Seguin, G., “Multi-core parallelism for ns-3 simulator”, INRIA Sophia-Antipolis, Tech. Rep, 2009.

[82] Stallman, R. M. *et al.*, Using and porting the GNU compiler collection. Free Software Foundation, 1999.

[83] Lattner, C., "LLVM and Clang: Next generation compiler technology", in The BSD Conference, 2008, pp. 1–2.

[84] Van Der Walt, S., Colbert, S. C., and Varoquaux, G., "The NumPy array: a structure for efficient numerical computation", Computing in Science & Engineering, vol. 13, no. 2, 2011, pp. 22–30.

[85] Millman, K. J., and Aivazis, M., "Python for scientists and engineers", Computing in Science & Engineering, vol. 13, no. 2, 2011, pp. 9–12.

[86] Hunter, J. D., "Matplotlib: A 2D graphics environment", Computing in Science & Engineering, vol. 9, no. 3, 2007, pp. 0090–95.

[87] L'ecuyer, P., Simard, R., Chen, E. J., and Kelton, W. D., "An object-oriented random-number package with many long streams and substreams", Operations Research, vol. 50, no. 6, 2002, pp. 1073–1075.

[88] Matsumoto, M., and Nishimura, T., "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator", ACM Transactions on Modeling and Computer Simulation (TOMACS), vol. 8, no. 1, 1998, pp. 3–30.

[89] Weingartner, E., Vom Lehn, H., and Wehrle, K., "A performance comparison of recent network simulators", in ICC'09. IEEE International Conference on Communications, 2009. IEEE, 2009, pp. 1–5.

[90] Noirie, L., Vigoureux, M., and Dotaro, E., "Impact of intermediate traffic grouping on the dimensioning of multi-granularity optical networks", in Optical Fiber Communication Conference. Optical Society of America, 2001.

[91] Qiao, C., and Yoo, M., "Optical burst switching (OBS)–a new paradigm for an optical internet", Journal of high speed networks, vol. 8, no. 1, 1999, pp. 69–84.

[92] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J., "Openflow: enabling innovation in campus networks", ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, 2008, pp. 69–74.

[93] Puype, B., Vasseur, J., Groebbens, A., De Maesschalck, S., Colle, D., Lievens, I., Pickavet, M., and Demeester, P., "Benefits of GMPLS for multilayer recovery", Communications Magazine, IEEE, vol. 43, no. 7, 2005, pp. 51–59.

[94] Cetinkaya, E. K., Broyles, D., Dandekar, A., Srinivasan, S., and Sterbenz, J. P., "A comprehensive framework to simulate network attacks and challenges", in ICUMT), 2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops. IEEE, 2010, pp. 538–544.

[95] Çetinkaya, E. K., Broyles, D., Dandekar, A., Srinivasan, S., and Sterbenz, J. P., "Modelling communication network challenges for future internet resilience, survivability, and disruption tolerance: A simulation-based approach", Telecommunication Systems, 2011, pp. 1–16.

[96] Sahasrabuddhe, L., Ramamurthy, S., and Mukherjee, B., "Fault management in IP-over-WDM networks: WDM protection versus IP restoration", Selected Areas in Communications, IEEE Journal on, vol. 20, no. 1, 2002, pp. 21–33.

[97] Li, C.-S., and Ramaswami, R., "Automatic fault detection, isolation, and recovery in transparent all-optical networks", Lightwave Technology, Journal of, vol. 15, no. 10, 1997, pp. 1784–1793.

[98] Assi, C., Ye, Y., Shami, A., Dixit, S., and Ali, M., "A hybrid distributed fault-management protocol for combating single-fiber failures in mesh-based DWDM optical networks", in Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE, vol. 3. IEEE, 2002, pp. 2676–2680.

[99] Zhang, J., and Mukheriee, B., "A review of fault management in WDM mesh networks: basic concepts and research challenges", Network, IEEE, vol. 18, no. 2, 2004, pp. 41–48.

[100] Kompella, K., and Swallow, G. (2006, February) RFC 4379: Detecting multi-protocol label switched (MPLS) data plane failures, [Online]. Available: http://www.ietf.org/rfc/rfc4379.txt

[101] Mohan, G., and Murthy, C. S. R., "Lightpath restoration in WDM optical networks", Network, IEEE, vol. 14, no. 6, 2000, pp. 24–32.

[102] Datta, P., Frederick, M., and Somani, A., "Sub-graph routing: A novel fault-tolerant architecture for shared-risk link group failures in WDM optical networks", in Design of Reliable Communication Networks, 2003.(DRCN 2003). Proceedings. Fourth International Workshop on. IEEE, 2003, pp. 296–303.

[103] Mohan, G., Murthy, C., and Somani, A. K., "Efficient algorithms for routing dependable connections in WDM optical networks", IEEE/ACM Transactions on Networking (TON), vol. 9, no. 5, 2001, pp. 553–566.

[104] Gerstel, O., Ramaswami, R., and Sasaki, G. H., "Fault tolerant multiwavelength optical rings with limited wavelength conversion", Selected Areas in Communications, IEEE Journal on, vol. 16, no. 7, 1998, pp. 1166–1178.

[105] Fawaz, W., Daheb, B., Audouin, O., Du-Pond, M., and Pujolle, G., "Service level agreement and provisioning in optical networks", Communications Magazine, IEEE, vol. 42, no. 1, 2004, pp. 36–43.

[106] Gozdecki, J., Jajszczyk, A., and Stankiewicz, R., "Quality of service terminology in IP networks", Communications Magazine, IEEE, vol. 41, no. 3, 2003, pp. 153–159.

[107] Marilly, E., Martinot, O., Papini, H., and Goderis, D., "Service level agreements: a main challenge for next generation networks", in Universal Multiservice Networks, 2002. ECUMN 2002. 2nd European Conference on. IEEE, 2002, pp. 297–304.

[108] Tornatore, M., Maier, G., and Pattavina, A., "WDM network design by ILP models based on flow aggregation", IEEE/ACM Transactions on Networking (TON), vol. 15, no. 3, 2007, pp. 709–720.

[109] Grover, W. D., and Doucette, J., "Advances in optical network design with p-cycles: Joint optimization and pre-selection of candidate p-cycles", in All-Optical Networking: Existing and Emerging Architecture and Applications/Dynamic Enablers of Next-Generation Optical Communications Systems/Fast Optical Processing in Optical Transmission/VCSEL and. IEEE, 2002, pp. WA2–49.

[110] Doshi, B. T., Dravida, S., Harshavardhana, P., Hauser, O., and Wang, Y., "Optical network design and restoration", Bell Labs Technical Journal, vol. 4, no. 1, 1999, pp. 58–84.

[111] Yang, X., and Ramamurthy, B., "Sparse regeneration in translucent wavelength-routed optical networks: Architecture, network design and wavelength routing", Photonic network communications, vol. 10, no. 1, 2005, pp. 39–53.

[112] Venables, B., Grover, W., and MacGregor, M., "Two strategies for spare capacity placement in mesh restorable networks", in Communications, 1993. ICC'93 Geneva. Technical Program, Conference Record, IEEE International Conference on, vol. 1. IEEE, 1993, pp. 267–271.

[113] Iraschko, R., MacGregor, M., and Grover, W., "Optimal capacity placement for path restoration in mesh survivable networks", in Communications, 1996. ICC'96, Conference Record, Converging Technologies for Tomorrow's Applications. 1996 IEEE International Conference on, vol. 3. IEEE, 1996, pp. 1568–1574.

[114] Iraschko, R. R., MacGregor, M. H., and Grover, W. D., "Optimal capacity placement for path restoration in stm or atm mesh-survivable networks", Networking, IEEE/ACM Transactions on, vol. 6, no. 3, 1998, pp. 325–336.

[115] Ramasubramanian, S., and Chandak, A., "Dual-link failure resiliency through backup link mutual exclusion", Networking, IEEE/ACM Transactions on, vol. 16, no. 1, 2008, pp. 157–169.

[116] Elrefaie, A., "Multiwavelength survivable ring network architectures", in Communications, 1993. ICC'93 Geneva. Technical Program, Conference Record, IEEE International Conference on, vol. 2. IEEE, 1993, pp. 1245–1251.

[117] Wasem, O., "Optimal topologies for survivable fiber optic networks using sonet self-healing rings", in Global Telecommunications Conference, 1991. GLOBECOM'91.'Countdown to the New Millennium. Featuring a Mini-Theme on: Personal Communications Services. IEEE, 1991, pp. 2032–2038.

[118] Ellinas, G., Hailemariam, A. G., and Stern, T. E., "Protection cycles in mesh WDM networks", Selected Areas in Communications, IEEE Journal on, vol. 18, no. 10, 2000, pp. 1924–1937.

[119] Schupke, D. A., Gruber, C. G., and Autenrieth, A., "Optimal configuration of p-cycles in WDM networks", in ICC, vol. 2, 2002, pp. 2761–2765.

[120] Oki, E., Matsuura, N., Shiomoto, K., and Yamanaka, N., "A disjoint path selection scheme with shared risk link groups in GMPLS networks", Communications Letters, IEEE, vol. 6, no. 9, 2002, pp. 406–408.

[121] Guo, Y., Kuipers, F., and Van Mieghem, P., "Link-disjoint paths for reliable qos routing", International Journal of Communication Systems, vol. 16, no. 9, 2003, pp. 779–798.

[122] Guo, L., and Li, L., "A novel survivable routing algorithm with partial shared-risk link groups (srlg)-disjoint protection based on differentiated reliability constraints in WDM optical mesh networks", Journal of lightwave technology, vol. 25, no. 6, 2007, pp. 1410–1415.

[123] Ogier, R., and Shacham, N., "A distributed algorithm for finding shortest pairs of disjoint paths", in INFOCOM'89. Proceedings of the Eighth Annual Joint Conference of the IEEE Computer and Communications Societies. Technology: Emerging or Converging, IEEE. IEEE, 1989, pp. 173–182.

[124] Suurballe, J., "Disjoint paths in a network", Networks, vol. 4, no. 2, 1974, pp. 125–145.

[125] Sidhu, D., Nair, R., and Abdallah, S., "Finding disjoint paths in networks", ACM SIGCOMM Computer Communication Review, vol. 21, no. 4, 1991, pp. 43–51.

[126] Ou, C. S., Zhang, J., Zang, H., Sahasrabuddhe, L. H., and Mukherjee, B., "New and improved approaches for shared-path protection in WDM mesh networks", Journal of Lightwave Technology, vol. 22, no. 5, 2004, p. 1223.

[127] Ramamurthy, S., and Mukherjee, B., "Survivable WDM mesh networks. part i-protection", in INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 2. IEEE, 1999, pp. 744–751.

[128] Rajagopalan, B., Pendarakis, D., Saha, D., Ramamoorthy, R. S., and Bala, K., "IP over optical networks: architectural aspects", Communications Magazine, IEEE, vol. 38, no. 9, 2000, pp. 94–102.

[129] Zhou, D., and Subramaniam, S., "Survivability in optical networks", IEEE network, vol. 14, no. 6, 2000, pp. 16–23.

[130] Demeester, P., Gryseels, M., Autenrieth, A., Brianza, C., Castagna, L., Signorelli, G., Clemenfe, R., Ravera, M., Jajszczyk, A., Janukowicz, D. *et al.*, "Resilience in multilayer networks", Communications Magazine, IEEE, vol. 37, no. 8, 1999, pp. 70–76.

[131] Pickavet, M., Demeester, P., Colle, D., Staessens, D., Puype, B., Depré, L., and Lievens, I., "Recovery in multilayer optical networks", Journal of Lightwave technology, vol. 24, no. 1, 2006, p. 122.

[132] De Maesschalck, S., Colle, D., Groebbens, A., Develder, C., Lievens, U., Lagasse, P., Pickavet, M., Demeester, P., Saluta, F., and Quagliatti, M., "Intelligent optical networking for multilayer survivability", Communications Magazine, IEEE, vol. 40, no. 1, 2002, pp. 42–49.

[133] Cetinkaya, E., "Modelling and design of resilient networks under challenges", Ph.D. dissertation, The University of Kansas, December 2013.

[134] Cetinkaya, E. K., and Sterbenz, J. P., "A taxonomy of network challenges", in Design of Reliable Communication Networks (DRCN), 2013 9th International Conference on the. IEEE, 2013, pp. 322–330.

[135] Sterbenz, J. P., Krishnan, R., Hain, R. R., Jackson, A. W., Levin, D., Ramanathan, R., and Zao, J., "Survivable mobile wireless networks: issues, challenges, and research directions", in Proceedings of the 1st ACM workshop on Wireless security. ACM, 2002, pp. 31–40.

[136] Ellison, R. J., Fisher, D. A., Linger, R. C., Lipson, H. F., and Longstaff, T., "Survivable network systems: An emerging discipline", DTIC Document, Tech. Rep., 1997.

[137] Chang, R. K., "Defending against flooding-based distributed denial-of-service attacks: A tutorial", Communications Magazine, IEEE, vol. 40, no. 10, 2002, pp. 42–51.

[138] Dilley, J., Maggs, B., Parikh, J., Prokop, H., Sitaraman, R., and Weihl, B., "Globally distributed content delivery", Internet Computing, IEEE, vol. 6, no. 5, 2002, pp. 50–58.

[139] Sterbenz, J. P., Cetinkaya, E. K., Hameed, M. A., Jabbar, A., and Rohrer, J. P., "Modelling and analysis of network resilience", in Communication Systems and Networks (COMSNETS), 2011 Third International Conference on. IEEE, 2011, pp. 1–10.

[140] Sterbenz, J., Hutchison, D., Çetinkaya, E., Jabbar, A., Rohrer, J., Schöller, M., and Smith, P., "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines", Computer Networks, vol. 54, no. 8, 2010, pp. 1245–1265.

[141] łgorzata Steinder, M., and Sethi, A. S., "A survey of fault localization techniques in computer networks", Science of computer programming, vol. 53, no. 2, 2004, pp. 165–194.

[142] Adler, S., "The slashdot effect: an analysis of three internet publications", Linux Gazette, vol. 38, 1999, p. 2.

[143] Boorstyn, R. R., and Frank, H., "Large-scale network topological optimization", Communications, IEEE Transactions on, vol. 25, no. 1, 1977, pp. 29–47.

[144] McQuillan, J., "Graph theory applied to optimal connectivity in computer networks", ACM SIGCOMM Computer Communication Review, vol. 7, no. 2, 1977, pp. 13–41.

[145] Costa, L. d. F., Rodrigues, F. A., Travieso, G., and Villas Boas, P. R., "Characterization of complex networks: A survey of measurements", Advances in Physics, vol. 56, no. 1, 2007, pp. 167–242.

[146] Fiedler, M., "Algebraic connectivity of graphs", Czechoslovak Mathematical Journal, vol. 23, no. 2, 1973, pp. 298–305.

[147] Wang, H., and Van Mieghem, P., "Algebraic connectivity optimization via link addition", in Proceedings of the 3rd International Conference on Bio-Inspired Models of Network, Information and Computing Sytems. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, p. 22.

[148] Berman, M., Chase, J. S., Landweber, L., Nakao, A., Ott, M., Raychaudhuri, D., Ricci, R., and Seskar, I., “Geni: A federated testbed for innovative network experiments”, Computer Networks, vol. 61, 2014, pp. 5–23.

[149] Gavras, A., Karila, A., Fdida, S., May, M., and Potts, M., “Future internet research and experimentation: the fire initiative”, ACM SIGCOMM Computer Communication Review, vol. 37, no. 3, 2007, pp. 89–92.

[150] Ball, M. O., “Computational complexity of network reliability analysis: An overview”, Reliability, IEEE Transactions on, vol. 35, no. 3, 1986, pp. 230–239.

[151] Abraham, J., “An improved algorithm for network reliability”, Reliability, IEEE Transactions on, vol. 28, no. 1, 1979, pp. 58–61.

[152] Gaver, D., “Time to failure and availability of paralleled systems with repair”, Reliability, IEEE Transactions on, vol. 12, no. 2, 1963, pp. 30–38.

[153] Barlow, R. E., and Wu, A. S., “Coherent systems with multi-state components”, Mathematics of Operations Research, vol. 3, no. 4, 1978, pp. 275–281.

[154] Xue, J., and Yang, K., “Dynamic reliability analysis of coherent multistate systems”, Reliability, IEEE Transactions on, vol. 44, no. 4, 1995, pp. 683–688.

[155] Sterbenz, J. P., Cetinkaya, E. K., Hameed, M. A., Jabbar, A., Qian, S., and Rohrer, J. P., “Evaluation of network resilience, survivability, and disruption tolerance: analysis, topology generation, simulation, and experimentation”, Telecommunication systems, vol. 52, no. 2, 2013, pp. 705–736.

[156] Floyd, S., and Paxson, V., “Difficulties in simulating the internet”, IEEE/ACM Transactions on Networking (TON), vol. 9, no. 4, 2001, pp. 392–403.

[157] Sterbenz, J. P., Medhi, D., Ramamurthy, B., Scoglio, C., Hutchison, D., Plattner, B., Anjali, T., Scott, A., Buffington, C., Monaco, G. E. *et al.*, “The great plains environment for network innovation (gpeni): a programmable testbed for future internet architecture research”, in Testbeds and Research Infrastructures. Development of Networks and Communities. Springer, 2011, pp. 428–441.

[158] Lacage, M., Ferrari, M., Hansen, M., Turletti, T., and Dabbous, W., “Nepi: using independent simulators, emulators, and testbeds for easy experimentation”, ACM SIGOPS Operating Systems Review, vol. 43, no. 4, 2010, pp. 60–65.

[159] Sebos, P., Yates, J., Greenberg, A., and Rubeinstein, D., "Effectiveness of shared risk link group auto-discovery in optical networks", in Optical Fiber Communication Conference and Exhibit, 2002. OFC 2002. IEEE, 2002, pp. 493–495.

[160] Strand, J., and Chiu, A. (2005, May) RFC 4054: Impairments and other constraints on optical layer routing, [Online]. Available: http://www.ietf.org/rfc/rfc4054.txt

[161] Doucette, J., Grover, W. D. *et al.*, "Capacity design studies of span-restorable mesh transport networks with shared-risk link group (srlg) effects", SPIE Opticomm. Citeseer, 2002.

[162] Spragins, J., "Dependent failures in data communication systems", Communications, IEEE Transactions on, vol. 25, no. 12, 1977, pp. 1494–1499.

[163] Lam, Y., and Li, V., "Reliability modeling and analysis of communication networks with dependent failures", Communications, IEEE Transactions on, vol. 34, no. 1, 1986, pp. 82–84.

[164] Burns, H., Chao, C.-W., Dollard, P., Mallon, R., Eslambolchi, H., and Wolfmeyer, P., "FASTAR operations in the real AT&T transport network", in Global Telecommunications Conference, 1993, including a Communications Theory Mini-Conference. Technical Program Conference Record, IEEE in Houston. GLOBECOM'93., IEEE. IEEE, 1993, pp. 229–233.

[165] Lapcevic, O., Lackovic, M., and Mikac, B., "Impact of dependent failures on the availability of the optical network", in Communication Systems, Networks and Digital Signal Processing, 2008. CNSDSP 2008. 6th International Symposium on. IEEE, 2008, pp. 423–427.

[166] Lee, H.-W., Modiano, E., and Lee, K., "Diverse routing in networks with probabilistic failures", Networking, IEEE/ACM Transactions on, vol. 18, no. 6, 2010, pp. 1895–1907.

[167] Strand, J., Chiu, A. L., and Tkach, R., "Issues for routing in the optical layer", Communications Magazine, IEEE, vol. 39, no. 2, 2001, pp. 81–87.

[168] Doucette, J., and Grover, W. D., "Shared-risk logical span groups in span-restorable optical networks: Analysis and capacity planning model", Photonic Network Communications, vol. 9, no. 1, 2005, pp. 35–53.

[169] Frederick, M. T., Datta, P., and Somani, A. K., "Evaluating dual-failure restorability in mesh-restorable WDM optical networks", in Computer Communications and Networks, 2004. ICCCN 2004. Proceedings. 13th International Conference on. IEEE, 2004, pp. 309–314.

[170] Shen, L., Yang, X., and Ramamurthy, B., "Shared risk link group (srlg)-diverse path provisioning under hybrid service level agreements in wavelength-routed optical mesh networks", IEEE/ACM Transactions on Networking (TON), vol. 13, no. 4, 2005, pp. 918–931.

[171] Zhang, Q., Sun, J., Xiao, G., and Tsang, E., "Evolutionary algorithms refining a heuristic: A hybrid method for shared-path protections in WDM networks under SRLG constraints", Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 37, no. 1, 2007, pp. 51–61.

[172] Coudert, D., Pérennes, S., Rivano, H., and Voge, M.-E., "Shared risk resource groups and survivability in multilayer networks", in Transparent Optical Networks, 2006 International Conference on, vol. 3. IEEE, 2006, pp. 235–238.

[173] Coudert, D., Datta, P., Pérennes, S., Rivano, H., and Voge, M.-E., "Shared risk resource group complexity and approximability issues", Parallel Processing Letters, vol. 17, no. 02, 2007, pp. 169–184.

[174] Yuan, S., Varma, S., and Jue, J. P., "Minimum-color path problems for reliability in mesh networks", in INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, vol. 4. IEEE, 2005, pp. 2658–2669.

[175] Cha, M., Chaovalitwongse, W. A., Ge, Z., Yates, J., and Moon, S., "Path protection routing with SRLG constraints to support iptv in WDM mesh networks", in INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings. IEEE, 2006, pp. 1–5.

[176] Agarwal, P., Efrat, A., Ganjugunte, S., Hay, D., Sankararaman, S., and Zussman, G., "The resilience of WDM networks to probabilistic geographical failures", in INFOCOM, 2011 Proceedings IEEE. IEEE, 2011, pp. 1521–1529.

[177] Rahnamay-Naeini, M., Pezoa, J., Azar, G., Ghani, N., and Hayat, M., "Modeling stochastic correlated failures and their effects on network reliability", in Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on. IEEE, 2011, pp. 1–6.

[178] Wosinska, L., Colle, D., Demeester, P., Katrinis, K., Lackovic, M., Lapcevic, O., Lievens, I., Markidis, G., Mikac, B., Pickavet, M., Puype, B., Skorin-Kapov, N., Staessens, D., and Tzanakaki, A., "Network resilience in future optical networks", Towards Digital Optical Networks, 2009, pp. 253–284.

[179] Perrone, L. F., Main, C. S., and Ward, B. C., “SAFE: Simulation automation framework for experiments”, in Proceedings of the 2012 Winter Simulation Conference. Winter Simulation Conference, 2012, pp. 249–260.

[180] Schupke, D. A., Autenrieth, A., and Fischer, T., “Survivability of multiple fiber duct failures”, in Third International Workshop on the Design of Reliable Communication Networks (DRCN), Budapest, Hungary, 2001.

[181] Saxena, J., and Goel, A., “Reliability and maintainability of passive optical component”, International Journal of Computer Trends and Technology, vol. 21, no. 1, 2011.

[182] De Maesschalck, S., Colle, D., Lievens, I., Pickavet, M., Demeester, P., Mauz, C., Jaeger, M., Inkret, R., Mikac, B., and Derkacz, J., “Pan-european optical transport networks: an availability-based comparison”, Photonic Network Communications, vol. 5, no. 3, 2003, pp. 203–225.

[183] Betker, A., Gerlach, C., Hülsermann, R., Jäger, M., Barry, M., Bodamer, S., Späth, J., Gauger, C., and Köhn, M., “Reference transport network scenarios”, MultiTeraNet Report, July, 2003.

[184] Cicic, T., Hansen, A. F., Kvalbein, A., Hartmann, M., Martin, R., Menth, M., Gjessing, S., and Lysne, O., “Relaxed multiple routing configurations: IP fast reroute for single and correlated failures”, Network and Service Management, IEEE Transactions on, vol. 6, no. 1, 2009, pp. 1–14.

[185] Ahuja, S., Ramasubramanian, S., and Krunz, M., “Srlg failure localization in optical networks”, IEEE/ACM Transactions on Networking (TON), vol. 19, no. 4, 2011, pp. 989–999.

[186] Xi, K., Chao, H. J., and Guo, C., “Recovery from shared risk link group failures using IP fast reroute”, in Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on. IEEE, 2010, pp. 1–7.

[187] Suchara, M., Xu, D., Doverspike, R., Johnson, D., and Rexford, J., “Network architecture for joint failure recovery and traffic engineering”, in Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems. ACM, 2011, pp. 97–108.

[188] Maniadakis, D., Balmpakakis, A., and Varoutas, D., “On the temporal evolution of backbone topological robustness”, in Network and Optical Communications (NOC), 2013 18th European Conference on and Optical Cabling and Infrastructure (OC&I), 2013 8th Conference on. IEEE, 2013, pp. 129–136.

[189] Schintler, L. A., Gorman, S. P., Reggiani, A., Patuelli, R., Gillespie, A., Nijkamp, P., and Rutherford, J., “Complex network phenomena in telecommunication systems”, Networks and Spatial Economics, vol. 5, no. 4, 2005, pp. 351–370.

[190] Çetinkaya, E. K., Alenazi, M. J., Cheng, Y., Peck, A. M., and Sterbenz, J. P. G., “On the fitness of geographic graph generators for modelling physical level topologies”, in Proceedings of the 5th IEEE/IFIP International Workshop on Reliable Networks Design and Modeling (RNDM), Almaty, September 2013.

[191] Segovia, J., Calle, E., and Vila, P., “Availability analysis of GMPLS connections based on physical network topology”, in Optical Network Design and Modeling, 2008. ONDM 2008. International Conference on. IEEE, 2008, pp. 1–6.

[192] Henderson, T. R., Roy, S., Floyd, S., and Riley, G. F., “ns-3 project goals”, in Proceeding from the 2006 workshop on ns-2: the IP network simulator. ACM, 2006, p. 13.

[193] Penrose, M., Random geometric graphs. Oxford University Press Oxford, 2003, vol. 5.

[194] Gabriel, K. R., and Sokal, R. R., “A new statistical approach to geographic variation analysis”, Systematic Biology, vol. 18, no. 3, 1969, pp. 259–278.

[195] Toussaint, G. T., “The relative neighbourhood graph of a finite planar set”, Pattern recognition, vol. 12, no. 4, 1980, pp. 261–268.

[196] Eppstein, D., Paterson, M. S., and Yao, F. F., “On nearest-neighbor graphs”, Discrete & Computational Geometry, vol. 17, no. 3, 1997, pp. 263–282.

[197] Waxman, B. M., “Routing of multipoint connections”, Selected Areas in Communications, IEEE Journal on, vol. 6, no. 9, 1988, pp. 1617–1622.

[198] Vinciguerra, S., Frenken, K., and Valente, M., “The geography of internet infrastructure: an evolutionary simulation approach based on preferential attachment”, Urban Studies, vol. 47, no. 9, 2010, pp. 1969–1984.

[199] Flaxman, A. D., Frieze, A. M., and Vera, J., “A geometric preferential attachment model of networks”, Internet Mathematics, vol. 3, no. 2, 2006, pp. 187–205.

[200] Yook, S.-H., Jeong, H., and Barabási, A.-L., “Modeling the internet’s large-scale topology”, Proceedings of the National Academy of Sciences, vol. 99, no. 21, 2002, pp. 13 382–13 386.

[201] Masuda, N., Miwa, H., and Konno, N., “Geographical threshold graphs with small-world and scale-free properties”, Physical Review E, vol. 71, no. 3, 2005, p. 036108.

[202] Bradonjić, M., Hagberg, A., and Percus, A. G., "The structure of geographical threshold graphs", Internet Mathematics, vol. 5, no. 1-2, 2008, pp. 113–139.

[203] Calvert, K. L., Doar, M. B., and Zegura, E. W., "Modeling internet topology", Communications Magazine, IEEE, vol. 35, no. 6, 1997, pp. 160–163.

[204] Jabbar, A., Shi, Q., Cetinkaya, E., and Sterbenz, J. P., "Ku-locgen: Location and cost-constrained network topology generator", ITTC Technical Report ITTC-FY2009-TR-45030-01, The University of Kansas, Lawrence, KS, Tech. Rep., 2008.

[205] Ma, X., Kim, S., and Harfoush, K., "Towards realistic physical topology models for internet backbone networks", in High-Capacity Optical Networks and Enabling Technologies (HONET), 2009 6th International Symposium on. IEEE, 2009, pp. 36–42.

[206] Pavan, C., Morais, R. M., Ferreira da Rocha, J. R., and Pinto, A. N., "Generating realistic optical transport network topologies", Journal of Optical Communications and Networking, vol. 2, no. 1, 2010, pp. 80–90.

[207] Liu, H., and Tobagi, F. A., "Physical topology design for all-optical networks", Optical Switching and Networking, vol. 5, no. 4, 2008, pp. 219–231.

[208] Li, Y., and Gouda, M. G., "IP fast reroute in networks with shared risk links", in NETWORKING 2010. Springer, 2010, pp. 213–226.

[209] Kandula, S., Katabi, D., and Vasseur, J.-P., "Shrink: A tool for failure diagnosis in IP networks", in Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data. ACM, 2005, pp. 173–178.

[210] Erdős, P., and Rényi, A., "On the evolution of random graphs", Publ. Math. Inst. Hungar. Acad. Sci, vol. 5, 1960, pp. 17–61.

[211] Cai, G., Hirtle, S., and Williams, J., "Mapping the geography of cyberspace using telecommunications infrastructure information", TeleGeo, 1999, pp. 6–7.

[212] Barabási, A.-L., and Albert, R., "Emergence of scaling in random networks", science, vol. 286, no. 5439, 1999, pp. 509–512.

[213] Jamakovic, A., and Uhlig, S., "Influence of the network structure on robustness", in Networks, 2007. ICON 2007. 15th IEEE International Conference on. IEEE, 2007, pp. 278–283.

[214] Mello, D. A., Schupke, D. A., Scheffel, M., and Waldman, H., "Availability maps for connections in WDM optical networks", in Design of Reliable Communication Networks, 2005.(DRCN 2005). Proceedings. 5th International Workshop on. IEEE, 2005, pp. 8–pp.

[215] Jamakovic, A., and Uhlig, S., "On the relationship between the algebraic connectivity and graph's robustness to node and link failures", in Next Generation Internet Networks, 3rd EuroNGI Conference on. IEEE, 2007, pp. 96–102.

[216] Manohar, P., Manjunath, D., and Shevgaonkar, R., "Routing and wavelength assignment in optical networks from edge disjoint path algorithms", Communications Letters, IEEE, vol. 6, no. 5, 2002, pp. 211–213.

[217] Li, G., Wang, D., Gallivan, T., and Doverspike, R., "On shared risk link group optimization", in Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2012 and the National Fiber Optic Engineers Conference. IEEE, 2012, pp. 1–3.

[218] Li, G., Wang, D., Gallivan, T., and Doverspike, R., "On shared risk link group optimization [invited]", Journal of Optical Communications and Networking, vol. 4, no. 11, 2012, pp. B52–B57.

[219] Shao, X., Bai, Y., Cheng, X., Yeo, Y.-K., Zhou, L., and Ngoh, L. H., "Best effort SRLG failure protection for optical WDM networks", Optical Communications and Networking, IEEE/OSA Journal of, vol. 3, no. 9, 2011, pp. 739–749.

[220] Maxima. (2014) Maxima, a computer algebra system. version 5.35.1. http://maxima.sourceforge.net/, [Online]. Available: http://maxima.sourceforge.net/

[221] Orlowski, S., Wessäly, R., Pióro, M., and Tomaszewski, A., "Sndlib 1.0–survivable network design library", Networks, vol. 55, no. 3, 2010, pp. 276–286.

[222] Lackovic, M., and Mikac, B., "Analytical vs. simulation approach to availability calculation of circuit switched optical transmission network", in Telecommunications, 2003. ConTEL 2003. Proceedings of the 7th International Conference on, vol. 2. IEEE, 2003, pp. 743–750.

[223] Kirk, D., "NVIDIA CUDA software and GPU parallel computing architecture", in ISMM, vol. 7, 2007, pp. 103–104.

[224] Keckler, S. W., Dally, W. J., Khailany, B., Garland, M., and Glasco, D., "GPUs and the future of parallel computing", IEEE Micro, vol. 31, no. 5, 2011, pp. 7–17.

[225] Dally, W. J., "The end of denial architecture and the rise of throughput computing", in Keynote speech at Desgin Automation Conference, 2010.

[226] Han, S., Jang, K., Park, K., and Moon, S., "Packetshader: a gpu-accelerated software router", ACM SIGCOMM Computer Communication Review, vol. 41, no. 4, 2011, pp. 195–206.

[227] Swenson, B. P., and Riley, G. F., "Simulating large topologies in ns-3 using brite and cuda driven global routing", in Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013, pp. 159–166.

[228] Riley, G. F., Ammar, M. H., and Zegura, E. W., "Efficient routing using nix-vectors", in High Performance Switching and Routing, 2001 IEEE Workshop on. IEEE, 2001, pp. 390–395.

[229] Harish, P., and Narayanan, P., "Accelerating large graph algorithms on the GPU using CUDA", in High performance computing–HiPC 2007. Springer, 2007, pp. 197–208.

[230] Xiao, G., and Pan, X., "Heuristic for the maximum disjoint paths problem in wavelength-routed networks with shared-risk link groups [invited]", Journal of Optical Networking, vol. 3, no. 1, 2004, pp. 38–49.

[231] Wu, E., and Liu, Y., "Emerging technology about GPGPU", in Circuits and Systems, 2008. APCCAS 2008. IEEE Asia Pacific Conference on. IEEE, 2008, pp. 618–622.

[232] Du, P., Weber, R., Luszczek, P., Tomov, S., Peterson, G., and Dongarra, J., "From CUDA to OpenCL: Towards a performance-portable solution for multi-platform GPU programming", Parallel Computing, vol. 38, no. 8, 2012, pp. 391–407.

List of Figures

List of Tables

Biography

Vedran Miletić was born in Rijeka in 1986. He received M.Ed. in mathematics and informatics from the University of Rijeka Faculty of Arts and Sciences (today Department of Mathematics), Rijeka, Croatia in 2009.

Since 2009 he is working as a research and teaching assistant at University of Rijeka Department of Informatics, Rijeka. In the same year he started his Ph.D. studies at University of Zagreb Faculty of Electrical Engineering. He is doing Ph.D. research at the Department for Telecommunications, under the supervision of professor Branko Mikac, Ph.D. His Ph.D. research is focused on evaluation of optical telecommunication network reliability and availability using Monte Carlo method and optimization of routing and wavelength assignment algorithm. From 2009 to 2013 he coordinated the activities of Croatian Linux Users Group local subsidiary in Rijeka. Since 2012 he is a developer of ns-3 network simulator, and also coordinator of NVIDIA CUDA Teaching Center at University of Rijeka. Since 2013 he is working as adjunct teaching assistant at University of Rijeka Faculty of Engineering.

He has participated in international program committees of one workshop and he served as a reviewer for two conferences.

List of Published Works

Papers in Conference Proceedings

1. Miletić, V., Šubić, T., Mikac, B., "Optimizing Maximum Shared Risk Link Group Disjoint Path Algorithm Using NVIDIA CUDA Heterogeneous Parallel Programming Platform.", Proceedings on the 2014 X International Symposium on Telecommunications (BIHTEL), October 2014., pp. 1–6.
2. Miletić, V., Maniadakis, D., Mikac, B., Varoutas, D., 2014., "On the Influence of the Underlying Network Topology on Optical Telecommunication Network Availability Under Shared Risk Link Group Failures.", Proceedings of the 2014 10th International Conference on the Design of Reliable Communication Networks (DRCN), April 2014., pp. 1–8.
3. Miletić, V., Mikac, B., Džanko, M., "Impact Evaluation of Physical Length of Shared Risk Link Groups on Optical Network Availability Using Monte Carlo Simulation.", Pro-

ceedings of the 2013 18th European Conference on Networks and Optical Communications (NOC) and 8th Conference on Optical Cabling & Infrastructure (OC&I), July 2013., pp. 249–255.

4. Džanko, M., Mikac, B., Miletić, V., "Analytical and Simulation Availability Models of ROADM Architectures.", Proceedings of the 12th International Conference on Telecommunications (ConTEL), June 2013., pp. 39–45.
5. Miletić, V., Mikac, B., Džanko, M., "Modelling Optical Network Components: A Network Simulator-Based Approach.", Proceedings on the 2012 IX International Symposium on Telecommunications (BIHTEL), October 2012., pp. 1–6.
6. Džanko, M., Mikac, B., Miletić, V., "Availability of All-Optical Switching Fabrics Used in Optical Cross-Connects.", Proceedings on the 35th Convention International MIPRO 2012, May 2012., pp. 613–617.

Životopis

Vedran Miletić rođen je u Rijeci 1986. godine. Diplomirao je u polju edukacije matematike i informatike na Filozofskom fakultetu (danas Odjel za matematiku Sveučilišta) Sveučilišta u Rijeci 2009. godine.

Od 2009. godine radi kao asistent na Odjelu za informatiku Sveučilišta u Rijeci. Iste godine započeo je doktorski studij na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu. Doktorska istraživanja provodi na Zavodu za telekomunikacije, pod mentorstvom prof.dr.sc. Branka Mikca. Njegovo doktorsko istraživanje fokusirano je na procjenu raspoloživosti optičke telekomunikacijske mreže korištenjem Monte Carlo metode i optimizaciju algoritma usmjeravanja i dodjele valnih duljina. Od 2009. do 2013. godine vodio je aktivnosti Riječke podružnice Hrvatske udruge Linux korisnika. Od 2012. godine jedan je od razvijača mrežnog simulatora ns-3 i koordinator NVIDIA CUDA nastavnog centra na Sveučilištu u Rijeci. Od 2013. godine radi kao asistent vanjski suradnik na Tehničkom fakultetu Sveučilišta u Rijeci.

Bio je član međunarodnog programskog odbora jedne radionice i recenzirao radove za dvije konferencije.